

książka kucharska

L^AT_EX

Marcin Borkowski

Bartłomiej Przybylski



książka kucharska

L^AT_EX

Marcin Borkowski

Bartłomiej Przybylski



Niektóre prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiejkolwiek formie, wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym w celach komercyjnych bez wyraźnej zgody autora jest zabronione.



Ta książka jest objęta licencją Creative Commons BY-NC-SA. Pełna treść licencji dostępna jest na stronie: <http://creativecommons.org/licenses/by-nc-sa/3.0/pl/legalcode>.

Znaki występujące w tekście mogą być zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autorzy oraz Polskie Towarzystwo Matematyczne dołożyli wszelkich starań, aby zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autorzy oraz Polskie Towarzystwo Matematyczne nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

© 2015 Marcin Borkowski & Bartłomiej Przybylski

Grafiki wykorzystane w książce pochodzą z zasobów <http://xkcd.com> udostępnianych na licencji Creative Commons BY-NC.

Niniejsza publikacja ma charakter bezpłatny i może być wykorzystywana jedynie w celach niekomercyjnych.

ISBN (e-book): 978-83-937220-3-7

ISBN (zestaw): 978-83-937220-4-4

Wydano w Polsce.

Wydawca:

Polskie Towarzystwo Matematyczne
ul. Śniadeckich 8, 00-956 Warszawa
<http://www.ptm.org.pl>
zgptm@ptm.org.pl

Spis treści

Wstęp	1
I. Elementarz	
T_EX działa inaczej	4
Trochę historii	4
T _E X a L ^A T _E X	5
L ^A T _E X a Word	6
Pytania	9
Zaczynamy	10
Silniki	10
Dystrybucje L ^A T _E Xa	11
Edytory oraz IDE	12
Sprawdźmy, czy działa!	14
Pytania i ćwiczenia	15
Zrozumieć dokument	16
Odstępy, instrukcje i otoczenia	16
Odstępy	16
Instrukcje	17
Otoczenia	19
Klasy dokumentów i pakiety	20
Klasy dokumentów	20
Pakiety	22
Pakiet L ^A ipsum	22
Plik źródłowy i jego konstrukcja	23
Komentarze i znaki specjalne	24
Komentarze	24

<i>Spis treści</i>	V
Znaki specjalne	25
Kodowanie pliku źródłowego i fontu	26
Krótka historia kodowań	26
Kodowanie pliku źródłowego	28
Kodowanie fontu	28
Pliki produkowane przez L ^A T _E Xa	29
Pytania i ćwiczenia	31
Dokumenty w języku polskim	34
Pakiet polski	35
Zestaw klas mwc _l s	36
Przykładowy dokument	37
Ćwiczenia	37
Podsumowanie	38
II. Receptury	
Wprowadzenie do receptur	41
Wyliczenia	42
Listy nienumerowane	43
Listy numerowane	44
Zmiana symbolu wypunktowania lub sposobu numeracji listy	45
Zmiana symbolu wypunktowania wybranych elementów listy	47
Zmiana symbolu wypunktowania wszystkich list nienumerowanych	48
Zmiana sposobu numeracji wszystkich list numerowanych	49
Wyliczenia „długie”	50
Rozdziały i paragrafy	52
Oznaczanie rozdziału	54
Oznaczanie podrozdziału i mniejszych jednostek	56
Jednostki nienumerowane	58
Dzielenie tytułu rozdziału na wiersze	60
Warianty tytułu do żywej paginy i spisu treści	62
Części większe od rozdziałów	62
Strona tytułowa	64
Mikrotypografia	65
Wyróżnianie fragmentu tekstu	65
Cudzysłowy, myślniki, łączniki...	66

<i>Spis treści</i>	VI
Przypisy	68
Adresy email i stron internetowych	69
Litery z akcentami	71
Twarde spacje	72
Kształt akapitu	73
Usuwanie i wstawianie wcięcia akapitowego	74
Dokumenty bez wcięć akapitowych	75
Dodatkowy odstęp pionowy	77
Wyrównanie akapitu inaczej niż do obu marginesów	78
Akapity węższe niż kolumna	80
Cytaty	83
Skład dwułamowy	88
Otoczanie fragmentu tekstu ramką	90
Automatyzacja	92
Odwołania do elementów numerowanych	92
Skorowidze	95
Spis treści i inne spisy	98
Bibliografie	100
Tabele	104
Najprostsza tabela	105
Komórki wielowierszowe	106
Gdzie jest moja tabelka?	108
Tabele nienumerowane	111
Łączenie komórek w poziomie	113
Łączenie komórek w pionie	115
Kolory	117
Nadawanie koloru elementom dokumentu	118
Mieszanie farb	119
Grafiki	121
Dołączanie grafiki	122
Wersje testowe dokumentu	125
Grupowanie grafik	126
Pływające grafiki	131
Obracanie grafik	134

Układ strony	136
Wymuszenie przejścia do nowej strony	136
Zakaz przejścia do nowej strony	137
Zmiana rozmiaru strony i marginesów	138
Standardowe style stopek i nagłówków	145
Zmiana układu stopek i nagłówków na pojedynczej stronie	146
Ręczna zmiana tekstu stopek i nagłówków	149
Nagłówki niestandardowe	151
Różne nagłówki i stopki dla stron parzystych i nieparzystych	154
O nagłówkach w książkach	160
Poziome strony	165
Otoczenia numerowane	167
Tworzenie otoczeń numerowanych	168
Numeracja zależna	170
Numeracja hierarchiczna	172
Numeracja hierarchiczna i zależna	174
Nazwy twierdzeń	176
Spisy otoczeń numerowanych	177
Dowody	179
Zmiana symbolu końca dowodu	180
Rodzaje otoczeń numerowanych	182
Zamiana numeru otoczenia i jego nazwy miejscami	184
Wzory i formuły matematyczne	186
Prosty wzór	186
Wzory wycentrowane	187
Ułamki, potęgi i indeksy dolne. Symbol Newtona.	190
Nawiasy	191
Symbole działań, relacji, zbiorów itp.	193
Popularne funkcje	194
Alfabety matematyczne	195
Akcenty nad literami	195
Funkcje dane kilkoma wzorami	195
Układy równań	196
Ciągi równości	197
Wzory związane ze sobą	200
Macierze	201

<i>Spis treści</i>	VIII
Wielokropki matematyczne	202
Odstępy w trybie matematycznym	203
Teksty we wzorach	205
Matematyka i tekst – zgrabne połączenie	206
Wielkości z jednostkami	208
Polecane lektury	209

Dodatki

Symbole	211
Podstawowe symbole specjalne	211
Alfabet grecki	212
Symbole matematyczne	213
Operatory	213
Relacje	213
Strzałki	214
Akcenty	214
Symbole duże	215
Kropki	215
Nawiasy	215
Inne symbole	216
Pakiety	217
Pakiet bamboleo	217
Pakiet latex-ksiazka-kucharska	218
Skorowidz poleceń i nazw	219

Wstęp

Masz przed sobą, drogi Czytelniku, książkę dotyczącą oprogramowania, które z roku na rok zdobywa coraz większą rzeszę użytkowników. System \LaTeX , bo o nim mowa, może być wykorzystywany do tworzenia wszelkiego rodzaju publikacji elektronicznych i drukowanych, od klasycznych dokumentów, przez prezentacje multimedialne, a na posterach skończywszy.

Książka ta dotyczy jednak pierwszego z zastosowań, czyli tworzenia klasycznych dokumentów – niezależnie czy mamy przez to na myśli krótkie teksty pisane prozą, czy też skomplikowane publikacje matematyczne. Stworzenie zbioru doświadczeń, z którego mogliby czerpać zarówno początkujący, jak i zaawansowani użytkownicy systemu \LaTeX wymagało od nas, autorów, wprowadzenia pewnego podziału treści. Tak oto, książka została podzielona na dwie części.

Pierwsza z nich, nazwana *Elementarz*, wprowadza Czytelnika w zagadnienia dotyczące podstaw korzystania z systemu \LaTeX . Zebraliśmy tam wszystkie informacje, które powinien (a w wielu przypadkach *musi*) znać każdy użytkownik tego oprogramowania. Dotyczą one w szczególności tego, dlaczego \LaTeX działa tak, a nie inaczej, jak się z nim porozumiewać i co zrobić, żeby stworzyć w nim zgodny z polskimi normami typograficznymi dokument zbudowany z akapitów tekstu. Staraliśmy się wzbogacić tę część o ciekawostki historyczne i głębsze wyjaśnienia pewnych zagadnień tak, aby nawet zaawansowani użytkownicy mogli z tej części wynieść coś nowego.

Druga część, nazwana *Receptury*, stanowi główną zawartość tej książki. Czytelnik znajdzie w niej gotowe przykłady pokazujące, jak umieścić pewne elementy (np. listy, grafiki, tabele, spisy) w dokumencie. Przykłady te, wzbogacone o ciekawe [komentarze, zbudowane zostały \(w większości\) jako sa-](#)

Staraliśmy się jak mogliśmy...

modzielne paragrafy, do których można zaglądać w razie potrzeby. *Receptury*, w przeciwieństwie do *Elementarza*, nie są bowiem podręcznikiem, a raczej encyklopedią, w której poszukuje się jedynie konkretnych haseł. Te hasła powinny zaspokoić potrzeby nawet wymagającego użytkownika.

Książka wyposażona jest też w dodatki. Większość z nich znajduje się tu, choć jeden jest szczególny. *Dodatek C*, dotyczący instalacji i konfiguracji środowiska pracy, został opublikowany jako plik PDF, który można pobrać z odpowiedniej strony Polskiego Towarzystwa Matematycznego, poświęconej tej książce. Jest tak dlatego, że nie przypuszczamy, aby ktokolwiek chciał korzystać z niego regularnie – raz zainstalowane oprogramowanie działa dobrze przez długi czas.

W tym samym miejscu (a więc na stronie Polskiego Towarzystwa Matematycznego) można też pobrać archiwum z plikami źródłowymi wszystkich przykładów wykorzystanych w *Recepturach*.

Wierzymy, że ta książka stanie się zachętą do korzystania z \LaTeX a dla tych, którzy dotąd się na to nie zdecydowali, oraz kompendium przydatnych wskazówek dla tych, którzy \LaTeX a mieli okazję już poznać.

Część I

Elementarz

T_EX działa inaczej

Trochę historii

Wszystko zaczęło się 30 marca 1977 roku. Donald E. Knuth [usiadł przy](#) swoim masywnym, drewnianym biurku ze zdobionymi nogami, w stylu typowym dla gabinetów Uniwersytetu Stanforda, by rozpocząć dzień od sprawdzenia bieżącej poczty. Przez niewielkie okno za jego plecami wpadał klimat poranka w San Francisco w towarzystwie ciepłego, wiosennego słońca. Knuth zapalił olbrzymie, kubańskie cygaro, by w kłębach dymu sięgnąć po koperty, które jego asystentka położyła wcześniej na rogu blatu. Leżało ich tam więcej niż zwykle, ale szczególną jego uwagę zwróciła jedna, nieco grubsza od pozostałych. Był to *proof* [książki, nad którą Knuth pracował przez ostatnich](#) kilka lat. Kiedy jednak otworzył przesyłkę i wysunął znajdujące się w środku kartki, z przerażenia otworzył usta, pozwalając cygaru wypaść, a w czasie, gdy upadało ono na szarą wykładzinę, zdał sobie sprawę, że wolałby zacząć ten dzień zupełnie inaczej.

No... może nie do końca tak to wyglądało. Nie wiemy, czy Knuth miał masywne, drewniane biurko, ani czy palił cygara. Niestety, nie wiemy nawet, czy miał asystentkę. Jednego jesteśmy jednak pewni – Donald E. Knuth nie był zadowolony z tego, jak miałyby wyglądać kolejna część jego *Sztuki programowania komputerów*. W swojej innej książce, *Typografia cyfrowa*, pisał potem:

Poświęciłem piętnaście lat na pisanie tamtych książek, ale jeżeli miały wyglądać fatalnie, nie chciałem pisać już więcej.

Knuth postanowił napisać własny system komputerowego składu tekstu, dając sobie na jego ukończenie mniej więcej sześć miesięcy. [I chociaż jego](#)

Donald E. Knuth, ur. w 1938 r. – amerykański matematyk i informatyk, obecnie emerytowany profesor Uniwersytetu Stanforda.

proof – w poligrafii wydruk próbny, który wykonuje się przed rozpoczęciem druku całego nakładu.

Tak naprawdę, kolejne wersje pojawiały się odpowiednio w latach 1978 (T_EX), 1982 (T_EX82) i 1989 (T_EX 3).

pierwsza wersja pojawiła się publicznie już w 1978 roku, dopiero jedenaście lat później (w 1989 r.) system ten, nazwany TeX, został uznany za ukończony. Od tego czasu do programu wprowadzane są jedynie poprawki błędów, a wraz z nimi do numeru wersji dodawane są kolejne cyfry rozwinięcia dziesiętnej liczby π . Do grudnia 2014 roku, czyli w ciągu dwudziestu pięciu lat od wydania trzeciej wersji, numer ten rozwinął się do 3.14159265.

TeX a LaTeX

TeX został utworzony z myślą o tym, aby prawie każdy mógł stworzyć wysokiej jakości książki niewielkim (choć lepszym słowem będzie *rozsądnym*) nakładem sił. Wraz z językiem opisu dokumentów o tej samej nazwie, językiem *Metafont*, służącym do opisu czcionek oraz rodziną krojów *Computer Modern*, TeX daje nie tylko olbrzymie możliwości typograficzne, ale też pewność, że tworzone dokumenty będą identyczne, niezależnie od specyfiki komputera użytego do pracy.

Generowanie dokumentu w TeXu polega na utworzeniu tekstowego pliku źródłowego, który następnie, w procesie zwanym *kompilacją*, przekształcany jest na wyjściowy dokument, najczęściej w formacie PDF.

W czasie tworzenia pliku źródłowego wykorzystuje się polecenia języka TeX. Dzięki nim można przekazać mechanizmowi składającemu tekst, jakich rezultatów od niego oczekujemy. TeX rozumie pewien zestaw podstawowych instrukcji, z których można budować, niczym z klocków LEGO, bardziej zaawansowane polecenia, zwane *makrami*. Nie trzeba było więc długo czekać na powstanie pakietu rozszerzającego możliwości TeXa. W rzeczywistości, już na początku lat osiemdziesiątych Leslie Lamport opublikował pokazny zbiór TeXowych makr, nazwany LaTeX. Dziś z podstawowej wersji systemu TeX mało kto korzysta, a LaTeX tak bardzo zdomował się w umysłach użytkowników, że często obu nazw używa się zamiennie. Obecna wersja LaTeXa to LaTeX 2_ε i o tej właśnie wersji traktuje ta książka.

LaTeX w swojej naturze odpowiada metodologii WYSIWYM, wedle której autor określa jedynie strukturę logiczną i treść dokumentu, pozostawiając w rękach automatycznego systemu zmartwienia dotyczące wyglądu i odpowiedniego rozmieszczenia elementów na stronie. Całość wynika z założenia,

Nazwa TeX pochodzi od greckiego $\tau\acute{\epsilon}\xi\nu\eta$ – „techné” – co oznacza „sztuka”, ale i „rzemiosło”. Z tego powodu czyta się ją *tech*.

Jeśli z jakiejś przyczyny niemożliwe jest wypisanie stylizowanego tekstu, należy pisać odpowiednio TeX i LaTeX.

Leslie Lamport, ur. w 1941 r. – amerykański informatyk.

LaTeX 2_ε czytamy *latech dwa epsilon*.

WYSIWYM – ang. *what you see is what you mean*.

że to, jak dokument wygląda, jest rzeczą całkowicie oddzielną względem tego, jakie informacje zawiera.

L^AT_EX a Word

Kiedy w 1983 roku firma *Microsoft* opublikowała pierwszą wersję swojego oprogramowania *Multi-tool Word* dla systemów *Xenix*, znanego później jako *Microsoft Word*, można było przewidywać rychły zmierzch systemów komputerowego składu tekstów opartych na plikach źródłowych opisujących dokument na rzecz edytorów WYSIWYG. Okazało się jednak, że wysoka elastyczność oraz jakość estetyczna i typograficzna tekstów generowanych przez system T_EX sprawia, że jego użytkowników nie tylko nie zaczęło ubywać, ale zaczęło być coraz więcej.

Spór L^AT_EX kontra *Word* ciągnie się nieprzerwanie od lat osiemdziesiątych. Istnieją bowiem grupy entuzjastów gotowych całym swym dobytkiem bronić wyboru jedyne go słusznego systemu składu tekstu (którykolwiek z dwóch wymienionych mamy na myśli). I chociaż najbardziej dyplomatyczna, a jednocześnie prawdziwa odpowiedź na pytanie „co jest lepsze?” brzmi „to zależy”, dobrze jest nie tylko być jej świadomym, ale też umieć ją zastosować w praktyce. To jeden z powodów, dla których warto znać oba narzędzia. Poniżej zebrałiśmy – bez najmniejszych skrupułów – niektóre cechy i zastosowania, które przemawiają na korzyść L^AT_EXa. Mamy nadzieję, że to przekona niepewnych, iż warto czytać tę książkę dalej.

– **Typografia i układ dokumentu.** Jedną z istotnych cech L^AT_EXa jest odzielenie treści od stylu. Wiele osób z całego świata włożyło niewyobrażalną pracę w to, aby dokumenty tworzone w L^AT_EXu były eleganckie i typograficznie nienaganne, tym samym zdejmując tę odpowiedzialność z autorów tekstów. Oznacza to, że pisząc dokument w L^AT_EXu możemy skupić się na jego treści, nie tracąc energii na zbędne rozmyślenia o formie. Tak naprawdę rzadko kiedy zachodzi bowiem potrzeba zmiany domyślnego układu dokumentu.

W praktyce dzieje się tak dzięki zastosowaniu *znakowania semantycznego*. Autor tekstu określa jedynie logiczne umiejscowienie poszczególnych elementów (takich jak tekst główny, nagłówki, tytuły czy przypisy), a L^AT_EX zaj-

WYSIWYG – ang. *what you see is what you get*.

Edytory WYSIWYG pozwalają na bieżąco obserwować i modyfikować finalny wygląd dokumentu.

muje się tym, aby efekt finalny był jak najlepszy. Jest to podejście bardzo podobne do występującego w rzeczywistości. Kiedy planujemy wydać książkę, zwykle wysyłamy jej tekst do wydawnictwa, w którym sztab specjalistów zajmuje się kwestiami składu. W przypadku L^AT_EXa rolę tych specjalistów przejmują system komputerowy, nauczony najważniejszych reguł gry.

Lata doświadczenia w korzystaniu z edytorów WYSIWYG mogą początkowo wzbudzać poczucie dyskomfortu związane z brakiem pełnej kontroli nad wyglądem tekstu. [Pamiętajmy jednak, że projektowanie pięknych dokumentów jest rzeczą trudną, wymagającą nierzadko wielu lat doświadczenia. I nawet jeśli wydaje nam się, że jesteśmy wirtuozami czcionek, odstępów i kolorów, wcale tak nie musi być.](#)

W rzeczywistości L^AT_EX pozwala wpłynąć na dowolny parametr stylu, ale lepiej zostawić to ekspertom.

– **Tryb matematyczny.** L^AT_EX nadaje się doskonale do składania tekstów wypełnionych wzorami matematycznymi. Początkowo opisywanie wzorów poleceniami może się wydawać wędrownką z zasłoniętymi oczami, ale po poznaniu podstaw okazuje się, że efekty są warte poświęceń. Niech argumentem będzie tu fakt, że L^AT_EX wykorzystywany jest do generowania wzorów matematycznych w serwisach internetowych takich jak *Wikipedia*, a z rozwiązań zastosowanych w L^AT_EXu czerpali autorzy analogicznych narzędzi, np. edytora formuł dostępnego w pakiecie *OpenOffice.org*.

– **Uniwersalność.** Pliki źródłowe L^AT_EXa są zwykłymi plikami tekstowymi – trudno sobie wyobrazić bardziej uniwersalny format. Takie *pliki źródłowe* można przetworzyć np. na popularny format PDF, korzystając z oprogramowania L^AT_EX zainstalowanego na dowolnym komputerze wyposażonym w system operacyjny *Windows*, *Linux* lub *Mac OS*. Dzięki całkowitej autonomii L^AT_EXa, efekty składu (a więc i końcowy dokument) będą identyczne niezależnie od wykorzystywanego środowiska pracy.

– **Rozszerzenia.** L^AT_EX to system umożliwiający szybkie tworzenie prostych dokumentów o jasnej strukturze, takich jak książki, artykuły czy raporty. Ponieważ jednak wymagania autorów często wykraczają poza podstawowe zastosowania, entuzjaści tworzą dodatkowe rozszerzenia umożliwiające wykorzystanie potencjału systemu L^AT_EX. Dziś są to tysiące dodatków pozwalających, przykładowo, tworzyć zaawansowane tabele, grafiki i wykresy, skrowidze, a nawet prezentacje multimedialne czy plakaty.

– **Doskonała obsługa obszernych tekstów.** Tworzenie rozległych dokumentów (np. książek) zawierających dziesiątki rozdziałów oraz setki wzorów, tabel i grafik to wymagająca praca. Zaletą \LaTeX a są wbudowane mechanizmy umożliwiające zachowanie pełnej kontroli nad strukturą dokumentu. Możliwość podziału pliku źródłowego na części, automatyczne numerowanie rozdziałów, paragrafów, przypisów i innych elementów, prosty w użyciu system odwołań, ale także łatwe generowanie spisów treści czy skorowidzów – to tylko niektóre funkcje dostępne na wyciągnięcie ręki.

– **Wsparcie.** Społeczność \LaTeX a to społeczność entuzjastów. Napisano wiele książek dotyczących tego systemu i otwarto wiele forów internetowych i grup dyskusyjnych umożliwiających wzajemną wymianę doświadczeń. Nie ma więc się czego obawiać – zawsze znajdzie się ktoś gotowy odpowiedzieć na nurtujące nas pytania i pomóc w przypadku niepowodzeń.

– **Koszt.** Mogłoby się wydawać, że system umożliwiający tworzenie profesjonalnie wyglądających dokumentów bez konieczności przejmowania się warstwą wizualną kosztuje dużo pieniędzy. Okazuje się jednak, że zarówno TeX, jak i \LaTeX są całkowicie darmowe. Co więcej, wspomniane wcześniej rozszerzenia także są dostępne bezpłatnie.

Żeby być jednak uczciwym, należy wspomnieć chociaż o najistotniejszych wadach \LaTeX a:

– **Inność.** \LaTeX opiera się na plikach źródłowych, a informacje o strukturze dokumentu i umieszczanych w nim elementach przekazywane są do kompilatora za pomocą poleceń tekstowych. Jest to zgoła inne podejście niż to, do którego przyzwyczaiły nas edytory WYSIWYG, takie jak *Microsoft Word*. Aby stworzyć jakiegokolwiek dokumenty w \LaTeX u, trzeba więc poznać przynajmniej podstawowe polecenia.

– **Tworzenie tekstów bez jasnej struktury.** \LaTeX służy do składania tekstów o jasnej strukturze. Tworzenie pozbawionych niej dokumentów może okazać się bardziej pracochłonne w \LaTeX u niż w jakimkolwiek edytorze wizualnym (choć to raczej kwestia doświadczenia). W praktyce jednak pisanie dłuższych tekstów niesie za sobą konieczność ich strukturyzacji. W ich przypadku mniejsze szanse na zawał serca związany ze stresem mają użytkownicy \LaTeX a.

– **Niezrozumiałe błędy.** Ponieważ tworzenie dokumentu w \LaTeX u wymaga skompilowania pliku źródłowego, może się okazać, że w czasie tego procesu wystąpiły pewne błędy i ostrzeżenia. Czasem mogą być one niezrozumiałe dla początkujących użytkowników, ale zawsze mają jakąś przyczynę. \LaTeX troszczy się bowiem o najdrobniejsze szczegóły i z przejęciem informuje nawet o błahych z pozoru problemach, takich jak wykraczające poza przyjmowaną normę odstępy między wyrazami, związane np. z justowaniem tekstu.

– **Recenzowanie tekstu.** *Microsoft Word*, w przeciwieństwie do \LaTeX a, udostępnia zaawansowane narzędzia komentowania i recenzowania tekstu. Próżno też szukać w \LaTeX u tak rozbudowanych możliwości śledzenia zmian. Nie ma się jednak co dziwić – \TeX nie jest procesorem tekstu, a jedynie narzędziem przetwarzającym pliki źródłowe na dokumenty.

Tych silnych na pozór wad można się jednak pozbyć – ponieważ pliki źródłowe \LaTeX a są plikami tekstowymi, można śledzić zmiany w ich zawartości przy użyciu systemów kontroli wersji, takich jak na przykład *Git*. Te z kolei są o wiele bardziej rozbudowane niż narzędzia dostarczane wraz z *Wordem*.

– **Skomplikowana zmiana stylu.** Zaawansowana zmiana wyglądu dokumentu lub próba stworzenia nowego szablonu wymaga dobrej znajomości \LaTeX a. W zdecydowanej większości przypadków rozwiązania dostępne domyślnie okazują się jednak całkowicie wystarczające.

– **\LaTeX nie jest dla każdego.** Choć dla większości jest to zaleta, dla niektórych może to być zdecydowana wada – \LaTeX nie pozwala w łatwy sposób tworzyć dokumentów nieestetycznych, nieprzemyślanych i o niejasnej strukturze. Nie jest więc to system dla każdego, tym bardziej dla osób, które boją się podejmować wyzwania i uczyć się nowych rzeczy.

W rzeczywistości, ze względu na sposób zapisu plików *Word*a, systemy kontroli wersji nie radzą sobie z analizą wprowadzanych w nich zmian.

Pytania

Pytanie 1. Dlaczego powstał system \TeX ? Co to jest \LaTeX ?

Pytanie 2. Jak wymawiamy nazwy \LaTeX i \TeX ? Z czego to wynika?

Pytanie 3. Co oznacza, że tworząc dokument w \LaTeX u stosujemy znakowanie semantyczne?

Pytanie 4. Jakie przewagi nad programem *Microsoft Word* ma \LaTeX ?

Zaczynamy

Rozpoczęcie pracy z \LaTeX em wymaga od użytkownika podjęcia dwóch zasadniczych decyzji. Pierwsza to wybór odpowiedniej dystrybucji \TeX a, druga zaś to wybór wygodnego środowiska do tworzenia pięknych dokumentów. Celem tego rozdziału jest ułatwienie Czytelnikowi zrozumienia porządku panującego w świecie \LaTeX a. To pozwoli podjąć świadomą decyzję o wyborze odpowiednich narzędzi.

Zanim będziemy mogli się swobodnie poruszać w meandrach występującego na rynku oprogramowania, powinniśmy wyjaśnić sobie niektóre podstawowe pojęcia. Ponieważ jednak nie jest to książka o zapleczu technicznym \LaTeX a, [ale o jego praktycznym wykorzystaniu w tworzeniu dokumentów](#), podarujemy sobie zbędne szczegóły.

Omawianie technicznych aspektów funkcjonowania \LaTeX a byłoby nudne dla autorów i zniechęcające dla Czytelnika. (BP)

Nudne? Niech każdy z autorów mówi za siebie, dobrze? (MB)

Silniki

Silnik to program komputerowy, który przetwarza napisany w języku \TeX plik źródłowy dokumentu na plik wynikowy (np. PDF). Ze względu na odmienne oczekiwania użytkowników dotyczące jego zachowania, powstało kilka silników różniących się od siebie nieznacznie działaniem. Można to porównać do różnych modeli samochodów osobowych – prowadzi się je tak samo, chociaż odpowiadają na różne nasze wymagania.

Obecnie używanych jest pięć silników (kompilatorów) \TeX a:

Nazw *silnik* i *kompilator* będziemy używać zamiennie.

- *TeX* to podstawowy silnik umożliwiający kompilowanie plików źródłowych \TeX a. Generuje pliki w formacie DVI, [który ustąpił już miejsca in-](#)

DVI – ang. *device independent*, czyli *niezależny od urządzenia*.

nym, popularniejszym formatom. Ten i inne powody sprawiły, że jest obecnie używany bardzo rzadko.

- *e-TeX* to kompilator rozszerzający nieznacznie możliwości silnika *TeX*. Nie jest już obecnie powszechnie wykorzystywany, a zastosowane w nim ulepszenia zostały zaimplementowane w innych, nowszych silnikach.
- *pdfTeX* oparty został na silniku *e-TeX*, ale rozszerza go o możliwości związane z generowaniem plików PDF. Tym samym *pdfTeX* może generować zarówno pliki DVI, jak i PDF. Jest to obecnie najpopularniejszy z używanych silników.
- *XeTeX* także został oparty na kompilatorze *e-TeX*, ale wspiera natywnie kodowanie UTF-8 oraz umożliwia dostęp do fontów (krojów pisma) zainstalowanych w systemie operacyjnym. Ta książka została złożona z użyciem silnika *XeTeX*.
- *LuaTeX*, nazywany początkowo drugą wersją *pdfTeX*, wspiera natywnie kodowanie UTF-8, ale też umożliwia wykorzystywanie w pracy języka programowania o nazwie *Lua*, dzięki któremu można (między innymi) uzyskać dostęp do fontów systemowych. Opiera się na silniku *pdfTeX*.

• **natywny** – wbudowany.

• O kodowaniu UTF-8 będziemy pisać w jednym z kolejnych rozdziałów.

Przypomnijmy, że \LaTeX jest, formalnie rzecz biorąc, rozbudowaną nakładką na system składu tekstu o nazwie \TeX . Nie zdziwi więc fakt, że część silników zaadaptowano do obsługi \LaTeX a, tworząc tzw. *formaty* o analogicznych do oryginałów nazwach (np. *pdfLaTeX*, *XeLaTeX* czy *LuaLaTeX*). Wyjątkiem jest jednak format *LaTeX*, który (wbrew pozostałym analogiom) jest nakładką na kompilator *pdfTeX*, a nie *TeX*. Używa go jednak do generowania plików DVI.

• **format** – w tym znaczeniu nakładka na silnik.

• Zainteresowany Czytelnik może się dowiedzieć, że nie istnieje format umożliwiający kompilację dokumentów w obecnej wersji \LaTeX a przy użyciu oryginalnego silnika *TeX*.

Na potrzeby tej książki będziemy zakładać wykorzystanie silnika *pdfTeX* przez format *pdfLaTeX*, uzyskując najszerzej spotykane dziś zestawienie.

Dystrybucje \LaTeX a

Jak już dobrze wiemy, \LaTeX to zbiór usprawnień dla systemu \TeX , ale także platforma, której funkcjonalność można rozbudowywać poprzez rozszerzenia. Rozszerzeń tych są jednak tysiące. Część z nich wykorzystywana jest dość powszechnie, ale niektóre mają niewielkie (często jednocyfrowe) grono

• W chwili pisania tych słów, internetowe archiwum CTAN liczyło 4826 pakietów.

użytkowników. Ze względu na dosyć szeroką gamę zastosowań \LaTeX a, pewne jest jedno – w ciągu całego swojego twórczego życia nie zdążymy odczuć potrzeby korzystania ze zdecydowanej większości z nich.

Żebyśmy mogli używać \LaTeX a bez większych problemów, musimy wyposażać się więc nie tylko w oprogramowanie silników, ale też w podstawowy zbiór rozszerzeń z możliwością jego rozbudowy, aplikacje dodatkowe, wspomagające choćby zarządzanie pakietami, aktualizacje czy tworzenie plików źródłowych, czy w końcu w fonty ¹ wykorzystywane do składania naszych dokumentów. Taki zestaw nazywamy *dystrybucją* systemu \LaTeX .

font – zestaw czcionek o określonych cechach typograficznych.

Jak można się domyślić, nie istnieje jedna, doskonała dystrybucja \LaTeX a. Użytkownicy mają różne wymagania oraz wypracowane przez lata przyzwyczajenia. W chwili obecnej na rynku dostępnych jest więc wiele różnych dystrybucji, możliwych do uruchomienia na niemalże dowolnym komputerze, niezależnie od zainstalowanego systemu operacyjnego. Z łatwością można znaleźć zestawy zarówno bezpłatne, jak i płatne (wzbogacone choćby o dodatkowe wsparcie techniczne czy bardziej zaawansowane aplikacje wspomagające). I choć wybór dystrybucji może mieć wpływ na wygodę zarządzania naszą instalacją \LaTeX a (np. na sposób instalacji dodatkowych pakietów), jedna rzecz jest dla wszystkich dystrybucji wspólna – dostarczają nam one wszystkiego, czego potrzeba do składania tekstów z użyciem tego wspianego systemu.

Proponujemy rozpoczęcie swojej przygody z \LaTeX em od instalacji jednej z najpopularniejszych bezpłatnych dystrybucji: *MiKTeX* (Windows), *TeX Live* (Windows i Linux) albo *MacTeX* (Mac OS).¹

Nie oznacza to, że z czasem odczujemy potrzebę użycia *płatnej* dystrybucji – jest to raczej bardzo mało prawdopodobne.

Edytory oraz IDE

Aby skorzystać z możliwości \LaTeX a, należy przygotować plik źródłowy zawierający opis dokumentu, a następnie (w procesie kompilacji) przekształcić

¹ Na stronach internetowych poszczególnych projektów można znaleźć nie tylko szczegółowe informacje na ich temat, ale także wymagane pliki instalacyjne. Są to w kolejności: <http://miktex.org>, <http://www.tug.org/texlive/> oraz <http://tug.org/mactex/>. W dodatku do elektronicznej wersji tej książki (*Dodatek C: Środowisko pracy*) umieściliśmy także instrukcję instalacji i podstawowej obsługi dystrybucji *MiKTeX* dla systemu Windows.

go w jeden z wyjściowych formatów (obecnie najczęściej PDF). Ze względu na fakt, że pliki źródłowe \LaTeX a są zwykłymi plikami tekstowymi, można je tworzyć i modyfikować z użyciem dowolnego edytora tekstu. Z reguły chcemy jednak, aby pomagał nam on w tym procesie, oferując dodatkowe funkcje, takie jak kolorowanie składni, podpowiedzi, czy też możliwość wywoływania kompilatora z poziomu samego edytora. Piszemy tu o *wywołaniu*, ponieważ kompilatory (silniki) nie są w żaden sposób związane z edytorem tekstu, którego używamy. Możemy więc używać dowolnego programu (a nawet kilku na przemian!), a efekty końcowe będą zawsze takie same.

Bardziej zaawansowane edytory, wyposażone dodatkowo w mechanizmy wspierające tworzenie plików źródłowych, takie jak podpowiadanie nazw poleceń, obsługa błędów i debugowanie, porządkowanie zawartości plików czy opcje równoczesnej pracy nad wieloma dokumentami, bywają określane mianem IDE.

Warto wyposażać się w wygodny edytor (lub IDE), ponieważ przyspiesza i ułatwia to codzienną pracę z \LaTeX em. Można sobie wyobrazić użytkownika, który czerpie satysfakcję z tworzenia swoich dokumentów w *Notatniku*, ale zakładamy, że rozsądni \TeX nicy wolą, gdy im się ułatwia pracę. Dlatego przedstawiamy trzy bezpłatne programy, które nie tylko wydają się być odpowiednie na początku przygody z \LaTeX em, ale też dostępne są w wersjach na systemy *Windows*, *Linux* i *Mac OS*.

- *TeXworks*² to prosty edytor plików źródłowych \LaTeX a dostarczany wraz z dystrybucjami *TeX Live* i *MikTeX*. Oferuje dodatkowo prosty system podpowiedzi. Dla dużej części użytkowników może się okazać wystarczający.
- *TeXmaker*³ to rozbudowane IDE służące do wygodnego tworzenia i zarządzania plikami źródłowymi \LaTeX a. Wśród wygodnych funkcji warto wspomnieć zaawansowane systemy podpowiedzi, zakładki, sprawdzania pisowni oraz obsługi błędów.
- *TeXstudio*⁴ to IDE oparte na *TeXmaker*, wzbogacone o dodatkowe opcje przydatne przy edycji rozbudowanych dokumentów, takie jak wsparcie dla systemów kontroli wersji.

² <http://texworks.org>

³ <http://www.xm1math.net/texmaker/>

⁴ <http://texstudio.sourceforge.net>

Edytor tekstu to program służący do tworzenia i modyfikowania plików tekstowych. Należy być uważnym, aby nie mylić edytorów tekstu (np. *Notatnik*) z procesorami tekstu (takimi jak np. *Microsoft Word*).

IDE – z ang. *integrated development environment*, czyli *zintegrowane środowisko programistyczne*.

\TeX nik (czyt. *technik*) to nazwa użytkowników \TeX a wprowadzona przez Donalda E. Knutha w jego książce pt. *\TeX . Przewodnik użytkownika*.

Wszystkie wymienione edytory oferują kolorowanie składni, wywoływanie kompilatora z poziomu programu oraz szybki podgląd plików wynikowych. Ponieważ wybór odpowiedniego edytora jest kwestią indywidualną (znowu, nie ma jednego doskonałego narzędzia), warto przetestować ich kilka i wybrać ten, który zdaje się być dla nas najbardziej odpowiedni. Nic nie stoi na przeszkodzie, aby na komputerze zainstalować wiele edytorów jednocześnie, porównując dostępne opcje równolegle.

Sprawdźmy, czy działa!

Czas stworzyć pierwszy dokument, który może być dobrym pretekstem do zapoznania się z podstawową obsługą wybranego edytora. Jest to najprostsza procedura, umożliwiająca kompilację dokumentów, w których nie wykorzystujemy zewnętrznego oprogramowania.

Wykonajmy następujące kroki:

1. Utwórzmy w dowolnym miejscu katalog (folder) o nazwie dokument.
2. Stwórzmy nowy plik źródłowy \LaTeX a o treści

```
\documentclass{minimal}

\begin{document}
Co z oczu, to z serca.
\end{document}
```

i zapiszmy go pod nazwą `dokument.tex` w katalogu z poprzedniego punktu. Pliki źródłowe \LaTeX a zapisujemy zawsze z rozszerzeniem `tex`.

3. Ustawmy nazwę kompilatora na `pdfLaTeX` i dokonajmy kompilacji.
4. W katalogu dokument powinny po chwili pojawić się pliki pomocnicze tworzone przez kompilator oraz plik wynikowy dokument.pdf, który możemy podejrzeć w odpowiednim okienku edytora lub przy użyciu dowolnej zewnętrznej przeglądarki plików PDF. Na następnej stronie można zobaczyć (w skali 1:1) fragment wygenerowanego dokumentu.

Choć moim zdaniem, Emacs jest bliżej doskonałości niż wszystkie pozostałe edytory razem wzięte. Użytkownikom nie obawiającym się używania zaawansowanego narzędzia, dającego jednak olbrzymie możliwości, polecam sprawdzenie tego programu. (MB)

W dodatku do elektronicznej wersji tej książki (Dodatek C: Środowisko pracy) zawarliśmy instrukcję instalacji i podstawowej obsługi jednego z edytorów – TeXmaker.

Jeśli w którymś momencie tej książki niezbędne będzie wykonanie dodatkowych czynności, wyraźnie to zaznaczymy.

Użytkownicy, którzy lubią majsterkować, mogą spróbować dokonać kompilacji z poziomu konsoli, wykonując w katalogu dokument polecenie `pdf\latex dokument.tex`.

Co z oczu, to z serca.

Pytania i ćwiczenia

Pytanie 5. Czym różni się silnik, dystrybucja oraz edytor \LaTeX a?

Pytanie 6. Jaka jest różnica między silnikiem a formatem?

Pytanie 7. Czy potrafisz wymienić nazwy kilku silników? A kilku formatów?

Pytanie 8. Przy użyciu jakiego oprogramowania możemy edytować pliki źródłowe \LaTeX a? Wymień kilka programów.

Ćwiczenie 9. Spróbuj skompilować dokument z końca tego rozdziału przy użyciu różnych formatów (*pdfLaTeX*, *XeLaTeX*, *LuaLaTeX*). Jakie występują różnice w efektach? Dlaczego tak jest?

Zrozumieć dokument

Celem tego rozdziału jest oswojenie Czytelnika z plikami źródłowymi dokumentów \LaTeX a. Oczywiście zebrane tu informacje nie wyczerpują tematu, ale pozwolą zrozumieć bez problemu dalszą część książki.

Odstępy, instrukcje i otoczenia

Odstępy

Jedną z zasadniczych różnic między tym, do czego pozwoliły nam przywyknąć godziny używania *Worda*, a tym jak działa \LaTeX , leży w *znakach niewidocznych* (spacjach i znakach tabulacji) oraz znakach nowej linii. Donald E. Knuth, opracowując zasady interpretacji plików źródłowych, musiał podjąć decyzję o tym, jak użycie tych znaków będzie wpływać na ostateczny efekt składu. Tak oto powstał zbiór zasad *a priori*, dotyczących interpretacji takich znaków: *a priori* – łac. z góry.

- każdy znak niewidoczny traktowany jest przez \TeX a jak odstęp,
- kilka odstępów występujących po sobie traktowanych jest jak pojedynczy odstęp,
- odstępy występujące na początku linii są ignorowane, Wyjątkiem od tej reguły jest np. otoczenie `verbatim`.
- wiele następujących bezpośrednio po sobie znaków nowej linii (jedna lub więcej linii przerwy) interpretowanych jest jako zakończenie akapitu,
- pojedynczy znak nowej linii traktowany jest jak odstęp.

Mając na uwadze te zasady, jasne staje się, że oba akapity z zaprezentowanego poniżej dokumentu zostaną zinterpretowane tak samo (choć oczy-

wiście zachęcamy do samodzielnego sprawdzenia tego faktu). Dla ułatwienia oznaczone zostały miejsca, gdzie zastosowano spacje i znaki tabulatora.

```
\documentclass{minimal}

\begin{document}
Co z oczu, to z serca.

    Co z oczu,
to z
serca.
\end{document}
```

Oczywiście sposób pisania dokumentu zaprezentowany w drugim przypadku nie jest zbyt wygodny. Ogólnie przyjęta zasada mówi, że kolejne akapity oddzielamy jednym pustym wierszem, a pierwszego wiersza akapitu nie wcinamy dodatkowo (ani tabulatorem, ani znakami odstępu).

Niektórzy użytkownicy na-
byli dodatkowo zwyczaj
stawiania dwóch znaków
odstępu po kropkach koń-
czących zdanie. Poprawia
to czytelność plików źró-
dłowych.

Instrukcje

Spójrzmy raz jeszcze na plik źródłowy dokumentu, który przygotowaliśmy pod koniec poprzedniego rozdziału.

```
\documentclass{minimal}

\begin{document}
Co z oczu, to z serca.
\end{document}
```

Poza samą treścią dokumentu, w pliku źródłowym umieściliśmy dodatkowe *instrukcje*, dzięki którym \LaTeX wie, w jaki sposób złożyć nasz dokument.

Każdy ciąg małych i wielkich liter alfabetu łacińskiego poprzedzony znakiem odwróconego ukośnika „\” traktowany jest przez kompilator jako instrukcja. W każdym przypadku koniec instrukcji wyznacza dowolny znak inny niż litera (np. spacja, cyfra, przecinek). Tak oto, na przykład, ciąg znaków

`\bambo1eo123` zostanie zinterpretowany jako instrukcja `\bambo1eo` i następujący po niej zestaw cyfr 123.

Argumenty obowiązkowe. Niektóre instrukcje oczekują dodatkowych informacji, zwanych *argumentami*. Na przykład, polecenie `\documentclass`, pozwalające wybrać klasę tworzonego dokumentu, oczekuje jako argumentu nazwy tej klasy. Jest to *argument obowiązkowy* tego polecenia, to znaczy, że nie może zostać pominięty. Argumenty obowiązkowe obejmujemy nawiasami klamrowymi. • **Nawiasy te mają szczególne znaczenie w języku T_EXa – pozwalają tworzyć grupy składające się ze znaków i innych grup⁵.** Dzięki temu wiadomo, gdzie argument się zaczyna i gdzie się kończy. W omawianym przykładzie wywołaliśmy polecenie `\documentclass` z argumentem `minimal`.

Jest całkiem sporo instrukcji, które wymagają więcej niż jednego argumentu obowiązkowego. Na przykład, polecenie `\frac` służące do budowania ułamków w trybie matematycznym, oczekuje dwóch argumentów: wartości licznika i mianownika. Każdy z nich obejmujemy osobną parą nawiasów klamrowych. W takich przypadkach istotna jest oczywiście kolejność, w jakiej przekazujemy parametry.

Argumenty opcjonalne. Wielokrotnie spotkamy się z sytuacją, w której instrukcja – poza argumentami obowiązkowymi – będzie przyjmować *argumenty opcjonalne*. To takie parametry, które nie są wymagane do poprawnego wykonania polecenia, ale mogą być wykorzystane, by wpłynąć na jego działanie. Argumenty opcjonalne obejmujemy nawiasami kwadratowymi i rozdzielamy przecinkami. Umieszczamy je zwykle od razu za nazwą polecenia, a więc jeszcze przed parametrami obowiązkowymi. •

Na przykład, wspomniane już wielokrotnie polecenie `\documentclass` przyjmuje pewne parametry opcjonalne. Jeśli chcielibyśmy zmienić domyśl-

⁵ Zainteresowanych Czytelników zachęcamy do przeczytania rozdziału *Grupowanie* z książki T_EX. *Przewodnik użytkownika* autorstwa Donalda E. Knutha. Z grubsza mówiąc, „grupowanie” oznacza, że fragment zamknięty w nawiasach klamrowych będzie przez T_EXa traktowany jak pojedynczy element. Jeśli polecenie oczekuje jednego argumentu, „wciąga” dokładnie jeden element następujący po nim. Gdybyśmy napisali `\documentclass minimal`, zostałoby to zinterpretowane jako polecenie `\documentclass` z argumentem `mi` następującym dalej ciągiem znaków `nimal`. Wynika to z konstrukcji języka T_EX. Użycie nawiasów klamrowych rozwiązuje ten problem.

`\documentclass`

• Nie zawsze jest to wymagane, ale to przyzwyczajenie pozwoli nam uniknąć wielu niepotrzebnych kłopotów.

`\frac`

• Niektóre polecenia wymagają innej składni – jest to zawsze jasno powiedziane w odpowiednim momencie.

ną wielkość pisma, którym składamy artykuł (klasa `article`) na 12pt, a zastosowany rozmiar papieru na A4, moglibyśmy napisać

```
\documentclass[12pt,a4paper]{article}
```

Lista dostępnych parametrów opcjonalnych jest zawsze wymieniona w dokumentacji pakietu zawierającego dane polecenie.

Odstępy po poleceniach. Jeśli polecenie występuje w tekście i nie przyjmuje żadnych argumentów obowiązkowych, to znak niewidoczny lub znak nowej linii występujący zaraz po tym poleceniu zostanie zignorowany. Traktowany jest bowiem jako informacja, gdzie kończy się nazwa polecenia. Tak oto wypisanie

```
\TeX to super rzecz!
```

spowoduje wyświetlenie tekstu bez spacji pomiędzy słowami `\TeX` oraz „to”. Aby uniknąć takich sytuacji, stosuje się jedną z trzech metod:

- umieszcza się polecenie w nawiasach klamrowych,
- umieszcza się zaraz za poleceniem parę pustych nawiasów klamrowych,
- umieszcza się za poleceniem znak odstępu ręcznie, poprzedzając go symbolem ukośnika.

`\TeX` to polecenie pozwalające wypisać nazwę systemu stylizowanym tekstem.

```
{\TeX} to super rzecz!  
\TeX{} to super rzecz!  
\TeX\ to super rzecz!
```

Nie odważymy się oceniać, która z tych metod jest najlepsza. Warto jednak wybrać jedną z nich i trzymać się jej konsekwentnie.

Otoczenia

Niektóre elementy umieszczane w dokumencie, jak choćby tabele, rysunki czy listy uporządkowane, wymagają zastosowania specyficznych reguł

składu. Należy więc wyraźnie określić, dla jakiego fragmentu pliku źródłowego mają być one stosowane. Wyznaczenie tych granic możliwe jest dzięki zastosowaniu *otoczeń*.


Otoczenia rozpoczynają się zwykle od polecenia `\begin`, a kończą poleceniem `\end`. Obie te instrukcje przyjmują co najmniej jeden argument obowiązkowy (nazwę otoczenia). Niektóre otoczenia wymagają jednak podania dodatkowych argumentów.

Już kilkakrotnie (choć dotychczas nieświadomie) spotkaliśmy się z otoczeniami w praktyce. Abyśmy mogli skompilować jakikolwiek dokument \LaTeX a, musi on zawierać otoczenie o nazwie `document`, wewnątrz którego umieszczamy wszystko to, co ma się w dokumencie znaleźć. To otoczenie ma więc charakter szczególny.

Klasy dokumentów i pakiety

Klasy dokumentów

Właściwie każdy dokument, który tworzymy, da się jakoś sklasyfikować. Gdy siadamy do pisania artykułu, wiemy, że piszemy artykuł, jeszcze zanim wystukamy na klawiaturze pierwsze słowo. Podobnie jest zresztą z listem, książką, raportem czy jakąkolwiek inną formą pisaną.

Zależnie od tego, co właściwie piszemy, zmieniają się (a przynajmniej powinny) nasze oczekiwania dotyczące wyglądu dokumentu. W przypadku książki chcielibyśmy, aby możliwe było tworzenie rozdziałów, a na jej początku znajdowała się wzniosła strona tytułowa, podczas gdy w przypadku artykułu wystarczy krótki nagłówek na pierwszej stronie tekstu. Podobnie zresztą, ponieważ książki dzielimy na dość długie rozdziały, chcielibyśmy, aby w nagłówku  każdej strony znajdował się tytuł tego, który aktualnie czytamy. W artykule takie wzbogacanie strony byłoby co najmniej niepożądane.

Dziesiątki, jeśli nie setki podobnych różnic wynikają z wielu lat doświadczenia w dziedzinie typografii i składu tekstów drukowanych i wyświetlanych. Lata te przyniosły efekt w postaci wypracowanych standardów, które powinny być stosowane przez każdego, kto przygotowuje książkę do druku. Aby dać autorom możliwość tworzenia dokumentów zgodnych z tymi zasadami,

Nagłówek (lub stopka), którego zawartość zmienia się w zależności od rozdziału lub paragrafu nazywamy *żywą paginą*. Temat żywej paginy jest rozszerzony w *Recepturach*.

nie zmuszając ich dodatkowo do samodzielnego zastanawiania się nad rzeczami, o których czasem nawet najlepsi specjaliści zapominają, wprowadzono klasy dokumentów. Klasy te to zbiory szczegółowych i uzasadnionych wytycznych dla \LaTeX a, opisujących zasady składania tekstu i innych elementów występujących w dokumencie. Ponieważ różnią się one w zależności od tego, jaki typ dokumentu składamy, naturalnie powstało wiele różnych klas, z których możemy korzystać.

Klasy te zawierają nierzadko, poza wytycznymi w zakresie typografii, wiele modyfikacji wpływających na wygląd finalnego dokumentu. To właśnie tak wyraźne oddzielenie warstwy wizualnej i treści dokumentu sprawia, że wiele wydawnictw dostarcza autorom własne, autorskie klasy, z których mogą (i powinni) oni korzystać. Istnieją też klasy specjalistyczne, dostarczane wraz z niektórymi pakietami, a pozwalające choćby tworzyć prezentacje multimedialne (`beamer`) czy postery (`tikzposter`).

Wraz z \LaTeX em otrzymujemy obligatoryjnie dostęp do dziesięciu klas, z których możemy korzystać. Niektóre z nich mają charakter czysto techniczny (np. służą do składania dokumentacji tworzonych pakietów), nie ma więc potrzeby ich omawiać. Pozostałe opisaliśmy krótko w tabeli 1.

Tabela 1. Niektóre klasy dostarczane z \LaTeX em.

Nazwa klasy	Opis
<code>minimal</code>	Nie implementuje żadnych dodatkowych zasad typografii, dlatego powinna być wykorzystywana tylko w celach testowych.
<code>article</code>	Służy do składania artykułów naukowych i krótkich raportów (bez podziału na rozdziały).
<code>book</code>	Wprowadza zasady typowe dla składu długich opracowań (książek), podzielonych na rozdziały.
<code>report</code>	Służy do tworzenia dłuższych raportów z podziałem na rozdziały i paragrafy, ale też krótkich książek.
<code>letter</code>	Wykorzystywana jest do pisania listów.
<code>slides</code>	Klasa ta zmienia domyślny font na bezszeryfowy i dużo większy niż zwykle. Dzięki temu nadaje się do tworzenia wyraźnych slajdów.

Warto zauważyć, że powyższe klasy przystosowane są do tworzenia dokumentów w języku angielskim – implementują one zasady składu tekstu charakterystyczne dla krajów anglosaskich. Gdybyśmy chcieli pisać w języku polskim, powinniśmy zadbać o zastosowanie zasad odpowiednich dla języka polskiego. Temu tematowi poświęciliśmy w całości kolejny rozdział.

O ile jednak klasy dokumentów pozwalają nam zdjąć z siebie odpowiedzialność za większość aspektów wyglądu i składu, o tyle same w sobie ograniczają się jedynie do nich, [nie dostarczając nam dodatkowych funkcji](#). Tę rolę przejmują pakiety.

Niektóre klasy dostarczają dodatkowe funkcje. Przykładem może być polecenie `\chapter` dostępne w klasie `book`, lecz niedostępne w klasie `article`.

Pakiety

Wcześniejsze rozważania wyrobiły w nas dość intuicyjne podejście do pojęcia pakietu, jako zbioru rozszerzeń możliwości \LaTeX a. Bardziej formalnie, *pakiet* to zbiór dodatkowych funkcjonalności, z których możemy korzystać w ramach dokumentu. Może on definiować nowe instrukcje i otoczenia, modyfikować zasadę działania tych, które już istnieją, ale też (choćby poprzez stosowanie niskopoziomowych poleceń \TeX a) wpływać na sposób, w jaki dokument zostanie przetworzony przez kompilator.

Zastosowania pakietów ograniczone są jedynie naszą wyobraźnią, a potrzeba jest matką wynalazku, zatem nie dziwi fakt, że obszerna już baza dostępnych rozszerzeń wciąż się powiększa. Dają nam one całą gamę możliwości: od modyfikacji parametrów strony (pakiet `geometry`), poprzez polecenia wspomagające tworzenie interaktywnych plików pdf (`hyperref`), listinżów (`listings`), po specjalistyczne pakiety do składu wzorów matematycznych (większość pakietów o nazwach rozpoczynających się od `ams`).

Wraz z każdą dystrybucją \LaTeX a otrzymujemy dostęp do pewnego podstawowego zbioru pakietów. Niektóre z nich są częścią samego \LaTeX a, inne dołączane są do niego zawsze, jako absolutne minimum potrzebne do pracy. Pozostałe możemy pobrać z internetu w dowolnym momencie.

Pakiet `lipsum`

W przykładach w dalszej części książki będziemy wykorzystywać fragmenty słynnego *Lorem ipsum*, tekstu przypominającego nieco łacinę, ale nie

niosącego ze sobą żadnej treści. Dzięki temu projektant i odbiorca makiety nie koncentrują się na tekście – bo ten i tak nie ma sensu – ale na wyglądzie całości. W przypadku, gdyby miało to znacząco wydłużyć przykłady, skorzystamy ze specjalnego pakietu `\lipsum`, pozwalającego wstawić nawet kilkadziesiąt akapitów bezsensownego tekstu jednym poleceniem.

Makieta w poligrafii nazywa się szablon dokumentu, na którym można pokazać np. rozmieszczenie elementów strony czy wybrane kroje pisma.

Nie polecamy jednak jego użycia przy pisaniu pracy dyplomowej.

Plik źródłowy i jego konstrukcja

Aby plik źródłowy został poprawnie zinterpretowany, musi mieć określoną strukturę. Ponieważ dokumenty \LaTeX są czytane są liniowo (znak po znaku), ważne jest, aby plik źródłowy rozpoczynał się od deklaracji, które pozwolą \LaTeX owi zrozumieć wszystkie zasady składu, jeszcze zanim umieści na kartce pierwszy znak. Musimy więc napisać, na jakiej klasie dokumentu będziemy się opierać oraz z jakich pakietów będziemy korzystać, a jeśli będzie taka potrzeba – również jakie zasady modyfikujemy. Część dokumentu, która dostarcza te informacje, nazywamy *preambułą*.

Preambuła rozpoczyna się od instrukcji `\documentclass`. Strukturę tego polecenia poznaliśmy już wcześniej. Jest ono obowiązkowe – pominięcie go lub umieszczenie w złym miejscu spowoduje błąd kompilacji.

Dopiero po deklaracji klasy możemy dołączać pakiety, z których będziemy korzystać. Wykorzystujemy do tego polecenie `\usepackage`. Jedynym argumentem obowiązkowym tego polecenia jest nazwa dołączanego pakietu. Niektóre pakiety przyjmują jednak dodatkowo argumenty opcjonalne. Na przykład, gdybyśmy chcieli wykorzystać pakiet `inputenc` z parametrem `utf8`, wpisalibyśmy

```
\usepackage[utf8]{inputenc}
```

`\documentclass`

`\usepackage`

Znaczenie pakietów wykorzystanych w tym paragrafie nie jest istotne. Wszystkie one zostaną omówione w dalszej części książki.

Proste, prawda? Jeśli wczytujemy wiele pakietów, a żadnemu z nich nie przekazujemy parametrów opcjonalnych, możemy je dołączyć jednocześnie, oddzielając ich nazwy przecinkami, na przykład

```
\usepackage{amsmath,amsfonts}
```

Kolejność wczytywania pakietów ma istotne znaczenie. Przykładowo, jeśli dołączymy dwa pakiety definiujące polecenie o tej samej nazwie, to w znaczącej części wypadków obowiązywać będzie definicja z pakietu dołączonego później. Zwykle jednak kompilator ostrzeże nas o takiej sytuacji.

Przykładowa preambuła może więc wyglądać w następujący sposób:

```
\documentclass[12pt,a4paper]{article}

\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
```

Druga część dokumentu nazywana jest częścią właściwą. Jej granice określa otoczenie `document`, to znaczy, że rozpoczyna się ona wraz z poleceniem `\begin{document}`, a kończy się z poleceniem `\end{document}`. Wszystko to, co znajduje się w pliku źródłowym za środowiskiem `document`, zostanie przez kompilator zignorowane.

Otoczenie `document` musi wystąpić w pliku źródłowym. Co więcej, cała treść dokumentu powinna znajdować się w jego wnętrzu. Gdy tak nie będzie, próba kompilacji pliku źródłowego najprawdopodobniej skończy się błędem.

Otoczenie nazywa się inaczej *środowiskiem*. W dalszej części książki będziemy tych nazw używać zamiennie.

Komentarze i znaki specjalne

Komentarze

Czasem zachodzi potrzeba wprowadzenia w pliku źródłowym tekstu, który nie powinien być przez kompilator interpretowany. Taką własność daje nam symbol `%`. Jeśli użyjemy go w pliku źródłowym, to wszystkie znaki znajdujące się za nim, najbliższy znak nowego wiersza i znaki niewidoczne występujące na początku następnej linii zostaną zignorowane.

Znak `%` jest, z uwagi na powyższe własności, wykorzystywany czasami w celu dzielenia długich linii np. tam, gdzie nie chcemy użyć znaku nowego

wiersza czy odstępu. Przykładowo, oba akapity z poniższego przykładu zostaną wyświetlone identycznie.

```
\documentclass{minimal}

\begin{document}
Cozoczu, tozserca.

Cozoczu, t%
 ozserca.
\end{document}
```

Znaki specjalne

Zapewne stało się już oczywiste, że niektóre symbole mają dla \LaTeX a szczególne znaczenie. Są to na przykład znak odwróconego ukośnika rozpoczynający każde polecenie, nawiasy klamrowe, ale też – o czym przekonaliśmy się przed chwilą – znak procenta. Takie znaki nazywamy *symbolami specjalnymi*. Oto ich pełna lista:

`\ { } # % $ & _ ~ ^`

We wcześniejszej części tego rozdziału pisaliśmy, że instrukcja \LaTeX a to ciąg małych i wielkich liter alfabetu łacińskiego poprzedzony znakiem odwróconego ukośnika. Cóż, nie jest to cała prawda. Istnieje druga grupa instrukcji – jeden znak *niebędący* literą, poprzedzony znakiem odwróconego ukośnika. Pewien podzbiór tych poleceń pozwala na wykorzystanie wymienionych wyżej znaków w tekście. Aby to zrobić, wystarczy poprzedzić je odwróconym ukośnikiem. Tak oto polecenie `\{` drukuje nawias klamrowy otwierający, `\}` nawias klamrowy zamykający itd. Od tej reguły występują jednak wyjątki:

- Polecenie `\\` jest poleceniem służącym do ręcznego przejścia do nowej linii wewnątrz akapitu. Aby wypisać w tekście symbol odwróconego ukośnika, należy użyć polecenia `\textbackslash`. Nie jest to najwygodniejsze, ale też nie jest to często wykorzystywany znak.

`\textbackslash`

- Polecenia `\~` i `\^` mają dodatkowe znaczenie typograficzne – wprowadzają akcent nad literą następującą bezpośrednio za nim. ↑ Dlatego, aby uniknąć przypadkowych niedogodności, warto wykorzystać znaną nam już metodę i bezpośrednio za poleceniem umieścić pustą parę nawiasów klamrowych, tworząc odpowiednio `\~{ }` i `\^{ }`. W praktyce oznacza to wstawienie akcentu przy pustym znaku, co sprowadza się do efektu, który chcemy uzyskać.

Więcej informacji na temat akcentów Czytelnik znajdzie w drugiej części książki.

Kodowanie pliku źródłowego i fontu

Plik źródłowy \LaTeX jest zwykłym plikiem tekstowym, zapisanym w pliku z rozszerzeniem `tex`. W rozumieniu komputera jest to jednak długi ciąg bitów układających się w bajty. ↑ To właśnie kolejne bajty z pliku interpretowane są jako liczby (od 0 do 255), do których z kolei przypisane są znaki i symbole. Umowny sposób, w jaki to przypisanie zachodzi, nazywamy *kodowaniem*.

bit – najmniejsza jednostka danych zapisywanych w komputerze: zero lub jeden.

bajt – ciąg 8 bitów.

Krótką historia kodowań

Kiedy komputery zyskiwały popularność, a jedynym sposobem interakcji z użytkownikiem był tryb tekstowy, zaszła potrzeba usystematyzowania sposobu, w jaki maszyna przekształca ciągi bitów w znaki wyświetlane na ekranie. Powstał więc standard ASCII, który w postaci tabeli jasno prezentował, jak ciągi 7 bitów (liczby z przedziału od 0 do 127) mają być interpretowane. Tak oto, dla przykładu, ciąg `1110100` ↑ interpretowany był jako mała litera „t”. Dzięki takiemu podejściu, możliwe było uzyskanie znaków sterujących (takich jak znak końca pliku), znaków specjalnych (takich jak nawiasy i cudzysłowy), cyfr oraz małych i wielkich liter alfabetu łacińskiego.

`1110100` w systemie dwójkowym równe jest 116 w systemie dziesiętnym.

W związku z rozwojem technologicznym i przejściem z procesorów 18- i 36-bitowych na 8-, 16- i 32-bitowe, zaszła potrzeba i możliwość rozszerzenia standardowej tabeli ASCII o dodatkowe symbole. Tak oto stosowane dotąd kody 7-bitowe zostały zastąpione kodami długości bajta, rozszerzając tym samym dwukrotnie zbiór możliwych do uzyskania znaków i zachowując kompatybilność wstecz. ↑ Znaki z klasycznego zestawu zajęły w tabeli ASCII

Kompatybilność polegała na tym, że kody znaków z tabeli ASCII poprzedzone zostały zerem, wypełniając pierwszą połowę nowej tabeli.

miejsca 0–127, a dodatkowe symbole przypisane zostały do dalszych kodów, to znaczy 128–255.

Szybko okazało się, że istnieje zapotrzebowanie na wykorzystanie znaków specjalnych i diakrytycznych, charakterystycznych dla języków występujących w poszczególnych częściach świata. Zbiór 256 znaków nie mógł jednak wyczerpać potrzeb wszystkich narodów. Powstało więc wiele odmian kodowania ASCII, przypisujących bajty o wartościach 128–255 do oczekiwanych symboli. Jednym z tak powstałych kodowań było kodowanie cp1250 (nazywane też *windows-1250*), które pozwalało poprawnie interpretować polskie znaki diakrytyczne. Do dzisiaj w polskojęzycznych wersjach systemu *Windows* pod pojęciem kodowania ASCII rozumie się kodowanie cp1250.

Rozwój interfejsów stosowanych w systemach operacyjnych wymusił stworzenie innych standardów, bardziej dostosowanych do wymagań użytkowników. Tak powstał cały zbiór standardów ISO, oferujących różne zestawy znaków. Dla krajów Europy Zachodniej (a więc i Polski) zastosowanie ma kodowanie ISO-8859-2. Jest ono zgodne ze standardem ASCII w tym sensie, że bajty o wartościach 0–127 interpretowane są tak samo, jak w podstawowym standardzie.

Istnienie wielu różnych kodowań stało się problematyczne, nie tylko ze względu na to, że wymagało świadomego wyboru tego stosowanego w dokumentach, ale także posiadania fontów, które znaki dostępne w danym kodowaniu potrafią skutecznie dostarczyć. To ograniczenie stało się uciążliwe zarówno dla użytkowników komputerów, jak i autorów fontów. Naturalna więc stała się idea powstania jednego, uniwersalnego standardu, [↑] zdolnego obsłużyć wszystkie symbole stosowane na świecie. Jako zbawienie przyszło kodowanie UTF-8, umożliwiające spójne i jednoznaczne przyporządkowanie do ciągów bitów aż 1 112 064 różnych znaków i symboli. Jego konstrukcja nie jest tak prosta, jak klasycznych kodowań wykorzystujących do zapisu znaku pojedyncze bajty, dlatego pominiemy jej opis⁶. Kodowanie UTF-8 zapewnia jednak kompatybilność wstecz z podstawową tabelą ASCII, a dzięki swojej bezsprzecznej uniwersalności staje się popularniejszy bez przerwy od lat. Być może za kilka lat nie będzie się już korzystać z niczego innego.

jeden, by wszystkimi rządzić...

⁶ Zainteresowanych zapraszamy do bliższego zapoznania się z tym standardem, choćby na stronie Wikipedii: <http://en.wikipedia.org/wiki/UTF-8>.

Dzisiaj większość edytorów (w tym wszystkie edytory \LaTeX a zaprezentowane w poprzednim rozdziale) pozwala tworzyć dokumenty kodowane przy użyciu UTF-8. Co więcej, w znacznej części przypadków jest to kodowanie domyślne. Mniej zaawansowanym użytkownikom radzimy tego nie zmieniać.

Kodowanie pliku źródłowego

Gdy silnik \LaTeX a interpretuje plik źródłowy, powinien wiedzieć, w jaki sposób zakodowane są znaki w nim wykorzystane. Dopóki obracamy się w zbiorze symboli z pierwszej połowy tabeli ASCII, żadne dodatkowe działania nie są konieczne. [Gdy jednak silnik trafi na bajt o wartości powyżej 127, powinien wiedzieć, według jakiego kodowania ma go interpretować.](#) Taką informację możemy (i powinniśmy) mu dostarczyć poprzez dołączenie do naszego dokumentu pakietu `inputenc` z odpowiednim parametrem opcjonalnym. Aby to zrobić, umieszczamy w preambule polecenie:

```
\usepackage[kodowanie]{inputenc}
```

Oczywiście słowo `kodowanie` powinniśmy zastąpić odpowiednim oznaczeniem, zgodnym z rzeczywistym kodowaniem pliku. Może to być `cp1250`, `latin2` (dla kodowania ISO-8859-2) lub `utf8` (dla UTF-8). W znaczącej większości wypadków odpowiednim wyborem będzie ostatni.

Kodowanie fontu

Kiedy \LaTeX wie już, dzięki poprawnej deklaracji kodowania pliku źródłowego, jakie znaki wprowadził użytkownik, następuje konieczność umieszczenia ich w wyjściowym dokumencie (zwykle w formacie PDF). W tym celu \LaTeX sięga do plików z fontami, by odszukać w nich pożądane znaki. I, podobnie jak w przypadku pliku źródłowego, stosuje w tym celu odpowiednie kodowanie. Standardowo \LaTeX wykorzystuje 7-bitowe fonty zapisane w oparciu o układ OT1. Daje to zbiór 128 symboli (symboli specjalnych, liter, cyfr i akcentów) pod wspólną nazwą *Computer Modern* (w skrócie CM). Jeśli zachodzi potrzeba wypisania znaku diakrytycznego, następuje to poprzez nałożenie na siebie odpowiedniej litery i akcentu. [Nie jest to rozwiązanie doskonałe – powodu-](#)

Wynika to wprost z tego, że każde kodowanie jest zgodne z podstawową tabelą ASCII – jego wybór nie wpływa na interpretację znaków o kodach między 0 a 127.

Zapisywanie znaków diakrytycznych występujących w języku polskim nie jest możliwe przy użyciu znaków z układu OT1

je bowiem parę komplikacji technicznych związanych m.in. z przenoszeniem wyrazów – a efekty nie zawsze są typograficznie zadowalające.

Na powyższy problem wynaleziono jednak lekarstwo. Aktualne dystrybucje \LaTeX a wyposażone są standardowo w zestaw krojów z rodziny *European Computer Modern* (EM), który stosuje 8-bitowe kodowanie T1. [Występuje tu pewna analogia do kodowań ASCII i ISO-8859-2](#) – pierwsza połowa znaków w tabeli fontów z grupy EM pokrywa się (w większości) z symbolami z rodziny CM, druga zaś wypełniona jest europejskimi znakami specjalnymi. Dzięki temu możliwe jest stosowanie *prawdziwych* polskich znaków diakrytycznych i znaków interpunkcyjnych zgodnych z polskimi normami typograficznymi (np. cudzysłówów). Aby zadeklarować korzystanie z krojów z rodziny EC, wystarczy skorzystać z pakietu `fontenc`, umieszczając w preambule następujący kod:

```
\usepackage[T1]{fontenc}
```

Podobnie jak w przypadku kodowań plików źródłowych, istnieje wiele kodowań fontów. Dokładniejsze ich omawianie mija się jednak z celem, jako że znajomość powyższych (OT1 oraz T1) wystarcza w zupełności do skutecznego składania dokumentów zarówno w języku angielskim, jak i w innych językach europejskich. Czytelników zainteresowanych zgłębieniem swojej wiedzy w tej tematyce zachęcamy jednak do sięgnięcia do literatury fachowej.

Pliki produkowane przez \LaTeX a

Za każdym razem, gdy dokonamy kompilacji dokumentu \LaTeX a, poza plikiem wynikowym mogą zostać utworzone inne pliki pomocnicze. I chociaż wydawać by się mogło, że ich obecność jest absolutnie niepotrzebna, to w rzeczywistości \LaTeX wykorzystuje je do poprawnego generowania dokumentów. Aby to zrozumieć, należy poznać przeznaczenie tych plików.⁷

⁷ Przedstawione w tabeli zestawienie obejmuje większość plików, z którymi Czytelnik może się spotkać, ale nie wszystkie. W zależności od uwzględnionego dodatkowego oprogramowania, lista ta mogłaby rozszerzyć się nawet o kilkanaście pozycji.

Warto tu wspomnieć jeszcze o rodzinie krojów *Latin Modern*, która wykształciła się z rodziny CM i oferuje pełne wsparcie dla polskich znaków diakrytycznych.

Rozszerzenie	Opis
.log	W tym pliku zapisywane są wszystkie informacje dotyczące przebiegu kompilacji, w tym informacje o ostrzeżeniach i błędach.
.aux	Jest to szczególny plik. Zapisywane są w nim informacje, które mogą zostać wykorzystane po wywołaniu kolejnej kompilacji. Przykładowo, gdy stosujemy w dokumencie odsyłacze do miejsc znajdujących się dalej, to podczas kompilacji w pliku .aux zbierane są informacje o położeniu elementów, do których się odwołujemy, by po ponownym jej wywołaniu mogły zostać wykorzystane.
.toc	W tym pliku zapisywane są informacje o nagłówkach, ich numeracji i położeniu w dokumencie. W czasie kolejnej kompilacji dane te są wykorzystywane do wygenerowania spisu treści.
.lof	Służy do przechowywania informacji o ilustracjach i ich położeniu w dokumencie. W czasie kolejnej kompilacji dane te są wykorzystywane do wygenerowania listy ilustracji.
.lot	Plik ten ma funkcję zbliżoną do pliku .lof, ale dotyczy występujących w dokumencie tabel.
.idx	Jeśli w dokumencie występują hasła, które powinny być gromadzone w indeksie (skorowidzu), to \LaTeX wygeneruje ten plik w czasie kompilacji. Zawiera on listę wszystkich haseł występujących w dokumencie. Aby przetworzyć plik .idx na plik .ind, który może zostać dołączony do pliku wynikowego, należy użyć programu <code>makeindex</code> lub <code>xindy</code> (dla indeksów z hasłami zawierającymi znaki zakodowane w UTF-8).
.ind	Wynik działania programu <code>makeindex</code> lub <code>xindy</code> . Jeśli ten plik istnieje, zostanie wykorzystany w czasie następnej kompilacji dokumentu.

<code>.ilg</code>	Plik <code>.ilg</code> ma przeznaczenie identyczne, jak plik <code>.log</code> , ale dotyczy pracy programu <code>makeindex</code> lub <code>xindy</code> .
<code>.synctex.gz</code>	Plik ten może być tworzony w procesie kompilacji dokumentu, jeśli kompilator wywoływany jest z poziomu edytora. Dzięki temu edytor może umożliwić użytkownikom przechodzenie do wybranych fragmentów kodu źródłowego dokumentu poprzez kliknięcie w odpowiednim miejscu podglądu pliku PDF (i odwrotnie).

Jak wynika wprost z powyższej tabeli, niektóre operacje (takie jak generowanie spisów treści) mogą wymagać co najmniej dwukrotnego skompilowania dokumentu, a inne (na przykład generowanie skorowidzu) nawet użycia dodatkowego programu pomiędzy jedną a drugą kompilacją. Wynika to z faktu, że \LaTeX przetwarza plik liniowo – niemożliwe jest więc cofnięcie się w celu uzupełnienia danych w już wygenerowanej wcześniej części dokumentu. Na szczęście wiele edytorów \LaTeX a pozwala na automatyzację procesu kompilacji tak, aby konieczność dokładnego rozumienia tych procesów była zbędna.

Pytania i ćwiczenia

Pytanie 10. Obejrzyj poniższy kod źródłowy pliku \LaTeX -a.

```
\documentclass[10pt,a4paper,twoside]{article}

\usepackage[utf8]{inputenc}

\begin{document}
\section{Płacek}

Co \textbf{z} oczu,\\\textit{t}o \textsl{z }serca.

\begin{center}
Ciemnota -- gorsza od zbrodni.
```

```
\end{center}  
\end{document}
```

Wskaż wszystkie wykorzystane polecenia i otoczenia oraz ich parametry obowiązkowe i opcjonalne. Jakiej klasy użyliśmy w tym dokumencie? Jakie pakiety zaimportowaliśmy? Gdzie zaczyna się i kończy preambuła? A część właściwa? Spróbuj skompilować ten dokument. Zapoznaj się z efektami.

Ćwiczenie 11. Spróbuj skompilować poniższy dokument, zmieniając nazwę klasy na jedną z zestawu: `book`, `article`, `report`, `letter`. Nie przejmuj się, jeśli nie rozumiesz wykorzystanych poleceń.

```
\documentclass{article}  
  
\title{Dokument}  
\author{Nazwisko autora}  
  
\begin{document}  
\maketitle  
  
Co z oczu, to z serca.  
\end{document}
```

Jakie widzisz różnice? Dlaczego są one dostrzegalne mimo braku zmian we właściwej treści dokumentu?

Ćwiczenie 12. Spróbuj skompilować poniższy dokument:

```
\documentclass{book}  
  
\begin{document}  
\chapter{Moja pierwsza książka}  
  
Witaj, świecie.  
\end{document}
```


Jak myślisz, dlaczego się nie udało? Spróbuj naprawić ten problem (*podpowiedź*: niektóre użyte litery nie występują w podstawowej tabeli ASCII). Zadeklaruj wykorzystanie fontów kodowanych przy użyciu T1. Dlaczego warto to zrobić?

Ćwiczenie 13. Zapoznaj się z zawartością pliku `.log` powstałego w wyniku kompilacji dokumentu z *ćwiczenia 12*. Porównaj rezultaty przed i po wprowadzeniu poprawek.

Dokumenty w języku polskim

Wcześniejsze rozważania pozwoliły nam wysnuć pewien wniosek: o ile \LaTeX dostarcza nam klasy pozwalające składać różnego rodzaju dokumenty (przypomnijmy sobie klasy `article`, `report`, czy `book`), o tyle uzyskane dzięki nim efekty będą zwykle zgodne z anglosaskimi normami składu tekstów. Będzie tak nawet, jeśli zadeklarujemy wykorzystanie znaków kodowanych przy użyciu UTF-8 lub fontów z układu T1.

Gdy tworzymy dokumenty w języku angielskim, możemy uznać za wygodny fakt, że tak się dzieje. W przypadku tekstów pisanych po polsku, powinniśmy jednak oczekiwać, że efekt naszej pracy będzie zgodny z wypracowanymi przez lata lokalnymi regułami. I rzeczywiście, zasady składania tekstów w języku polskim zostały zebrane w normach branżowych, takich jak chociażby *BN-76/7440-02 Zasady składania tekstów w języku polskim*, czy *BN-76/7440-03 Zasady formowania kolumn książek, broszur i czasopism*.

Wspomniane dokumenty to wielostronicowe zestawy wytycznych obejmujących aspekty takie jak numerowanie tytułów, składanie wyliczeń, paragrafów, pagin (nagłówek i stopek), ale też łamanie wierszy. Okazuje się, że wiele reguł obowiązujących w języku polskim różni się od analogonów w języku angielskim. Wśród elementarnych przykładów, dzięki którym łatwo uświadomić sobie znaczenie tych różnic, można wymienić choćby poniższe.

- **Rozmiar papieru.** W Stanach Zjednoczonych domyślnym formatem papieru jest *US Letter*. W Polsce stosujemy formaty metryczne, najczęściej serii A i B, jak na przykład A4 czy B5.
- **Numerowanie nagłówek.** W języku angielskim po numerze porządkowym występującym w tytule nie stawia się kropki – w języku polskim tak.

- **Wcięcie akapitowe.** W anglosaskim zwyczaju jest, aby pierwszy akapit tekstu nie był wcięty. Polskie teksty zwykły się do niego nie stosować.
- **Sposób składania tytułów.** Przykładowo, w nagłówku artykułu w języku angielskim nazwisko autora umieszcza się pod tytułem – w języku polskim nad (o ile nie poprzedza ono np. adresu).

Zasad tych są dziesiątki – ciężko więc oczekiwać, że ktokolwiek je zapamięta, nie mówiąc już o konsekwentnym stosowaniu się do nich. Tutaj uaktywnia się znacząca przewaga \LaTeX a nad alternatywnymi systemami składu tekstu. \LaTeX umożliwi nam bowiem dość niewielkim kosztem upewnienie się, że nasze dokumenty będą (na tyle, na ile to możliwe) zgodne z obowiązującymi w Polsce normami. Jest to w dużej części możliwe dzięki dwóm rozszerzeniom, które omawiamy poniżej.

Pakiet polski

Z rozdziału dotyczącego pakietów dowiedzieliśmy się, że niektóre z nich mogą nie tylko modyfikować zasady działania już istniejących instrukcji \LaTeX a, ale też wpływać na sposób, w jaki interpretowany jest dokument. Jednym z pakietów, które tak działają jest pakiet o nazwie `polski`. Udostępnia on nie tylko dodatkowe kroje pisma z rodziny CM, zawierające polskie znaki diakrytyczne (pamiętamy, że fonty CM standardowo takich znaków nie posiadają), ale także pozwala \LaTeX owi poprawnie dzielić polskie wyrazy, gdy zaistnieje potrzeba złamania wiersza.

Czytelnik, który wykonał polecenia z *ćwiczenia 12* na końcu poprzedniego rozdziału, zauważył zapewne, że w otrzymanym dokumencie pojawiło się angielskie słowo „chapter”. Otóż, poza wyżej wymienionymi, raczej mało interesującymi dla przeciętnego użytkownika własnościami, pakiet `polski` modyfikuje domyślne (angielskie) nazwy elementów występujących w dokumencie. Dzięki jego użyciu hasła takie jak choćby „chapter”, „figure” czy „table of contents” zmieniają się na odpowiadające im w języku polskim „rozdział”, „rysunek” oraz „spis treści”.

Biorąc to wszystko pod uwagę, wydaje się więc rozsądne dodać omawiany pakiet do dokumentów pisanych w języku polskim. Robimy to (klasycznie już) dołączając do preambuły wiersz:

Pakiet `polski` przyjmuje pewne parametry opcjonalne, które pozwalają wpłynąć na jego funkcjonowanie. Nie będziemy ich jednak omawiać, uznając je za zbyt zaawansowane.

```
\usepackage{polski}
```

Użycie pakietu `polski` zwalnia nas z dołączania w preambule dokumentu rozszerzenia `fontenc` omówionego w poprzednim rozdziale. Jest tak dlatego, że jednym z zadań pakietu `polski` jest zapewnienie doboru odpowiednich fontów przy generowaniu pliku wynikowego.

Zestaw klas `mwc\ls`

Pakiet `polski` dostarcza \LaTeX owi informacje o tym, jakich słów używać i jak poprawnie zachowywać się wędrując między tymi wprowadzonymi przez polskiego użytkownika. Brakującym ogniwem w drodze do pięknych dokumentów w naszym ojczystym języku zdaje się być jedynie zbiór reguł składu tekstu, określonych w polskich normach.

Zainstalowanie pakietu `mwc\ls` [↑](#) pozwala nam na korzystanie w naszych dokumentach ze zmodyfikowanych klas, służących do składu tekstów w języku polskim. Występuje przy tym następująca zależność:

Tabela 3. Odpowiedniki standardowych klas do składu dokumentów w języku polskim.

Typ dokumentu	Klasa standardowa	Klasa z pakietu <code>mwc\ls</code>
artykuł	<code>article</code>	<code>mwart</code>
książka	<code>book</code>	<code>mwbk</code>
raport	<code>report</code>	<code>mwrep</code>

Klasy wymienione w trzeciej kolumnie wprowadzają szereg usprawnień, począwszy od zmiany domyślnego rozmiaru papieru z *US Letter* na *A4*, a na subtelnym modyfikacjach rozmiaru fontu wykorzystywanego w tytułach paragrafów skończywszy. Stosowanie ich zdaje się być dobrym gestem szacunku wobec polskiej typografii.

Pakiet `mwc\ls` autorstwa Marcina Wolińskiego jest jednym z pakietów, które rozszerzają możliwości \LaTeX a poprzez dostarczenie mu dodatkowych klas dokumentów. Zestawu `mwc\ls` nie dołącza się do dokumentu przy użyciu polecenia `\usepackage`.

Przykładowy dokument

Poniżej prezentujemy dość krótki plik źródłowy \LaTeX a, będący podstawą do stworzenia artykułu zgodnego z polskimi normami typograficznymi. Może być on stosowany jako szkielet, na którym można opierać swoje przyszłe dokumenty. Niektóre z wykorzystanych instrukcji mogą być niejasne, ale wyjaśnienie ich znaczenia zawarte jest w kolejnej części książki.

```
\documentclass{mwart}

\usepackage[utf8]{inputenc}
\usepackage{polski}

\title{Tytuł dokumentu}
\author{Autor dokumentu}

\begin{document}
\maketitle

To jest mój artykuł.
\end{document}
```

Ćwiczenia

Ćwiczenie 14. Korzystając z szablonu zaproponowanego na końcu poprzedniego paragrafu, spisz w kilku akapitach swoje odczucia dotyczące \LaTeX a. Spróbuj dokonać zmian w pliku źródłowym i porównaj otrzymane efekty.

Podsumowanie

Donald E. Knuth wyświadczył nam wielką przysługę – nie tylko dlatego, że stworzył oprogramowanie $\text{T}_{\text{E}}\text{X}$, na którym opiera się popularny dziś system składu tekstu o nazwie $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$, ale także dlatego, że wciąż pokazuje nam, jak ważna jest dbałość o szczegół, nie tylko typograficzny.

Owiane legendą są już nagrody Knutha. Zadeklarował on, że zapłaci 2,56 dolara każdemu, kto znajdzie nieznaną wcześniej błąd w którejkolwiek z jego książek – niezależnie czy będzie to błąd techniczny, typograficzny czy historyczny. Jak tłumaczy, 256 centów nawiązuje do heksadecymalnej wartości dolara. [W jednym z udzielonych wywiadów Knuth przyznał, że do maja 2005 roku wypisał już z tego tytułu czeki na łączną wartość ponad dwudziestu tysięcy dolarów.](#)

Jeden heksadecymalny dolar to 0×100 centów. 0×100 w systemie heksadecymalnym jest równe 256 w systemie dziesiętnym.

W przypadku oprogramowania $\text{T}_{\text{E}}\text{X}$ i *Metafont*, system nagród za znalezione błędy wygląda trochę inaczej. Zainspirowany problemem ziaren na szachownicy,⁸ Knuth postanowił, że w pierwszym roku wynagrodzi kwotą 2,56 dolara osoby, które znajdą nowy błąd w którymś z tych programów, a co roku nagroda ta będzie się podwajać. Działo się tak, dopóki nie osiągnęła ona kwoty 327,68 dolara. Dziś pomiędzy kolejnymi zgłoszeniami błędów mijają lata, a czeki wystawione przez Knutha mają charakter bardziej kolekcjonerski niż praktyczny, będąc wyróżnieniem o wiele cenniejszym niż waluta.

Ta dbałość o szczegóły prezentowana przez Knutha doskonale uzupełnia się z dążeniem do nienagannie wyglądających dokumentów, których tworzenie miał od początku umożliwiać $\text{T}_{\text{E}}\text{X}$. Dzisiaj korzystamy z rozbudowanej na

⁸ Zainteresowanych odsyłamy na stronę internetową poświęconą temu problemowi: http://en.wikipedia.org/wiki/Wheat_and_chessboard_problem.

kładki o nazwie \LaTeX , dostarczającej nam (zarówno bezpośrednio, jak i poprzez zewnętrzne klasy i pakiety) wiele nowych możliwości.

\LaTeX wymaga jednak zmiany pewnych przyzwyczajeń. Nie działa on tak, jak znane nam narzędzia WYSIWYG, na przykład *Microsoft Word*. Opiera się na plikach tekstowych, a informacje dotyczące tego, w jaki sposób wygenerować finalny dokument, przekazywane są do silnika przy użyciu specjalnych poleceń. Dzięki znakowaniu semantycznemu i oddzieleniu warstwy wizualnej od treści, \LaTeX pozwala skupić się na istocie pracy, pozostawiając kwestie wizualne ekspertom, którzy przygotowali już odpowiednie klasy.

Czytelnik, który zapoznał się z zagadnieniami poruszonymi w tej części książki, gotowy jest już do składania prostych dokumentów zawierających podzielony na akapity ciągły tekst w języku polskim. Wraz z doświadczeniem pojawi się jednak konieczność logicznego podziału dokumentu oraz umieszczenia w nim innych elementów. Może też zająć potrzeba zastosowania innych rozwiązań pozwalających nam osiągnąć zamierzone efekty wizualne. Odpowiednie receptury prezentowane są w kolejnej części.

Część II

Receptury

Wprowadzenie do receptur

Ta część książki poświęcona jest recepturom, to znaczy gotowym rozwiązaniom prowadzącym do uzyskania określonych efektów. Podzielona ona została na kilka części tematycznych w sposób umożliwiający szybkie odszukanie wybranego zagadnienia. Staraliśmy się jednak, wszędzie tam gdzie to możliwe, wplatać praktyczne wskazówki i uwagi dotyczące zarówno \LaTeX a, jak i ogólnych zasad typografii związanych z omawianymi elementami.

Niemal każda z receptur rozpoczyna się od przykładu i oraz kodu źródłowego *pełnego dokumentu*, prowadzącego do uzyskania pokazanego efektu. Pierwszy wiersz kodu źródłowego stanowi komentarz zawierający nazwę przykładu, pod jaką występuje on w materiałach dołączanych do wersji elektronicznej książki.

Aby pokazać efekt w naturalnej wielkości, musieliśmy dokonać zmiany parametrów finalnego dokumentu w taki sposób, aby możliwe było umieszczenie go w skali 1:1 na stronach tej książki. Za to działanie odpowiada pakiet `latex-ksiazka-kucharska`, dołączony do każdego z przykładów. Ma on jednak charakter techniczny i tak należy go traktować – jego pominięcie nie wpływa bowiem w żaden sposób na poprawność przykładowego pliku.

Niektóre z receptur mają charakter komentarza uzupełniającego, nie są więc wzbogacone przykładem.

Zawartość tego pakietu została dokładniej opisana w jednym z dodatków na końcu tej książki.

Wyliczenia

Autor dokumentu często staje przed koniecznością prezentacji informacji w postaci list – zarówno nienumerowanych, jak i numerowanych. \LaTeX wyposażony jest w otoczenia `itemize` oraz `enumerate`, umożliwiające wykonanie tego zadania. Dzięki zastosowaniu pakietu `enumitem`, możliwa staje się modyfikacja parametrów wyglądu tych list.

W tym rozdziale prezentujemy przykłady związane z listami umieszczanymi w dokumentach składanych w języku polskim (opartych na klasach autorstwa Marcina Wolińskiego). W przypadku klas anglosaskich (np. `article`), wypunktowania i wyliczenia mogą wyglądać inaczej.

Listy nienumerowane

- Lorem ipsum,
- dolor sit amet:
 - consectetuer adipiscing elit,
 - ut purus elit vestibulum ut, placerat ac, adipiscing vitae, felis.

```
% wyliczenia-nienumerowane.tex
\documentclass{mwart}
\usepackage{latex-ksiazka-kucharska}

\begin{document}
\begin{itemize}
\item Lorem ipsum,
\item dolor sit amet:
  \begin{itemize}
\item consectetuer adipiscing elit,
\item ut purus elit vestibulum ut, placerat ac,
      adipiscing vitae, felis.
  \end{itemize}
\end{itemize}
\end{document}
```

Do składania list nienumerowanych (inaczej nazywanych nieuporządkowanymi) służy środowisko `itemize`. Poszczególne punkty rozpoczynają się od polecenia `\item`. Dzięki temu można tworzyć wewnątrz punktu nie tylko akapity, ale także kolejne listy, tworząc wypunktowanie wielopoziomowe.

`\item`

Zwróćmy przy okazji uwagę na to, że – jeżeli kolejne punkty wylczenia kończymy przecinkami, a nie kropkami (a tak należy robić, gdy nie są one pełnymi zdaniami) – to kolejny punkt rozpoczynamy małą literą.

Listy numerowane

1. Lorem ipsum,
2. dolor sit amet:
 - a) consetetuer adipiscing elit,
 - b) ut purus elit.

```
% wylczenia-numerowane.tex
\documentclass{mwart}
\usepackage{latex-ksiazka-kucharska}

\begin{document}
\begin{enumerate}
\item Lorem ipsum,
\item dolor sit amet:
  \begin{enumerate}
  \item consetetuer adipiscing elit,
  \item ut purus elit.
  \end{enumerate}
\end{enumerate}
\end{document}
```

Składanie list numerowanych jest możliwe z wykorzystaniem środowiska `enumerate`. Środowiska tego używa się całkowicie analogicznie do omówionego wcześniej `itemize`.

Zmiana symbolu wypunktowania lub sposobu numeracji listy

- * Lorem ipsum,
 - * dolor sit amet.
- (a) Lorem ipsum,
 - (b) dolor sit amet.

```
% wyliczenia-zmiana-numeracji.tex
\documentclass{mwart}
\usepackage{latex-ksiazka-kucharska}
\usepackage{enumitem}

\begin{document}
\begin{itemize}[label=\textasteriskcentered]
\item Lorem ipsum,
\item dolor sit amet.
\end{itemize}

\begin{enumerate}[label={(\alph*)}]
\item Lorem ipsum,
\item dolor sit amet.
\end{enumerate}
\end{document}
```

Standardowo, \LaTeX nie pozwala na łatwą zmianę symbolu wyliczenia. Istnieje jednak kilka pakietów, które to ułatwiają. Obecnie najpopularniejszy z nich to `enumitem`. Pozwala on na dodanie do każdego środowiska składającego wyliczenie parametru opcjonalnego, zawierającego różne ustawienia, zapisane w postaci $\langle parametr \rangle = \langle wartość \rangle$. Zauważmy, że w tym przypadku ar-

gumenty opcjonalne podawane są *po* argumentie obowiązkowym polecenia `\begin` – w taki sposób podaje się *argumenty opcjonalne otoczeń*. Jak już wiemy, jeśli zachodzi potrzeba przekazania kilku takich argumentów, należy je oddzielić przecinkiem.

Parametr `label` pozwala określić, jak będą oznaczane kolejne punkty listy; polecenie `\textasteriskcentered` wypisuje z kolei gwiazdkę w połowie wysokości małej litery. W praktyce, etykietą elementów listy może być dowolny symbol lub ciąg znaków. Symbole specjalne udostępniane przez \LaTeX a zostały zebrane w odpowiednim dodatku na końcu tej książki.

W przypadku list numerowanych użycie stałych punktorów mija się jednak z celem. Na szczęście możemy skorzystać ze zbioru poleceń opisanych w tabeli 4. Istotną częścią tych instrukcji jest symbol gwiazdki [znajdujący się na ich końcach](#) – jego pominięcie doprowadzi z pewnością do wystąpienia błędu kompilacji.

`\textasteriskcentered`

Formalnie rzecz biorąc, symbol ten nazywa się *asteryskiem*.

Tabela 4. Style numeracji w pakiecie `enumitem`

Polecenie	Opis	Przykład
<code>\arabic*</code>	liczby arabskie	1, 2, 3
<code>\alph*</code>	małe litery alfabetu łacińskiego	a, b, c
<code>\Alph*</code>	duże litery alfabetu łacińskiego	A, B, C
<code>\roman*</code>	małe liczby rzymskie	i, ii, iii
<code>\Roman*</code>	duże liczby rzymskie	I, II, III

Próba wykorzystania któregoś z opisanych w powyższej tabeli poleceń w kontekście środowiska `itemize` zakończy się błędem.

Dobrym przyzwyczajeniem jest umieszczanie skomplikowanych wartości parametru `label` w nawiasach klamrowych. Jest to szczególnie ważne, jeśli częścią wzorca numeracji lub wypunktowania jest znak przecinka.

Zmiana symbolu wypunktowania wybranych elementów listy

- Lorem ipsum,
 - * dolor sit amet,
 - consectetur adipiscing elit.
1. Lorem ipsum,
 - * dolor sit amet,
 2. consectetur adipiscing elit.

```
% wyliczenia-zmiana-wybranych.tex
\documentclass{mwart}
\usepackage{latex-ksiazka-kucharska}

\begin{document}
\begin{itemize}
\item Lorem ipsum,
\item[\textasteriskcentered] dolor sit amet,
\item consectetur adipiscing elit.
\end{itemize}

\begin{enumerate}
\item Lorem ipsum,
\item[\textasteriskcentered] dolor sit amet,
\item consectetur adipiscing elit.
\end{enumerate}
\end{document}
```

Czasami istnieje potrzeba zmiany symbolu wypunktowania tylko dla wybranych elementów listy. Polecenie `\item` przyjmuje parametr opcjonalny (umieszczany w nawiasach kwadratowych), dzięki któremu możemy określić

`\item`

niestandardowy symbol wypunktowania. Ten sposób działa zarówno w przypadku list numerowanych, jak i nienumerowanych.

Uwaga. W tym przypadku korzystanie z poleceń opisanych w tabeli 4 nie jest możliwe, nawet gdy zmiana dotyczy elementu listy numerowanej.

Zmiana symbolu wypunktowania wszystkich list nienumerowanych

- * Lorem ipsum,
- * dolor sit amet:
 - consectetur adipiscing elit,
 - ut purus elit, vestibulum ut.

```
% wyliczenia-zmiana-wszystkich-nienumerowanych.tex
\documentclass{mwart}
\usepackage{latex-ksiazka-kucharska}
\usepackage{enumitem}
\setitemize[1]{label=\textasteriskcentered}
\setitemize[2]{label={--}}

\begin{document}
\begin{itemize}
\item Lorem ipsum,
\item dolor sit amet:
  \begin{itemize}
\item consectetur adipiscing elit,
\item ut purus elit, vestibulum ut.
```



```

\end{itemize}
\end{itemize}
\end{document}

```

Wraz pakietem `enumitem` udostępnione nam zostaje polecenie `\setitemize`. Umieszczone w preambule dokumentu, pozwala zmienić parametry *wszystkich* wyliczeń nienumerowanych. Argumentem obowiązkowym tego polecenia jest zbiór ustawień w postaci $\langle parametr \rangle = \langle wartość \rangle$. Jeśli istnieje potrzeba zmiany kilku parametrów jednocześnie, należy je oddzielić przecinkami.

```
\setitemize
```

Instrukcję `\setitemize` możemy wywołać z parametrem opcjonalnym – numerem poziomu listy, którego modyfikacja ma dotyczyć. Przykładowo, deklaracja `\setitemize[2]{label={--}}` oznacza, że punkty wyliczeń nienumerowanych na drugim poziomie zagnieżdżenia (czyli wewnątrz jednego wyliczenia) będą oznaczane półpauzą.

Zmiana sposobu numeracji wszystkich list numerowanych

- ```

A. Lorem ipsum,
B. dolor sit amet:
 (i) consetetuer adipiscing elit,
 (ii) ut purus elit.

```

```

% wyliczenia-zmiana-wszystkich-nienumerowanych.tex
\documentclass{mwart}
\usepackage{latex-ksiazka-kucharska}
\usepackage{enumitem}

```

```
\setenumerate[1]{label={\Alph*.}}
\setenumerate[2]{label={(\roman*)}}

\begin{document}
\begin{enumerate}
\item Lorem ipsum,
\item dolor sit amet:
 \begin{enumerate}
 \item consectetur adipiscing elit,
 \item ut purus elit.
 \end{enumerate}
\end{enumerate}
\end{document}
```

W przypadku list numerowanych, zmiana sposobu numeracji odbywa się analogicznie do przypadku list nienumerowanych, przy czym poleceniem pozwalającym zmieniać parametry list numerowanych jest `\setenumerate`. Deklarując styl numeracji (`label`) w ramach polecenia `\setenumerate`, możemy posługiwać się poleceniami z *tabeli 4*.

`\setenumerate`

## Wyliczenia „długie”

1. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna.
2. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo.

```
% wyliczenia-dlugie.tex
\documentclass{mwart}
\usepackage{polski}
\usepackage{latex-ksiazka-kucharska}

\begin{document}
\begin{enumerate*}
\item Lorem ipsum dolor sit amet, consectetur
 adipiscing elit. Ut purus elit, vestibulum ut,
 placerat ac, adipiscing vitae, felis. Curabitur dictum
 gravida mauris. Nam arcu libero, nonummy eget,
 consectetur id, vulputate a, magna.
\item Donec vehicula augue eu neque. Pellentesque
 habitant morbi tristique senectus et netus et
 malesuada fames ac turpis egestas. Mauris ut leo.
\end{enumerate*}
\end{document}
```

Zwykle poszczególne elementy wyliczeń mają kilka, rzadko kilkanaście słów. Zdarzają się jednak czasem wyliczenia, w których poszczególne elementy mają i po kilka zdań. W takiej sytuacji środowiska `enumerate` czy `itemize` dają niezbyt estetyczny efekt, gdyż długawy tekst jest cały wcięty z lewej strony. Polskie zwyczaje typograficzne nakazują składać wyliczenia zbudowane z dłuższych elementów bez takiego wcięcia. Wychodząc im na przeciw, klasy z pakietu `mwc-ls` definiują warianty „z gwiazdką” (czyli otoczenia `enumerate*` oraz `itemize*`), w których poszczególne elementy wyliczenia rozciągają się od lewego marginesu do prawego. Należy mimo to unikać sytuacji, w której jeden element wyliczenia ma więcej niż jeden akapit, bowiem wewnątrz tych otoczeń wcięcie akapitowe jest wyłączone.

I słusznie, gdyż wcięcie akapitowe sugerowałoby koniec wyliczenia.

## Rozdziały i paragrafy

Pisanie uporządkowanej pracy wymaga od autora określenia hierarchicznej struktury dokumentu. Książka dzieli się na rozdziały, te z kolei na paragrafy, podparagrafy itd. [Również prace takie jak artykuły czy raporty, wymagają zachowania pewnego porządku.](#)

Celem tego rozdziału jest przybliżenie Czytelnikowi poleceń  $\LaTeX$ a związanych z tym zagadnieniem.

Paragraf to po angielsku *section*, a angielskie słowo *paragraph* oznacza *akapit*. Będziemy konsekwentnie używać stosowanej w Polsce terminologii.

## Rozdział 1

# Lorem ipsum

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.

Rysunek 1. Wyciąg z dokumentu wynikowego (strona 1)

## Oznaczanie rozdziału

Wyciąg z dokumentu wynikowego pokazano na *rysunku 1*, s. 53.

```
% rozdzialy-rozdzialy.tex
\documentclass{mwbk}
\usepackage{polski}
\usepackage{latex-ksiazka-kucharska}

\begin{document}
\chapter{Lorem ipsum}
Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Ut purus elit, vestibulum ut, placerat ac,
adipiscing vitae, felis.

\chapter{Nam dui ligula}
Nam dui ligula, fringilla a, euismod sodales,
sollicitudin vel, wisi. Morbi auctor lorem non
justo. Nam lacus libero, pretium at, lobortis vitae.
\end{document}
```

Tworzenie rozdziałów możliwe jest tylko w niektórych klasach, np. `mwbk`. Rozpoczęcie nowego sygnalizujemy  $\LaTeX$ owi poleceniem `\chapter`. Argumentem obowiązkowym tego polecenia jest nazwa rozdziału.

Każdy rozdział rozpoczyna się na nowej stronie, a w przypadku składu dwustronnego na stronie prawej, czyli o nieparzystym numerze. Jeśli chcemy, aby  $\LaTeX$  składał książki jednostronnie (wówczas rozdział może rozpocząć się na stronie o numerze parzystym), powinniśmy wywołać klasę `mwbk` z parametrem opcjonalnym `oneside`.

```
\documentclass[oneside]{mwbk}
```

Wpływa to jednak na wiele aspektów składu, np. na sposób stosowania marginesów wewnętrznych i zewnętrznych – musimy więc być ostrożni.

`\chapter`

## Rozdział 1

# Lorem ipsum

Lorem ipsum dolor sit amet.

### 1.1. Nam dui ligula

Nam dui ligula, fringilla a, euismod sodales.

#### 1.1.1. Nulla malesuada

Nulla malesuada porttitor diam.

#### Quisque ullamcorper

Quisque ullamcorper placerat ipsum. Cras nibh.

**Fusce mauris** Fusce mauris. Vestibulum luctus nibh at lectus.

Rysunek 2. Wyciąg z dokumentu wynikowego

## Oznaczanie podrozdziału i mniejszych jednostek

Wyciąg z dokumentu wynikowego pokazano na *rysunku 2*, s. 55.

```
% rozdzialy-podrozdzialy.tex
\documentclass{mwbk}
\usepackage{polski}
\usepackage{latex-ksiazka-kucharska}

\begin{document}
\chapter{Lorem ipsum}
Lorem ipsum dolor sit amet.

\section{Nam dui ligula}
Nam dui ligula, fringilla a, euismod sodales.

\subsection{Nulla malesuada}
Nulla malesuada porttitor diam.

\subsubsection{Quisque ullamcorper}
Quisque ullamcorper placerat ipsum. Cras nibh.

\paragraph{Fusce mauris}
Fusce mauris. Vestibulum luctus nibh at lectus.
\end{document}
```

Oczywiście, wiele książek poza rozdziałami ma podrozdziały (paragrafy). W  $\text{\LaTeX}$  oznacza się je poleceniem `\section`, którego parametry (obowiązkowy i dwa opcjonalne) są analogiczne do parametrów polecenia `\chapter`.  $\text{\LaTeX}$  pozwala stosować aż sześciopoziomową strukturę rozdziałów (w klasach `article` i pochodnych pięciopoziomową, bez polecenia `\chapter`), jednak w typowych sytuacjach struktura głębsza niż trzy poziomy (rozdział, paragraf i podparagraf – oznaczany poleceniem `\subsection`) wydaje się nie tylko niepotrzebna, ale też nadzwyczaj nieprzyjazna dla odbiorcy.

`\section`

`\subsection`



## Lorem ipsum

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.

Rysunek 3. Wyciąg z dokumentu wynikowego (strona 1)

## Jednostki nienumerowane

Wyciąg z dokumentu wynikowego pokazano na *rysunku 3*, s. 57.

```
% rozdzialy-nienumerowane.tex
\documentclass{mwbk}
\usepackage{polski}
\usepackage{latex-ksiazka-kucharska}

\begin{document}
\chapter*{Lorem ipsum}
Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Ut purus elit, vestibulum ut, placerat ac,
adipiscing vitae, felis.

\chapter{Nam dui ligula, fringilla a, euismod sodales,
sollicitudin vel, wisi}
Nam dui ligula, fringilla a, euismod sodales,
sollicitudin vel, wisi. Morbi auctor lorem non
justo.
\end{document}
```

Polecenie `\chapter*` wstawia rozdział bez numeru. Nadaje się ono szczególnie do wstępów, dodatków, bibliografii itp. Analogiczne polecenie `\section*` wstawia nienumerowany paragraf. Czytelnik z łatwością odgadnie, jak umieścić w dokumencie nienumerowany podparagraf.

**Uwaga.** W standardowych klasach  $\LaTeX$ a rozdziały, paragrafy itp. bez numeru nie są umieszczane w spisie treści (por. s. 98), co jest źródłem frustracji wielu użytkowników  $\LaTeX$ a. W klasach `mwc` `ls` problem ten nie występuje.

`\chapter*`

`\section*`

Rozdział 1

**Nam dui ligula,  
fringilla a, euismod sodales,  
sollicitudin vel, wisi**

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo.

Rysunek 4. Wyciąg z dokumentu wynikowego

## Dzielenie tytułu rozdziału na wiersze

Wyciąg z dokumentu wynikowego pokazano na *rysunku 4*, s. 59.

```
% rozdzialy-wiersze.tex
\documentclass{mwbk}
\usepackage{polski}
\usepackage{latex-ksiazka-kucharska}

\begin{document}
\chapter{Nam dui ligula, \\\fringilla a, euismod sodales,
 sollicitudin vel, wisi}
Nam dui ligula, fringilla a, euismod sodales,
sollicitudin vel, wisi. Morbi auctor lorem non justo.
\end{document}
```

Z uwagi na to, że tytuły rozdziału czy paragrafu to zwykle pierwsze rzeczy, jakie rzucają się czytelnikowi w oczy po otwarciu książki na danej stronie, dobrze jest zadbać o ich dobry wygląd. Nie ułatwia tej kwestii fakt, że tytuł składany jest często większym niż zwykle stopniem pisma, co oznacza, że dłuższe tytuły mogą zostać podzielone na wiersze w nienajlepszym miejscu. Dobrym zwyczajem jest bowiem, by w tytułach nie rozdzielać między wierszami słów, które są ze sobą związane, a także by nie zostawiać np. pojedynczego słowa w ostatnim, króciutkim wierszu.

Polecenie `\\` w tytule rozdziału powoduje, że w tym miejscu zostanie złamany wiersz. Można też oznaczyć twarde spacje znakiem tyldy (`~`). Ma to tę zaletę, że – w przeciwieństwie do polecenia `\\` – nie powoduje złamania wiersza w spisie treści czy żywej paginie.

Więcej na temat znaku tyldy na stronie 25.

## Spis treści

Rozdział 1. Nam dui ligula, fringilla a . . . . . 3

Rysunek 5. Wyciąg z dokumentu wynikowego (strona 1)

## Warianty tytułu do żywej paginy i spisu treści

Wyciąg z dokumentu wynikowego pokazano na rysunku 5, s. 61.

```
% rozdzial-pagina.tex
\documentclass{mwbk}
\usepackage{polski}
\usepackage{latex-ksiazka-kucharska}

\begin{document}
\tableofcontents
\chapter[Nam dui ligula]
 [Nam dui ligula, fringilla a]
 {Nam dui ligula, fringilla a, euismod sodales}
\end{document}
```

Może się zdarzyć, że tytuł rozdziału bądź paragrafu jest zbyt długi, aby mieścił się komfortowo w żywej paginie (normy zalecają nieprzekraczanie  $\frac{2}{3}$  szerokości wiersza). Wówczas krótszy wariant tytułu do żywej paginy podaje się w parametrze opcjonalnym polecenia `\chapter` czy `\section` (w nawiasach kwadratowych).

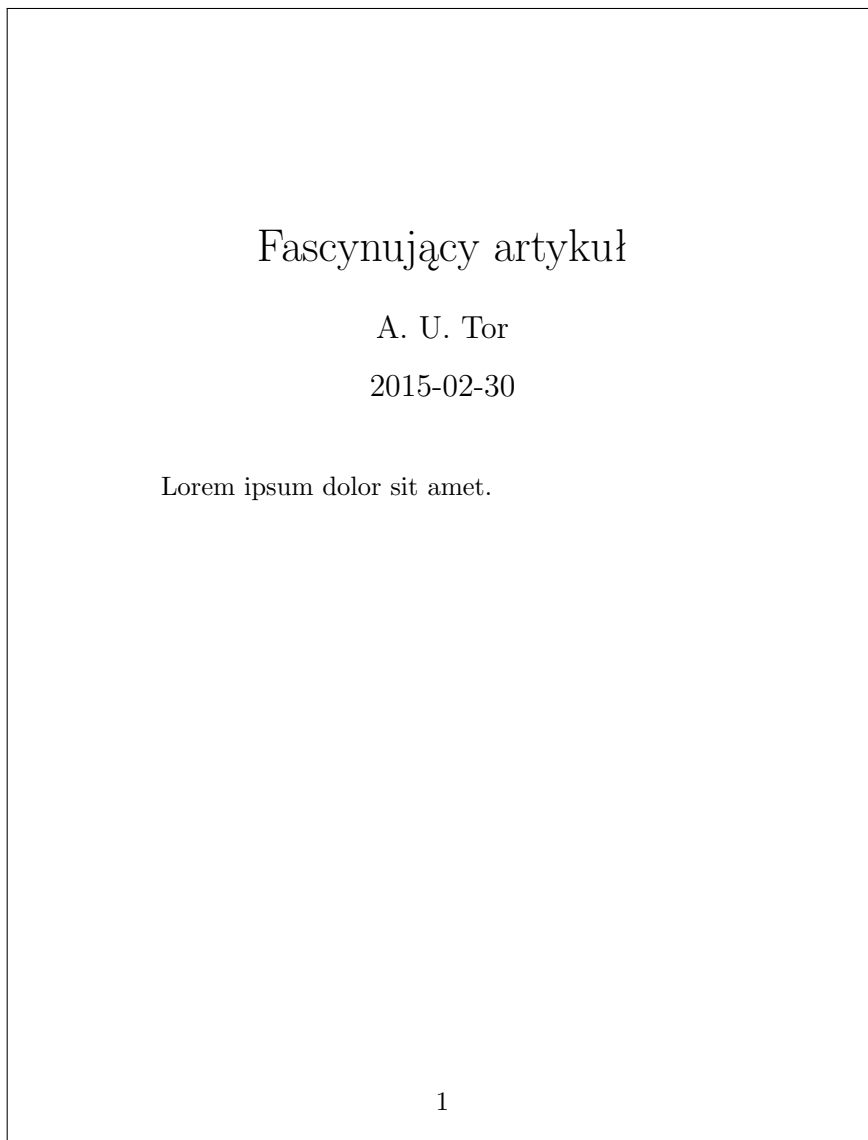
Może się też zdarzyć, że potrzebny jest również specjalny wariant tytułu do spisu treści (będzie tak np. w sytuacji, gdy wymusiliśmy złamanie wiersza w tytule poleceniem `\`). Można wtedy podać wariant tytułu do spisu treści w drugim parametrze opcjonalnym.

Żywa pagina to, mówiąc bardzo obrazowo, nagłówek strony, który zawiera nazwę aktualnie czytanego rozdziału lub paragrafu.

## Części większe od rozdziałów

Szczególnie duże dzieła można podzielić na części, składające się z wielu rozdziałów każda. Do rozpoczęcia części służy polecenie `\part`, które dzieła analogicznie do polecenia `\chapter` z tą różnicą, że pierwszy argument opcjonalny, wykorzystywany do deklarowania wariantu tytułu do żywej paginy, nie jest tu wykorzystywany. Możemy zatem użyć np. konstrukcji `\part[][Tytuł do spisu treści]{Właściwy tytuł}`.

`\part`



Rysunek 6. Wyciąg z dokumentu wynikowego

## Strona tytułowa

Wyciąg z dokumentu wynikowego pokazano na rysunku 6, s. 63.

```
% rozdzialy-tytulowa.tex
\documentclass{mwart}
\usepackage{polski}
\usepackage{latex-ksiazka-kucharska}

\begin{document}
\author{A. U. Tor}
\title{Ciekawy tekst}
\date{2015-02-30}
\maketitle

Lorem ipsum dolor sit amet.
\end{document}
```

W standardowych klasach  $\LaTeX$ a wsparcie dla tworzenia strony tytułowej jest raczej ubogie. Można podać autora, tytuł i (ewentualnie) datę powstania dokumentu, a następnie poleceniem `\maketitle` umieścić te dane w pliku wynikowym. W przypadku klas służących do składu książek (np. `mwbk`) tytułaria umieszczone zostaną ponadto na oddzielnej stronie.

Gdy chcemy zrobić bardziej skomplikowaną stronę tytułową, możemy użyć metod opisanych w rozdziale poświęconym akapitom (zob. s. 78 i 80). Warto też pamiętać o wydaniu polecenia `\thispagestyle{empty}` (zob. s. 146), dzięki czemu strona tytułowa będzie wliczana do numeracji, ale nic nie pojawi się w jej nagłówku ani stopce.

`\maketitle`



# Mikrotypografia

## Wyróżnianie fragmentu tekstu

Lorem *ipsum* dolor sit amet, *consectetuer* adipiscing **elit**. *Ut purus elit*, vestibulum *ut*, *placemat ac*, *adipiscing vitae*, *felis*.

```
% mikrotypografia-wyrozniwanie.tex
\documentclass{mwart}
\usepackage{latex-ksiazka-kucharska}

\begin{document}
Lorem \emph{ipsum} dolor sit amet, \textit{consectetuer}
adipiscing \textbf{elit}. \emph{Ut purus elit,
\emph{vestibulum} ut, placemat ac, adipiscing vitae,
felis.}
\end{document}
```

Standardowym sposobem wyróżnienia fragmentu tekstu jest *kursywa*.  
Poleceniem  $\LaTeX$ a służącym do wyróżniania jest `\emph`, które składa frag-

`\emph`

ment tekstu w nawiasach klamrowych kursywą, gdy tekst wokół jest prosty, i pismem prostym, gdy tekst wokół jest kursywny. Poleceniem, które zawsze włącza kursywę, jest `\textit`.

`\textit`

Znacznie rzadziej stosowane jest wytłuszczenie. Dobrą zasadą jest unikanie wytłuszczenia, o ile tylko się da – powinno się je stosować tylko wtedy, gdy jest to naprawdę uzasadnione (czyli bardzo rzadko). Ogólna zasada jest następująca: wytłuszczenie służy do zwrócenia uwagi czytelnika, gdy *prze-gląda* tekst, a kursywa, gdy *go czyta*. Właściwym użyciem wytłuszczenia są więc np. nagłówki podrozdziałów czy hasła w słownikach, które chcemy móc znaleźć „na pierwszy rzut oka”, bez uważnej lektury całości. Wytłuszczenie włącza się komendą `\textbf`.

`\textbf`

## Cudzysłów, myślniki, łączniki...

Cudzysłów „polskie”, „angielskie” i wreszcie – rzadko spotykane – «francuskie».

Lata 1918–1939, twierdzenie Hahna–Banacha, Maria Skłodowska-Curie

```
% mikrotypografia-cudzyslowy.tex
\documentclass{mwart}
\usepackage[utf8]{inputenc}
\usepackage{polski}
\usepackage{latex-ksiazka-kucharska}

\begin{document}
Cudzysłów „polskie”, „angielskie” i wreszcie
\ppauza rzadko spotykane \ppauza <<francuskie>>.
```

```
Lata 1918--1939, twierdzenie Hahna--Banacha,
Maria Skłodowska\dywiz Curie
\end{document}
```

W różnych krajach zapisuje się cudzysłowy w różny sposób. „Angielskie” cudzysłowy różnią się od „polskich”, zaś «francuskie» wyglądają jeszcze inaczej. Sposób wpisywania cudzysłowów w  $\text{\LaTeX}$ u jest zależny od użytego fon- tu, ale przy standardowych ustawieniach używa się do nich podwójnych apo- strofów (zwykłych i „odwróconych”) oraz podwójnych symboli, które moż- na opisać jako „większe/mniejsze”.

Poziome kreski są również źródłem licznych nieporozumień, gdyż na stan- dardowej klawiaturze można znaleźć wyłącznie znak „minus”. Nieco uprasz- czając, można przyjąć, że potrzebujemy głównie dwóch ich rodzajów: łączni- ka (dywizu) oraz *półpauzy*. Użycie obu z nich, gdy dołączony do dokumentu został pakiet `połski`, jest bardzo proste.

Dywiz uzyskuje się w  $\text{\LaTeX}$ u przy użyciu polecenia `\dywi z` (wersja do wy- rzaków złożonych, pozwalająca na przeniesienie do nowego wiersza) lub minu- sa (wersja blokująca przeniesienie do nowego wiersza). Półpauzę natomiast uzyskuje się poleceniem `\ppauza` (wersja ze spacjami) lub dwoma następu- jącymi zaraz po sobie minusami (wersja bez spacji).

Łącznika należy używać w wyrazach złożonych (jak biało-czerwony), zaś półpauzy w większości pozostałych przypadków – jako myślnika, jako ozna- czenia zakresów liczbowych (np. 0–18) czy w nazwach twierdzeń autorstwa dwóch lub więcej osób (jak lemat Kuratowskiego–Zorna).

**Uwaga.** W wyrazach złożonych zawierających znak – lub -- domyślnie wy- łączone jest automatyczne dzielenie wyrazów. Polecenie `\dywi z` pozwala ominąć ten problem w przypadku łącznika. Niestety, nie ma analogicznego polecenia dla półpauzy. Gdy więc chcemy móc podzielić np. „lemat Kura- towskiego–Zorna”, możemy ręcznie wskazać pożądane miejsce dzielenia: `Kura\–towskiego–Zorna`.

Głównym powodem, dla którego użyliśmy niezbyt poprawnej nazwy „od- wrócony apostrof” (ang. *backtick*) na znak ‘, jest to, że właściwą nazwę, czyli *grawis*, zna prawdopodobnie poniżej promiła czytelników...

`\dywi z`

`\ppauza`

## Przypisy

tas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Nulla malesuada porttitor diam.<sup>1</sup> Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ip-

---

<sup>1</sup> Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit.

```
% mikrotypografia-przypisy.tex
\documentclass{mwart}
\usepackage{latex-ksiazka-kucharska}
\usepackage{lipsum}

\begin{document}
\lipsum[4-5]

Nulla malesuada porttitor diam.\footnote{Donec felis
erat, congue non, volutpat at, tincidunt tristique,
libero. Vivamus viverra fermentum felis. Donec
nonummy pellentesque ante. Phasellus adipiscing semper
elit.} Proin fermentum massa ac quam. Sed diam
turpis, molestie vitae, placerat a, molestie nec,
leo. Maecenas lacinia. Nam ipsum ligula, eleifend at,
accumsan nec, suscipit a, ipsum. Morbi blandit ligula
feugiat magna.
\end{document}
```

Przede wszystkim przypisów lepiej unikać, gdyż rozpraszają one czytelnika. Gdy musimy ich jednak użyć, wykorzystujemy do tego celu polecenie `\footnote`. Polecenie to dba o automatyczną numerację przypisów.

`\footnote`

## Adresy email i stron internetowych

Email wysłany na adres `knuth@stanford.edu` nie trafi do prof. Donalda E. Knutha. Wyjaśnienie, dlaczego, można znaleźć na stronie `http://www-cs-faculty.stanford.edu/~uno/email.html`.

```
% mikrotypografia-adresy.tex
\documentclass{mwart}
\usepackage[utf8]{inputenc}
\usepackage{polski}
\usepackage{latex-ksiazka-kucharska}
\usepackage{url}

\begin{document}
Email wysłany na adres \url{knuth@stanford.edu} nie
trafi do prof. Donalda E.~Knutha. Wyjaśnienie,
dlaczego, można znaleźć na stronie
\url{http://www-cs-faculty.stanford.edu/~uno/email.html}.
\end{document}
```

Adresy stron internetowych i niektóre adresy email wymagają szczególnego potraktowania. Po pierwsze, tradycyjnie składa się je pismem, w którym znaki mają stałą szerokość. Po drugie, często występują w nich znaki, które dla  $\text{\LaTeX}$ a mają szczególne znaczenie, jak np. znaki podkreślenia czy tyldy.

Po trzecie, przenoszenie ich do następnego wiersza wymaga niestosowania myślnika, gdyż mógłby on być łatwo wzięty za element adresu. Wszystkie te problemy rozwiązuje pakiet `url`.

Niestety, z przyczyn technicznych polecenie `\url` z tego pakietu nie może być używane w niektórych miejscach dokumentu (np. w przypisach lub tytułach paragrafów), jeżeli jego argument zawiera znaki specjalne  $\LaTeX$ a. W takiej sytuacji należy użyć polecenia `\urldef` (na przykład w preambule dokumentu). Gdy adres zawiera nawiasy klamrowe, kończy się znakiem `\` lub zawiera pewne inne, w normalnej sytuacji niedozwolone ciągi znaków, można po poleceniu `\url` użyć jakiegoś znaku nie występującego w adresie (np. kreski pionowej) i tym samym znakiem oznaczyć koniec adresu.

Można również sprawić, by adres w pliku pdf był aktywnym linkiem, tj. by kliknięcie na nim otwierało np. przeglądarkę z odpowiednią stroną internetową. Służy do tego pakiet `hyperref`, który również definiuje polecenie `\url`, tym razem generujące link.

`\url`

Powody tego stanu rzeczy są subtelne i Czytelnik prawdopodobnie nie chce znać odpowiedzi na pytanie, dlaczego tak jest.

`\urldef`

`@#^{\`

```
% mikrotypografia-urldef.tex
\documentclass{mwart}
\usepackage{latex-ksiazka-kucharska}
\usepackage{url}

\urldef\dziwnyadres\url|@#^{\|

\begin{document}
\dziwnyadres
\end{document}
```

## Litery z akcentami

```
Grüß dich!
déjà vu
reguła de l'Hôpitala
ïj
```

```
% mikrotypografia-akcenty.tex
\documentclass{mwart}
\usepackage{latex-ksiazka-kucharska}

\begin{document}
Gr\u\ss{} dich!

d\`{e}j\`{a} vu

regu{\l}a de l'H^{o}pitala

\v{\i}\^{j}
\end{document}
```

Istnieje szereg komend  $\LaTeX$ a służących do składu akcentów i liter spoza alfabetu łacińskiego. Niektóre z nich widać w przykładowym pliku zaprezentowanym powyżej. Warto pamiętać, że  $\LaTeX$  „zjada” spacje po słowach kontrolnych (zob. s. 19). Warto również zwrócić uwagę na polecenia  $\backslash i$  oraz  $\backslash j$ , dające litery „i” i „j” bez kropki, nadające się do umieszczenia nad nimi specjalnego akcentu.

$\backslash i$   
 $\backslash j$

## Twarde spacje

Niniejsze (niezbyt sensowne) zdanie zawiera jednoliterowy przyimek w bardzo niekorzystnym miejscu, a zatem trochę głupio wygląda.

Niniejsze (niezbyt sensowne) zdanie zawiera jednoliterowy przyimek w bardzo niekorzystnym miejscu, a jednak wygląda nienajgorzej.

```
% mikrotypografia-spacje.tex
\documentclass{mwart}

\usepackage[utf8]{inputenc}
\usepackage{polski}
\usepackage{latex-ksiazka-kucharska}

\begin{document}
Niniejsze (niezbyt sensowne) zdanie zawiera
jednoliterowy przyimek w bardzo niekorzystnym miejscu,
a zatem trochę głupio wygląda.

Niniejsze (niezbyt sensowne) zdanie zawiera
jednoliterowy przyimek w bardzo niekorzystnym miejscu,
a~jednak wygląda nienajgorzej.
\end{document}
```

Jeżeli nie chcemy, aby wiersz został złamany w miejscu wystąpienia danego odstępu (np. pomiędzy słowami w wyrażeniu „a zatem”), powinniśmy zamiast odstępu użyć znaku tyldy (~).



## Kształt akapitu

Podstawową składową każdego dokumentu zawierającego tekst są akapity. Niewiele jest jednak dzieł (może poza powieściami) zawierających wyłącznie tekst ciągły. Umiejętność manipulowania ich właściwościami, takimi jak położenie, wyrównanie, wcinanie, czy otaczanie ramką pozwala znacząco przyspieszyć pracę.

Celem tego rozdziału jest przybliżenie Czytelnikowi podstawowych metod zarządzania akapitami w systemie  $\text{\LaTeX}$ .

## Usuwanie i wstawianie wcięcia akapitowego

Ten akapit ma normalne wcięcie. To oznacza, że jego pierwszy wiersz zaczyna się w pewnej odległości od lewego marginesu.

Ten akapit nie ma wcięcia. To oznacza, że jego pierwszy wiersz zaczyna się tuż przy lewym marginesie.

```
% akapit-wciecia.tex
\documentclass{mwart}
\usepackage[utf8]{inputenc}
\usepackage{polski}
\usepackage{latex-ksiazka-kucharska}

\begin{document}
\indent
Ten akapit ma normalne wcięcie. To oznacza, że jego
pierwszy wiersz zaczyna się w pewnej odległości od
lewego marginesu.

\noindent
Ten akapit nie ma wcięcia. To oznacza, że jego pierwszy
wiersz zaczyna się tuż przy lewym marginesie.
\end{document}
```

Standardowo  $\LaTeX$  rozpoczyna każdy akapit wcięciem akapitowym. W sytuacji, gdy nie chcemy, żeby tak było, możemy rozpocząć akapit poleceniem `\noindent`. W odwrotnym przypadku, to znaczy jeśli  $\LaTeX$  wcięcia *nie* robi, a zależy nam, żeby było inaczej, możemy je wstawić poleceniem `\indent` (w powyższym przykładzie jest ono zbędne – gdyby je usunąć,  $\LaTeX$  i tak roz-

```
\noindent
\indent
```

począłby akapit wcięciem). Najczęściej jednak nie ma potrzeby używania tych poleceń, ponieważ kwestię wcięć najlepiej pozostawić systemowi składu tekstów.

## Dokumenty bez wcięć akapitowych

Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
Ut purus elit, vestibulum ut, placerat ac.

Pellentesque habitant morbi tristique senectus et netus  
et malesuada fames ac turpis egestas. Mauris ut leo.

```
% akapit-bez-wciiec.tex
\documentclass{mwart}
\usepackage{latex-ksiazka-kucharska}
\usepackage{parskip}

\begin{document}
Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Ut purus elit, vestibulum ut, placerat ac.

Pellentesque habitant morbi tristique senectus et netus
et malesuada fames ac turpis egestas. Mauris ut leo.
\end{document}
```

Czasami zdarza się, że chcemy złożyć dokument, w którym akapity będą rozdzielone odstępem. W takim przypadku nie należy stosować wcięć akapitowych. Aby uzyskać taki wygląd dokumentu w  $\text{\LaTeX}$ u, należy załadować pakiet `parskip`.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.

Curabitur dictum gravida mauris. Ut purus elit, vestibulum ut, placerat ac

Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna.

Donec vehicula augue eu neque. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Mauris ut leo.

Rysunek 7. Wyciąg z dokumentu wynikowego

## Dodatkowy odstęp pionowy

Wyciąg z dokumentu wynikowego pokazano na rysunku 7, s. 76.

```
% akapit-odstep.tex
\documentclass{mwart}
\usepackage{latex-ksiazka-kucharska}

\begin{document}
Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Ut purus elit, vestibulum ut, placerat ac,
adipiscing vitae, felis.

\smallskip
Curabitur dictum gravida mauris. Ut purus elit, vestibulum
ut, placerat ac

\medskip
Nam arcu libero, nonummy eget, consectetur id,
vulputate a, magna.

\bigskip
Donec vehicula augue eu neque. Lorem ipsum dolor sit
amet, consectetur adipiscing elit.

\vspace{1.5cm}
Pellentesque habitant morbi tristique senectus et netus
et malesuada fames ac turpis egestas.

\vfill
Mauris ut leo.
\end{document}
```

W zasadzie lepiej unikać ręcznego wstawiania odstępów. Przykładowo, jeżeli chcemy uzyskać większy niż zwykle odstęp przed każdym paragrafem, zamiast ręcznie umieszczać odpowiedni kod przed poleceniem `\section`,

lepiej zmodyfikować wygląd *wszystkich* występujących w dokumencie paragrafów, korzystając z odpowiednich narzędzi (np. możliwości klas `mw*`, opisanych w ich dokumentacji). Zdarza się jednak potrzeba ingerencji w standardowy odstęp, jaki  $\LaTeX$  umieszcza np. przed akapitem wycentrowanym. Mamy wówczas do dyspozycji kilka poleceń. Możemy wstawić „mały”, „średni” i „duży” odstęp pionowy poleceniami `\smallskip`, `\medskip` i `\bigskip`. Jeżeli potrzebujemy odstępu innej wielkości (najczęściej większego), możemy skorzystać z polecenia `\vspace` (lub jego wariantu `\vspace*`, który – w przeciwieństwie do wersji bez gwiazdki – potrafi wstawić odstęp również na początku strony). Polecenie to przyjmuje jeden argument obowiązkowy, będący wyrażoną w pewnej mierze (np. pt, cm, em) wysokością odstępu. Wreszcie, w wyjątkowych sytuacjach może przydać się polecenie `\vfill`, które wstawia odstęp tak długi, jak tylko zmieści się na stronie. Jeżeli na jednej stronie umieścimy kilka poleceń `\vfill`, „podziela” się one równo dostępnym miejscem.

Warto pamiętać, że wszystkie instrukcje wstawiające odstępy pionowe najlepiej umieszczać *na początku nowego akapitu* (czyli na przykład po pustej linii w pliku źródłowym); umieszczenie ich wewnątrz akapitu może dać nieoczekiwane rezultaty.

```
\smallskip
\medskip
\bigskip
\vspace
\vspace*
\vfill
```

## Wyrównanie akapitu inaczej niż do obu marginesów

Poznań, 31.02.2015

Jerzy Brzęczyszczkiewicz  
ul. Akacjowa 555  
Psia Wólka

Podanie

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu

```
% akapit-wyrownanie.tex
\documentclass{mwart}
\usepackage[utf8]{inputenc}
\usepackage{polski}
\usepackage{latex-ksiazka-kucharska}

\begin{document}
\begin{flushright}
 Poznań, 31.02.2015
\end{flushright}

\begin{flushleft}
 Jerzy Brzęczyszczkiewicz\\
 ul. Akacyjowa 555\\
 Psia Wólka
\end{flushleft}

\begin{center}
 Podanie
\end{center}
Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Ut purus elit, vestibulum ut, placerat ac,
adipiscing vitae, felis. Curabitur dictum gravida
mauris. Nam arcu libero, nonummy eget, consectetur id.
\end{document}
```

Dość często zdarza się, że chcemy wycentrować pewne fragmenty tekstu; czasem też potrzeba dosunąć coś (na przykład datę) do prawego marginesu. Otoczenia `flushleft`, `flushright` i `center` sprawiają, że tekst w nich zawarty jest odpowiednio: wyrównany do lewego marginesu, wyrównany do prawego marginesu lub wycentrowany. Szczególnie przydatne jest wewnątrz tych otoczeń polecenie `\\`, które rozpoczyna nowy wiersz.

Warto też zauważyć, że środowiska `flushleft`, `flushright` i `center` wstawiają nieco światła przed i po swojej zawartości – zwykle nie ma więc potrzeby wstawiania przed nimi i po nich dodatkowych odstępów.

\\

## Akapity węższe niż kolumna

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.

Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id.

Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames.

```
% akapit-zwezenia.tex
\documentclass{mwart}
\usepackage{latex-ksiazka-kucharska}
\usepackage{lipsum}

\begin{document}
Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Ut purus elit, vestibulum ut, placerat ac,
adipiscing vitae, felis.

\begin{flushright}
\parbox{6cm}{Curabitur dictum
 gravida mauris. Nam arcu libero, nonummy eget,
 consectetur id.}
\end{flushright}

\begin{center}
\begin{minipage}{0.75\linewidth}
```



```

 Donec vehicula augue eu neque. Pellentesque
 habitant morbi tristique senectus et netus et
 malesuada fames.
 \end{minipage}
\end{center}

\end{document}

```

Nierzadko chcemy umieścić w dokumencie akapit węższy niż szerokość kolumny tekstu (np. mający lewą krawędź w połowie szerokości strony – typowym zastosowaniem jest nazwisko i adres odbiorcy oficjalnego pisma). Środowisko `flushright` do tego celu nie wystarczy, gdyż wyrównuje ono poszczególne wiersze do prawego marginesu zamiast wyrównać je do lewej krawędzi – tyle, że położonej w niestandardowym miejscu. W takiej sytuacji możemy wykorzystać polecenie `\parbox` lub środowisko `minipage`. Różnice między nimi są dość subtelne; nie wnikając w szczegóły powiedzmy tyle, że `\parbox` nadaje się do króciutkich, jednoakapitowych fragmentów, zaś `minipage` do większych, w szczególności zawierających dodatkowe elementy, jak wyliczenia czy przypisy. Jedno i drugie ma parametr obowiązkowy – szerokość składanego obszaru. Jak widać w przykładzie, można ją podać zarówno w centymetrach (czy innych jednostkach), jak i jako ułamek szerokości kolumny. Polecenie `\parbox` ma drugi parametr obowiązkowy, w którym umieszczamy wyrównywany tekst; w przypadku środowiska `minipage` oczywiście trafia on do wnętrza środowiska.

`\parbox`

Oto krótki utwór liryczny. Jak widać, jest on otoczony małymi odstępami pionowymi. Warto sprawdzić, co będzie, gdy spróbujemy umieścić w nim wiersz nie mieszczący się w szerokości kolumny tekstu.

Lorem ipsum dolor sit amet  
Consectetur adipiscing elit.  
Donec bibendum mauris,  
Nec neque tristique convallis  
Fusce eu vestibulum leo.

Poniżej znajduje się inny cytat, tym razem prozą. Jest on złożony z wcięciami po obu stronach oraz (również) małymi odstępami pionowymi przed i po cytatem.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec bibendum mauris nec neque tristique convallis. Fusce eu vestibulum leo. Fusce tincidunt, justo eget mollis malesuada, leo felis feugiat neque, sed rutrum ipsum enim commodo purus. Nunc tempus quam ut nisi egestas tempor. Fusce tincidunt, justo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec bibendum mauris nec neque tristique convallis. Fusce eu vestibulum leo.

Fusce tincidunt, justo eget mollis malesuada, leo felis feugiat neque, sed rutrum ipsum enim commodo purus. Nunc tempus quam ut nisi egestas tempor.

## Cytaty

Wyciąg z dokumentu wynikowego pokazano na *rysunku 8*, s. 82.

```
% akapit-cytaty.tex
\documentclass{mwart}
\usepackage[utf8]{inputenc}
\usepackage{polski}
\usepackage{latex-ksiazka-kucharska}

\begin{document}
Oto krótki utwór liryczny. Jak widać, jest on
otoczony małymi odstępami pionowymi. Warto sprawdzić,
co będzie, gdy spróbujemy umieścić w~nim wiersz nie
mieszczący się w~szerokości kolumny tekstu.

\begin{verse}
 Lorem ipsum dolor sit amet\\
 Consectetur adipiscing elit.\\
 Donec bibendum mauris,\\
 Nec neque tristique convallis\\
 Fusce eu vestibulum leo.
\end{verse}

Poniżej znajduje się inny cytat, tym razem prozą. Jest
on złożony z~wcięciami po obu stronach oraz (również)
małymi odstępami pionowymi przed i~po cytatem.
\begin{quote}
 Lorem ipsum dolor sit amet, consectetur adipiscing
 elit. Donec bibendum mauris nec neque tristique
 convallis. Fusce eu vestibulum leo. Fusce tincidunt,
 justo eget mollis malesuada, leo felis feugiat neque,
 sed rutrum ipsum enim commodo purus. Nunc tempus quam
 ut nisi egestas tempor. Fusce tincidunt, justo
\end{quote}
```

```
\begin{quotation}
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Donec bibendum mauris nec neque tristique convallis.
Fusce eu vestibulum leo.

Fusce tincidunt, justo eget mollis malesuada, leo felis
feugiat neque, sed rutrum ipsum enim commodo purus. Nunc
tempus quam ut nisi egestas tempor.
\end{quotation}
\end{document}
```

Niekiedy zachodzi potrzeba umieszczenia w dokumencie cytatu (lub po prostu wyróżnionego, nieco dłuższego fragmentu). Standardowym sposobem w  $\text{\LaTeX}$ u jest środowisko `quote` (dla krótszych, jednoakapitowych fragmentów) lub `quotation` (do fragmentów liczących więcej niż jeden akapit). Wyróżnione nimi fragmenty są otoczone niewielkimi odstępami pionowymi oraz wcięte po lewej i prawej stronie na szerokość wcięcia akapitowego.

W przypadku fragmentów pisanych wierszem (co może oznaczać zarówno poezję, jak i inne teksty, w których podział na wersy jest istotny) można też użyć środowiska `verse` i oznaczać koniec każdego wersu poleceniem `\\`. Działa ono bardzo podobnie do `quote`, ale z następującą różnicą: gdy któryś z wersów nie mieści się w linii, jest kontynuowany od nowej, ale z wcięciem. Ten sposób pozwala odróżnić tekst znajdujący się w nowej linii dlatego, że jest kolejnym wersem, od tekstu znajdującego się w nowej linii dlatego, że nie zmieścił się w poprzedniej.

Na koniec dodajmy, że umieszczenie polecenia `\\` na końcu *ostatniego* wersu w środowisku `verse` nie jest konieczne, ale w niczym nie przeszkadza.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi

dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum

urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci

## Skład dwułamowy

Wyciąg z dokumentu wynikowego pokazano na *rysunkach 9–11*, s. 85–87.

```
% akapit-dwułamowy.tex
\documentclass[twocolumn]{mwart}
\usepackage{latex-ksiazka-kucharska}
\usepackage{lipsum}

\begin{document}
\lipsum[1]

\onecolumn
\lipsum[1]

\twocolumn[{\lipsum[4]}]
\lipsum[1]
\end{document}
```

Może się zdarzyć, że będziemy chcieli stworzyć dokument złożony w dwóch łamach. [Najprostszym sposobem na osiągnięcie tego celu jest wykorzystanie parametru opcjonalnego twocolumn standardowych klas L<sup>A</sup>T<sub>E</sub>Xa \(lub klas mw\\*\)](#); ponadto, można przełączyć się na skład jednołamowy poleceniem `\onecolumn` i z powrotem na dwułamowy poleceniem `\twocolumn`. Polecenie `\twocolumn` przyjmuje parametr opcjonalny, którym jest tekst składany w jednym łamie *przed* resztą, składaną w dwóch. Niestety, nie jest to sposób najlepszy. Po pierwsze, każde takie polecenie automatycznie rozpoczyna nową stronę, co nie zawsze jest tym, czego oczekujemy. Po drugie, na ostatniej stronie rozdziału (czy całego dzieła) kolumny nie są wyrównywane. Ten ostatni problem naprawiają np. pakiety `balance` oraz `flushend`. Niestety, działanie jednego i drugiego pozostawia nieco do życzenia.

Jak widać, skład dwułamowy nie jest (niestety) mocną stroną L<sup>A</sup>T<sub>E</sub>Xa. (Ma się to zmienić wraz z nadejściem wersji L<sup>A</sup>T<sub>E</sub>X3, na którą jednak będzie trzeba jeszcze trochę poczekać...) Na szczęście istnieje pakiet, który rozwiązu-

Zwanych popularnie, choć błędnie, *kolumnami*.

```
\onecolumn
\twocolumn
```



je oba powyższe problemy (ale za to dostarcza własnych). Nazywa się on `multicol`, a przykład jego użycia znajduje się na poniżej.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris.

|                                                                                                        |                                                                                                           |
|--------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|
| Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pel- | lentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. |
|--------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|

Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat.

```
% akapit-multicol.tex
\documentclass{mwart}
\usepackage{latex-ksiazka-kucharska}
\usepackage{multicol}
\usepackage{lipsum}

\begin{document}
Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Ut purus elit, vestibulum ut, placerat ac,
adipiscing vitae, felis. Curabitur dictum gravida
mauris.
\begin{multicols}{2}
 Nam arcu libero, nonummy eget, consectetur id,
```

```

vulputate a, magna. Donec vehicula augue eu
neque. Pellentesque habitant morbi tristique senectus
et netus et malesuada fames ac turpis egestas. Mauris
ut leo.
\end{multicols}
Cras viverra metus rhoncus sem. Nulla et
lectus vestibulum urna fringilla ultrices. Phasellus eu
tellus sit amet tortor gravida placerat.
\end{document}

```

## Otoczanie fragmentu tekstu ramką

>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.

Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id.

Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada.

```

% akapit-ramka.tex
\documentclass{mwart}
\usepackage{latex-ksiazka-kucharska}
\usepackage{mdframed}
\usepackage{lipsum}

\begin{document}

```

```
>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Ut purus elit, vestibulum ut, placerat ac,
adipiscing vitae, felis.
```

```
\begin{mdframed}
Curabitur dictum gravida mauris. Nam arcu libero,
nonummy eget, consectetur id.
\end{mdframed}
```

```
Donec vehicula augue eu neque. Pellentesque
habitant morbi tristique senectus et netus et
malesuada.
```

```
\end{document}
```

Czasem zachodzi potrzeba umieszczenia pewnego fragmentu pracy (akapitu lub kilku) w ramce. Na szczęście taki efekt łatwo otrzymać, korzystając z otoczenia `mdframed` dostarczanego wraz z pakietem o tej samej nazwie.

Domyślnie zawartość otoczenia `mdframed` umieszczana jest w czarnej ramce. Sam pakiet umożliwia jednak modyfikowanie wielu parametrów, takich jak choćby obramowania, tła, marginesy wewnętrzne i zewnętrzne, czy cienie. Szczegółowy opis metod wpływania na te cechy opisany jest w dokumentacji pakietu.

# Automatyzacja

Jedną z istotnych zalet  $\text{\LaTeX}$ a jest automatyzacja niektórych czynności, w szczególności (choć nie tylko) związanych z numerowaniem różnych obiektów. W tym rozdziale przyjrzymy się poleceniom pomagającym zrzucić z siebie jarzmo nudnej, mechanicznej pracy.

## Odwołania do elementów numerowanych

### 1. Pierwszy paragraf

### 2. Drugi paragraf

Zobacz paragraf 1 na stronie 1.

```
% automatyzacja-odwolania.tex
\documentclass{mwart}
\usepackage{latex-ksiazka-kucharska}
```

```

\begin{document}
\section{Pierwszy paragraf}
\label{sec:pierwszy}

\section{Drugi paragraf}
Zobacz paragraf~\ref{sec:pierwszy} na
stronie~\pageref{sec:pierwszy}.
\end{document}

```

Polecenie `\label` nadaje etykietę numerowi obiektu, po którym zostało użyte. Może to być rozdział, paragraf, wzór, rysunek, tabela – cokolwiek, co jest przez  $\LaTeX$ a automatycznie numerowane. Do tego numeru można się odwoływać w innych miejscach dokumentu poleceniem `\ref`, które wstawia numer zapamiętanego obiektu, oraz `\pageref`, które wstawia numer strony, na której tenże obiekt wystąpił.

Etykieta nie powinna zawierać znaków specjalnych, a duże i małe litery są odróżniane – zatem `\label{foo}` i `\label{FOO}` definiują różne etykiety. Nie trzeba chyba dodawać, że etykiety powinny być unikatowe – w jednym dokumencie żadna etykieta nie powinna się pojawić dwukrotnie.

`\label`

`\ref`

`\pageref`

Tzn. może zawierać niektóre z nich, ale zamiast pamiętać, które się „nadają”, prościej nie używać ich wcale.

## Indeks pojęć

### D

definicja, 1

### P

$\pi$ , 1

pojęcie

ważne, 1

### S

s, 1

### Ś

ś, 1

### T

t, 1

### W

ważne pojęcie, *zob.* poję-  
cie, ważne

Rysunek 12. Wyciąg z dokumentu wynikowego (strona 3)

## Skorowidze

Wyciąg z dokumentu wynikowego pokazano na *rysunku 12*, s. 94.

```
% automatyzacja-skorowidz.tex
\documentclass{mwbk}
\usepackage[utf8]{inputenc}
\usepackage{polski}
\usepackage{latex-ksiazka-kucharska}
\usepackage{makeidx}
\makeindex

\begin{document}
\chapter{Rozdział pierwszy}
W~tym rozdziale definiujemy
\index{definicja}
\index{pojęcie!ważne}%
\index{ważne pojęcie|see{pojęcie, ważne}}%
ważne pojęcie, oznaczane symbolem
\index{pi@π}π. Będzie ono przedmiotem naszych
rozważań w~dalszym ciągu.

\renewcommand{\indexname}{Indeks pojęć}
\printindex
\end{document}
```

Sporządzanie skorowidza jest tematem złożonym; na szczęście  $\text{\LaTeX}$  ułatwia nam go tak bardzo, jak tylko komputer może to ułatwić.

Aby umieścić skorowidz w dokumencie, należy wykonać kilka czynności wstępnych. Pierwszą z nich jest załadowanie pakietu `makeidx` i użycie w preambule polecenia `\makeindex`. Drugą – wydanie polecenia `\printindex` w miejscu, gdzie skorowidz ma się pojawić (prawdopodobnie w okolicach końca dokumentu). Niestety, na tym proste czynności się kończą, bowiem trzecia z nich to samo przygotowanie skorowidza.

```
\makeindex
\printindex
```

Przygotowanie skorowidza polega na umieszczeniu w miejscach, do których powinny się w nim znaleźć odwołania, polecenia `\index`. Polecenie to w najprostszej postaci ma jeden argument – hasło do skorowidza.

Może się jednak zdarzyć, że z różnych powodów ta prosta forma polecenia `\index` nam nie wystarczy. Jedną z takich sytuacji jest ta, w której indeksowane hasło zawiera np. formuły matematyczne, co może spowodować problemy przy sortowaniu alfabetycznym. W takim przypadku możemy użyć formy `\index{<klucz>@<hasło>}`; w indeksie, w miejscu, w którym (zgodnie z porządkiem alfabetycznym) pojawiłby się `<klucz>`, pojawi się wówczas `<hasło>`.

Często spotyka się też indeks wielopoziomowy (choć więcej niż dwa poziomy wydają się w większości przypadków zbędną ekstrawagancją). Wówczas hasło „podstawowe” od „podhasła” oddzielamy wykrzyknikiem.

Wreszcie, może się zdarzyć, że będziemy chcieli umieścić w indeksie odwołanie nie do tekstu głównego, ale do innego hasła. W takim przypadku używa się (nieco dziwnej) składni `\index{<hasło1> | see{<hasło2>}}`.

Dla porządku odnotujmy jeszcze, że tytuł rozdziału ze skorowidzem (domyślnie „Index” lub „Skorowidz”, gdy używamy pakietu `polSKI`) można zmienić poleceniem `\renewcommand`, definiując ponownie polecenie `\indexname`. W powyższym przykładzie zrobiliśmy to tuż przed wywołaniem polecenia `\printindex`.

Niestety, na tym praca autora skorowidza się nie kończy (choć pozostał już tylko jeden drobiazg).  $\LaTeX$  nie potrafi (niestety) sam posortować skorowidza alfabetycznie. Do sortowania skorowidzy w języku angielskim służy program `makeindex`. Jeśli chcemy posortować skorowidz polski, najwygodniej użyć programu `texindy`.

```
texindy <plik>.idx -L polish -C <kodowanie>
```

gdzie `<kodowanie>` to np. `utf8` lub `cp1250`. Cały cykl pracy wygląda więc następująco: przetwarzamy plik  $\LaTeX$ em (być może więcej niż jeden raz, by zadziałały odwołania i spis treści), następnie sortujemy skorowidz programem `texindy`, wreszcie raz jeszcze wywołujemy  $\LaTeX$ a, aby umieścić posortowany skorowidz w dokumencie.

```
\index
```

*De facto* oznacza to, że w odpowiednim miejscu w indeksie zostanie wywołane polecenie `\see` z odpowiednim parametrem.

```
\renewcommand
```

```
\indexname
```

Instalacja i konfiguracja programu `texindy` opisana jest w *Dodatku C: Środowisko pracy* do wersji elektronicznej tej książki.

Spis haseł do skorowidza  $\LaTeX$  umieszcza w pliku z rozszerzeniem `idx`; posortowana lista trafia do pliku z rozszerzeniem `ind`.



## Spis treści

|                                          |   |
|------------------------------------------|---|
| <b>Rozdział 1. Lorem ipsum</b> . . . . . | 3 |
| 1.1. Nam dui ligula . . . . .            | 3 |
| 1.1.1. Nulla malesuada . . . . .         | 3 |
| <b>Lorem ipsum</b> . . . . .             | 5 |
| Nam dui ligula . . . . .                 | 5 |
| Nulla malesuada . . . . .                | 5 |

Rysunek 13. Wyciąg z dokumentu wynikowego (strona 1)

## Spis treści i inne spisy

Wyciąg z dokumentu wynikowego pokazano na *rysunku 13*, s. 97.

```
% automatyzacja-spis.tex
\documentclass{mwbk}
\usepackage{polski}
\usepackage{latex-ksiazka-kucharska}

\begin{document}
\tableofcontents
\chapter{Lorem ipsum}
Lorem ipsum dolor sit amet.

\section{Nam dui ligula}
Nam dui ligula, fringilla a, euismod sodales.

\subsection{Nulla malesuada}
Nulla malesuada porttitor diam.

\chapter*{Lorem ipsum}
Lorem ipsum dolor sit amet.

\section*{Nam dui ligula}
Nam dui ligula, fringilla a, euismod sodales.

\subsection*{Nulla malesuada}
Nulla malesuada porttitor diam.
\end{document}
```

Dzięki znakowaniu semantycznemu, o którym mówiliśmy w pierwszej części książki,  $\text{\LaTeX}$  wie, jak brzmia tytuły rozdziałów czy paragrafów. Nie wiadać więc powodu, dla którego mielibyśmy ręcznie przygotowywać spis treści. Wprawdzie sposobów, na jakie można opracować *wygląd* takiego spisu są pewnie setki, poprzestaniemy tutaj jednak na standardowym układzie gra-

ficznym proponowanym przez twórców  $\LaTeX$ a. Dzięki temu oszczędzimy też sobie pracy przy jego modyfikowaniu. (Gdybyśmy jednak bardzo chcieli zmienić wygląd spisu treści, możemy w tym celu użyć np. pakietu `tocloft`.) Spis treści umieszcza się w wybranym miejscu (zazwyczaj na początku książki lub artykułu) poleceniem `\tableofcontents`.

Zwróćmy uwagę, że spis treści obejmuje domyślnie rozdziały, paragrafy i podparagrafy, zaś jednostki niższego rzędu już się w nim nie pojawiają.

Wprawdzie nie trzeba wiedzieć dokładnie, w jaki sposób  $\LaTeX$  tworzy spis treści, należy jednak pamiętać o podstawowej zasadzie. Na początku dokumentu, gdzie spis treści ma być umieszczony,  $\LaTeX$  nie wie jeszcze, na jaką stronę trafi który rozdział. To oznacza, że w celu umieszczenia spisu treści  $\LaTeX$  korzysta z informacji zapisanych podczas poprzedniego przebiegu kompilacji. Oznacza to, że aby uzyskać poprawny spis treści, należy skompilować plik dwukrotnie, a czasami nawet trzykrotnie.

Podobnie jak spis treści można umieścić w dokumencie spisy tabel czy rysunków. Służą do tego polecenia `\listoftables` oraz `\listoffigures`, które należy umieścić w miejscu, gdzie mają się one pojawić.

```
\tableofcontents
```

Można to zmienić poleceniem typu `\setcounter{tocdepth}{5}`.

Jeżeli spis treści znajduje się na początku dokumentu i zajmuje więcej niż jedną stronę, po drugiej kompilacji, gdy  $\LaTeX$  po raz pierwszy włącza spis treści do dokumentu, cały następujący po nim tekst przesuwany do przodu o jedną lub więcej stron, zatem potrzeba kolejnej kompilacji, by numery stron w spisie treści odpowiadały faktycznej zawartości pliku.

```
\listoftables
```

```
\listoffigures
```

## Bibliografie

Książka [1] zawiera sporo ciekawych rzeczy, zaś – dla odmiany – artykuł [2] zawiera głównie bzdury.

### Bibliografia

- [1] Marcin Borkowski i Bartłomiej Przybylski. *LaTeX. Książka kucharska*. Warszawa: PTM, 2015.
- [2] Alan D. Sokal. “Transgressing the Boundaries: Toward a Transformative Hermeneutics of Quantum Gravity”. W: *Social Text* 46/47 (1996), s. 217–252.

```
% automatyzacja-bibliografia.tex
\documentclass{mwart}
\usepackage[utf8]{inputenc}
\usepackage{polski}
\usepackage{latex-ksiazka-kucharska}
\usepackage[language=polish]{biblatex}
\addbibresource{automatyzacja-bibliografia.bib}

\begin{document}
Książka~\cite{latex-kk} zawiera sporo ciekawych rzeczy,
zaś~\ppauza dla odmiany~\ppauza artykuł~\cite{sokal}
zawiera głównie bzdury.

\printbibliography
\end{document}
```

```
% automatyzacja-bibliografia.bib
@article{sokal,
 author = {Sokal, Alan D.},
 title = {Transgressing the Boundaries: Toward
 a Transformative Hermeneutics of
 Quantum Gravity},
 journaltitle = {Social Text},
 date = 1996,
 volume = {46/47},
 pages = {217-252},
}

@book{latex-kk,
 author = {Borkowski, Marcin and Przybylski,
 Bartłomiej},
 title = {\LaTeX. Książka kucharska},
 date = 2015,
 publisher = {PTM},
 location = {Warszawa},
}
```

Ważną częścią wielu publikacji jest bibliografia czy spis literatury. Jak wiadomo każdemu, kto kiedykolwiek taki spis sporządził, jego przygotowanie jest żmudną i nie nazbyt ciekawą pracą. Na szczęście  $\LaTeX$  zapewnia narzędzia, które wprowadzicie nie czynią tej pracy fascynującą, ale na pewno mniej żmudną. Dzięki temu bibliografia ma też szansę zawierać mniej błędów – komputery myślą się dużo rzadziej niż ludzie wykonujący bezmyślną pracę.

Istnieją przynajmniej trzy narzędzia  $\LaTeX$ owe wspomagające proces sporządzania bibliografii. Najstarszym jest `bibtex`, program napisany w latach osiemdziesiątych ubiegłego stulecia przez Orena Patashnika. Niestety, narzędzie to nie bardzo pozwala na lokalizację spisów literatury, co raczej wyklucza je z obszarów zastosowań, w których język dokumentu jest inny niż angielski. Nowszym narzędziem, powstałym na początku obecnego wieku, jest `amsrefs` – pakiet służący do składu bibliografii i cytowań, napisany przez Michaela Downesa i rozwijany po jego śmierci przez Davida M. Jonesa, a fir-

mowany przez *American Mathematical Society*. Jest to bardzo dobry pakiet, którego nie opiszemy tylko dlatego, że zajmiemy się najnowszym (pierwsza stabilna wersja została opublikowana w 2010 roku) i obecnie najlepszym pakietem obsługującym bibliografię o nazwie `biblatex`.

Pewien problem z pakietem `biblatex` polega na tym, że jest on rozbudowanym oraz skomplikowanym narzędziem z wieloma możliwościami dostosowania do najrozmaitszych sposobów cytowania i tworzenia bibliografii. My nie będziemy analizować jego bardziej zaawansowanych możliwości, odsyłając zainteresowanych do jego bogatej dokumentacji.

Tworzenie bibliografii przy użyciu pakietu `biblatex` polega na przygotowaniu pliku z *bazą danych* cytowanych źródeł. Plik ten powinien mieć rozszerzenie `.bib` i format podobny, jak to widzimy w przykładzie. Warto zauważyć kilka spraw. Po pierwsze, każda pozycja w spisie literatury ma *etykietę*, której używa się później do jej cytowania. Etykietami bibliografii rządzą te same zasady, co każdymi innymi (por. s. 92).

Po drugie, w przypadku cytowania książki, artykułu czy innego źródła mającego więcej niż jednego autora, oddzielamy autorów ciągiem znaków `„and”` (słowo „and” otoczone z obu stron znakiem odstępu). W przypadku, gdyby słowo „and” (ze spacjami) było częścią nazwiska autora (co jest możliwe np. w przypadku, gdy jako autor figuruje instytucja, w której nazwie to słowo występuje), najprościej całą tę nazwę wziąć w nawiasy klamrowe.

Po trzecie, dane autora najlepiej wpisywać w formie „Nazwisko, Imię”, przy czym zarówno nazwisko, jak i imię może składać się z kilku części.

Po czwarte, zwróćmy uwagę, że podając opcję `[language=polish]` pakietowi `biblatex` automatycznie uzyskujemy spolszczoną bibliografię.

Po piąte, aby uzyskać bibliografię, w pliku  $\LaTeX$ owym musimy umieścić kilka rzeczy. Dołączamy pakiet `biblatex`, deklarujemy nazwę pliku z bazą danych o cytowanych pozycjach (polecenie `\addbibresource`), wreszcie umieszczamy samą bibliografię (polecenie `\printbibliography`). Cytowane w tekście prace oznaczamy poleceniem `\cite`, którego argumentem jest nadana wcześniej (w pliku `.bib`) etykieta.

Po szóste,  $\LaTeX$  niestety sam nie potrafi przygotować bibliografii – trzeba mu w tym pomóc. Po dokonaniu kompilacji wywołujemy program `biber`. Narzędzie to zajmie się m.in. wyłuskaniem z bibliografii pozycji, które faktycz-

Format nazwiska jest tak naprawdę dużo bardziej skomplikowany i pozwala na używanie przyrostków (jak *von*) i przyrostków (jak *jr*), ale te subtelnosci pominiemy.

```
\addbibresource
\printbibliography
\cite
```

Opis instalacji i konfiguracji tego narzędzia znajduje się w *Dodatku C: Środowisko pracy*, dołączonym do wersji elektronicznej tej książki.

nie były cytowane i posortowaniem ich alfabetycznie. Po ponownej kompilacji  $\LaTeX$ em otrzymujemy plik PDF z bibliografią.

Możliwości pakietu `biblatex` są dużo, dużo większe. Możemy przy jego użyciu m.in. umieścić w dokumencie kilka bibliografii czy stosować inny niż liczby w nawiasach kwadratowych styl cytowania; te zagadnienia wykraczają jednak poza ramy niniejszej książeczki.

## Tabele

Skład tabel jest kwestią złożoną. Pierwszym problemem jest samo przygotowanie tabeli; drugim, umieszczenie jej na stronie, z podpisem (automatycznie numerowanym), w odpowiednim miejscu. Ponieważ tabela powinna mieścić się na jednej stronie (o ile to możliwe), nie zawsze można umieścić ją w miejscu, w którym polecenia za nią odpowiedzialne są umieszczone w pliku źródłowym. Na przykład, jeżeli do końca strony zostało, powiedzmy, pięć wierszy, a autor w tym miejscu umieści kod tabeli wysokości sześciu wierszy, możemy albo złożyć ją od nowej strony (co skutkuje brzydką, białą plamą na poprzedzającej stronie), albo przenieść ją w jakieś inne, rozsądne miejsce (np. na górę strony).  $\LaTeX$  stosuje to drugie rozwiązanie w sposób automatyczny, co z jednej strony jest bardzo wygodne, z drugiej zaś nie zawsze jest łatwe do kontrolowania (niejeden temat na forach dotyczących  $\LaTeX$ a zaczyna się od pytania „co zrobić, żeby moje tabele były tam, gdzie ja chcę, a nie gdzie chce  $\LaTeX$ ?”). W tym rozdziale poruszone zostały liczne zagadnienia związane z tą tematyką.



## Najprostsza tabela

| foobar | barbaz | bazqux |
|--------|--------|--------|
| 123    | 456    | 789    |
| abc    | def    | ghi    |

```
% table-prosta.tex
\documentclass{mwart}
\usepackage{latex-ksiazka-kucharska}
\usepackage{booktabs}

\begin{document}
\begin{center}
\begin{tabular}{lcr}
\toprule
foobar & barbaz & bazqux & \\
\midrule
123 & 456 & 789 & \\
abc & def & ghi & \\
\bottomrule
\end{tabular}
\end{center}
\end{document}
```

Chociaż  $\text{\LaTeX}$  umożliwia składanie tabel bez wykorzystania dodatkowych pakietów, najlepszym (z powodów estetycznych) narzędziem do składu tabel jest pakiet `booktabs`. Zwróćmy uwagę na trzy cechy, które odróżniają ładne tabele od brzydkich: po pierwsze, w ładnej tabeli nie powinno być linii pionowych; po drugie, w ładnej tabeli nie powinno być linii podwójnych; po trzecie,

ładna tabela nie może się dusić – potrzebuje dużo światła (zarówno między wierszami, jak i kolumnami). Pakiet `booktabs` pomaga nam w przestrzeganiu tych zasad.

Jak widać, przykładowa tabela jest wycentrowana (odpowiada za to oczywiście środowisko `center`) – teoretycznie nie musi tak być (choć nie widać powodów, żeby tabele wyrównywać do lewego marginesu). Za samą tabelę odpowiada środowisko `tblar`, którego parametr obowiązkowy opisuje strukturę wiersza tabeli. Nasza tabela składa się z trzech kolumn, pierwszej wyrównanej do lewej strony (`l`), drugiej wycentrowanej (`c`) i trzeciej wyrównanej do prawej strony (`r`). Linie składamy poleceniami `\toprule`, `\midrule` i `\bottomrule`; jak widać, górna i dolna linia są nieco szersze, różnią się też ilością wolnego miejsca przed i po linii (co łatwo sprawdzić zamieniając polecenia `\toprule` i `\bottomrule` miejscami). Każdy wiersz tabeli kończy się poleceniem `\\`, zaś poszczególne komórki w ramach jednego wiersza oddziela się etką; [przed i po poleceniu `\\`](#) i znaku `&` można dodać dowolną liczbę spacji, dzięki czemu kod tabeli w pliku źródłowym będzie bardziej czytelny (czy – realistycznie patrząc – raczej mniej nieczytelny...).

```
\toprule
\midrule
\bottomrule
```

```
\\
```

nazwa tego symbolu bierze się stąd, że stanowi on *ligaturę*, czyli połączone kształty liter „e” oraz „t” – co widać jeszcze w niektórych fontach. Po łożeniu etki znaczy „i”.

## Komórki wielowierszowe

| Liczba | Komentarz                                                                 |
|--------|---------------------------------------------------------------------------|
| 1      | Jedynka to interesująca liczba, gdyż nie jest ani pierwsza, ani złożona.  |
| 2      | Dwójka to interesująca liczba, gdyż jest jedyną parzystą liczbą pierwszą. |

```
% tabele-wielowierszowe.tex
\documentclass{mwart}
\usepackage{latex-ksiazka-kucharska}
\usepackage[utf8]{inputenc}
\usepackage{polski}
\usepackage{booktabs}

\begin{document}
\begin{center}
\begin{tabular}{cp{5cm}}
\toprule
Liczba & Komentarz \\
\midrule
1 & Jedyńska to interesująca liczba, gdyż nie jest
ani pierwsza, ani złożona. \\
2 & Dwójka to interesująca liczba, gdyż jest jedyną
parzystą liczbą pierwszą. \\
\bottomrule
\end{tabular}
\end{center}
\end{document}
```

Niekiedy zawartość komórki tabeli jest długa, i jest sens rozdzielić ją na dwa lub więcej wierszy. Zamiast używać kodów `lcr`, oznaczających odpowiednio kolumny wyrównane do lewej, wycentrowane i wyrównane do prawej, można wówczas użyć kodu `p{...}`, który nakazuje  $\LaTeX$ owi złożyć kolumnę zadanej szerokości, w której komórki mogą składać się z wielu wierszy.

## Gdzie jest moja tabelka?

Tabela 1. Przykładowa tabela

| foobar | barbaz | bazqux |
|--------|--------|--------|
| 123    | 456    | 789    |
| abc    | def    | ghi    |

W tabeli 1 można znaleźć pewne dane, które z pewnością wstrząsną czytelnikiem.

### Spis tabel

|                                 |   |
|---------------------------------|---|
| 1. Przykładowa tabela . . . . . | 1 |
|---------------------------------|---|

```
% tabele-gdzie.tex
\documentclass{mwart}
\usepackage[utf8]{inputenc}
\usepackage{polski}
\usepackage{latex-ksiazka-kucharska}
\usepackage{booktabs}

\begin{document}
W~tabeli~\ref{tab:przykladowa-tabela} można znaleźć
pewne dane, które z~pewnością wstrząsną czytelnikiem.
\begin{table}
\centering
\caption{Przykładowa tabela}
```

```

\label{tab:przykladowa-tabela}
\begin{tabular}{lcr}
\toprule
foobar & barbaz & bazqux \\
\midrule
123 & 456 & 789 \\
abc & def & ghi \\
\bottomrule
\end{tabular}
\end{table}

\listoftables
\end{document}

```

Jak już zauważyliśmy we wstępie do tego rozdziału, pewnym problemem jest znalezienie dla tabeli właściwego miejsca w dokumencie; jeśli tak się złoży, że kod tabeli wypadnie w miejscu znajdującym się blisko dołu strony, tabela powinna „przepełnąć” gdzieś indziej. Z tego powodu tabele (a także np. rysunki) są w  $\text{\LaTeX}$ u realizowane przez tzw. „elementy pływające” (ang. *floats*). „Pływającą” tabelę otaczamy środowiskiem `table`. Ponieważ dba ono o właściwe odstępy pionowe nad i pod tabelą, nie należy wewnątrz niego używać środowiska `center` (które wprowadza dodatkowo własne odstępy); zamiast tego, centrujemy tabelę poleceniem `\centering`.

`\centering`

Polecenie `\listoftables`, wspomniane na s. 99, pozwala łatwo uzyskać spis wszystkich tabel umieszczonych w dokumencie przy użyciu środowiska `table`.

Ponieważ tabela może znajdować się nieco dalej, niż miejsce, w którym została wstawiona, logiczne jest, że należy ją zaopatrzyć w podpis. Służy do tego polecenie `\caption`. Oczywiście, podpisy są (standardowo) automatycznie numerowane, a skoro tabele mają *numery*, można się na nie *powoływać*. Zaraz po poleceniu `\caption` można wstawić polecenie `\label`, i następnie używać komend `\ref` oraz `\pageref` do wstawiania numeru tabeli i numeru strony, na której ona występuje.

`\caption`

Oczywiście, czasami chcemy, aby tabela jednak znalazła się dokładnie w tym miejscu, w którym chcemy, a nie tam, gdzie umieści ją  $\text{\LaTeX}$  – nie za-

wsze umieszcza on tabele dokładnie tam, gdzie życzy sobie tego użytkownik. W takiej sytuacji można środowisku `tab le` podać dodatkowy argument opcjonalny `h` (gdy chcemy, by tabela znalazła się właśnie w tym miejscu, gdzie została wpisana do pliku źródłowego). Można też użyć litery `b` (ang. *bottom*), `t` (ang. *top*) lub `p` (ang. *page*), gdy chcemy, by tabela znalazła się odpowiednio: na dole strony, na górze strony lub na osobnej stronie.

Można też podać kilka liter; np. argument `ht` oznacza, że tabela ma być w tym miejscu, gdzie jest środowisko `tab le`, a gdyby to było niemożliwe – bo np. miejsce to wypada na samym dole strony, a tabela jest spora – wówczas chcemy, by znalazła się na górze strony.

W tabeli 1 można znaleźć ważne dane.

Tabela 1. Przykładowa tabela

|     |     |     |
|-----|-----|-----|
| 123 | 456 | 789 |
|-----|-----|-----|

```
% tabela-gdzie-2.tex
\documentclass{mwart}

\usepackage[utf8]{inputenc}
\usepackage{polski}
\usepackage{latex-ksiazka-kucharska}
\usepackage{booktabs}

\begin{document}
W~tabeli~\ref{tab:przykladowa-tabela} można znaleźć
ważne dane.
\begin{table}[htbp]
```

```

\centering
\caption{Przykładowa tabela}
\label{tab:przykladowa-tabela}
\begin{tabular}{rrr}
\toprule
123 & 456 & 789 \\
\bottomrule
\end{tabular}
\end{table}
\end{document}

```

## Tabele nienumerowane

Przykładowa tabela bez numeru

|     |     |
|-----|-----|
| 123 | 456 |
| abc | def |

Tabela 1. Przykładowa tabela z numerem i kropką po nim

|     |     |
|-----|-----|
| 123 | 456 |
| abc | def |

A tu reszta dokumentu.

```

% tabele-nienumerowane.tex
\documentclass{mwart}

```

```
\usepackage[utf8]{inputenc}
\usepackage{polski}
\usepackage{latex-ksiazka-kucharska}
\usepackage{booktabs}
\usepackage{ccaption}
\captiontitlefont{\small}
\captiondelim{. }

\begin{document}
\begin{table}
 \centering
 \legend{Przykładowa tabela bez numeru}
 \begin{tabular}{cc}
 \toprule
 123 & 456 \\
 abc & def \\
 \bottomrule
 \end{tabular}
\end{table}

\begin{table}
 \centering
 \caption{Przykładowa tabela z~numerem i~kropką po nim}
 \begin{tabular}{cc}
 \toprule
 123 & 456 \\
 abc & def \\
 \bottomrule
 \end{tabular}
\end{table}
A~tu reszta dokumentu.
\end{document}
```

Standardowe podpisy są automatycznie numerowane i zawierają słowo „tabela”, po którym następuje numer. Możemy jednak chcieć umieścić w naszym dokumencie tabelę nienumerowaną (na przykład wówczas, gdy jest to jedyna tabela w całym tekście). Jednym ze sposobów jest użycie (zamiast



polecenia `\caption`) polecenia `\legend` z pakietu `ccaption`. Pakiet ten pozwala na różnorodną modyfikację wyglądu podpisów. Niestety, niweczy on wysiłek klasy `mwart` (czy `mwbk`), która pilnuje, by po numerze tabeli następowała (polskim zwyczajem) kropka, a nie dwukropek. Nic straconego: pakiet `ccaption` pozwala to zmienić przy użyciu polecenia `\captiondelim`.

Warto przestudiować jego dokumentację, jeżeli standardowy wygląd podpisów tabel czy grafik nam nie odpowiada. Istnieje też konkurencyjny pakiet `caption`, również pozwalający na daleko idące modyfikacje wyglądu podpisów.

`\legend``\captiondelim`

## Łączenie komórek w poziomie

|        |        |        |
|--------|--------|--------|
| foobar | barbaz | bazqux |
| 123    | 456    | 789    |
| abc    | qwerty |        |

```
% table-poziom.tex
\documentclass{mwart}
\usepackage{latex-ksiazka-kucharska}
\usepackage{booktabs}

\begin{document}
\begin{center}
\begin{tabular}{lcr}
\toprule
foobar & barbaz & bazqux\\
\midrule
\end{tabular}
\end{center}
\end{document}
```

```

123 & 456 & 789 \\
\cmidrule(lr){2-3}
abc & \multicolumn{2}{c}{qwerty} \\
\bottomrule
\end{tabular}
\end{center}
\end{document}

```

Stosunkowo często zachodzi potrzeba *połączenia komórek* w tabeli. Łączenie komórek w poziomie (tj. łączenie kilku komórek leżących w tym samym wierszu) jest proste; wystarczy w odpowiednim miejscu wstawić polecenie `\multicolumn`. Przyjmuje ono trzy parametry: liczbę komórek do połączenia, styl komórki powstałej z połączenia (np. `l`, `c` lub `r`) i zawartość połączonych komórek. W naszym przykładzie łączyliśmy dwie komórki, tekst w powstałej w ten sposób komórce centrujemy oraz umieszczamy tam słowo „qwerty”.

`\multicolumn`

Jak widać, możemy też narysować linię poziomą szerokości części tabeli. Służy do tego polecenie `\cmidrule` z pakietu `booktabs`. Ma ono dość niestandardową składnię, czyli parametr opcjonalny w *nawiasach okrągłych*. Litera `l` oznacza, że linia ta ma zostać skrócona tak z lewej, jak i z prawej strony; daje to dużo bardziej elegancki wygląd niż linia rozciągająca się przez całą szerokość kolumny (lub kolumn). Parametr obowiązkowy `2-3` oznacza, że linia ma się rozciągać od drugiej do trzeciej kolumny. Polecenie `\cmidrule` ma tak naprawdę jeszcze jeden parametr opcjonalny, który można umieścić – w nawiasach kwadratowych – przed parametrem w nawiasach okrągłych. Jest to *grubość linii* – ale tej lepiej nie zmieniać.

`\cmidrule`

## Łączenie komórek w pionie

|        |            |        |
|--------|------------|--------|
| foobar | barbaz     | bazqux |
| qwerty | 123<br>456 | asdf   |

```
% table-pion.tex
\documentclass{mwart}
\usepackage{latex-ksiazka-kucharska}
\usepackage{booktabs}
\usepackage{multirow}

\begin{document}
\begin{center}
\begin{tabular}{lcr}
\toprule
foobar & barbaz & bazqux \\
\midrule
\multirow{2}{1.5cm}{qwerty}
& 123 & \multirow{2}{1.5cm}[1ex]{asdf} \\
& 456 & \\
\bottomrule
\end{tabular}
\end{center}
\end{document}
```

Łączenie komórek w pionie (tj. łączenie kilku komórek leżących w jednej kolumnie) jest w  $\text{\LaTeX}$ u nieco trudniejsze. Służy do tego pakiet `multirow`. Definiuje on polecenie `\multirow`, które przyjmuje pięć parametrów, z cze-

`\multirow`

go trzy obowiązkowe. Pierwszy parametr (obowiązkowy) określa liczbę komórek, które łączymy. Drugi parametr (opcjonalny) pominiemy jako potrzebny w bardziej skomplikowanych tabelach. Trzeci parametr (obowiązkowy) określa szerokość tekstu, który umieścimy w łączonych komórkach. Czwarty parametr (opcjonalny) pozwala ręcznie wypozycjonować tekst, który ma się znaleźć w połączonych komórkach; jest to po prostu odległość, o którą tenże tekst będzie przesunięty w górę w stosunku do domyślnego umiejscowienia (oczywiście, podając odległość ujemną, otrzymamy przesunięcie w dół). Odległość tę warto podać w jednostkach  $ex$ , które skalują się razem ze stopniem pisma, co oznacza, że w przypadku zmiany rozmiaru liter, odległość podana w  $ex$  ma szansę pozostać poprawna. Wreszcie, piąty parametr (obowiązkowy) to tekst, który ma się znaleźć w połączonych komórkach.

# Kolory

Kolory są – wbrew pozorom – bardzo skomplikowaną sprawą. W tym rozdziale krótko opowiemy, jak można przydać barw dokumentom  $\text{\LaTeX}$ owym, korzystając z predefiniowanych kolorów, a następnie pokażemy, jak definiować własne kolory.

Zanim poświęcimy kilka stron na omówienie kwestii wielobarwnych dokumentów, najpierw kilka słów ostrzeżenia. Jeżeli naszym celem jest tworzenie dokumentów nieestetycznych i nieczytelnych, używanie wielu kolorów jest jednym z najprostszych sposobów. Jeżeli natomiast zależy nam na eleganckim i czytelnym składzie, warto ograniczyć kolory do minimum. (Liczba kolorów ma też znaczący wpływ na koszt druku.) W zasadzie można przyjąć, że (niemal) jedynymi miejscami, w których użycie kolorów *jest* uprawnione, są rysunki i wykresy oraz prezentacje; w tekście drukowanym rozsądnie jest pozostać przy tradycyjnej czerni i bieli, ewentualnie wzbogaconej dobrze wybranym odcieniem szarości. Jeżeli zdecydujemy się na kolory – czy nawet odcień szarości – należy *koniecznie* sprawdzić, czy postać dokumentu na urządzeniu docelowym jest satysfakcjonująca – kolory potrafią płać figle, zarówno na drukarkach czy maszynach drukarskich, jak i na rzutnikach komputerowych.

## Nadawanie koloru elementom dokumentu

Co z oczu, to z serca.

```
% kolor-wbudowany.tex
\documentclass{mwart}
\usepackage{latex-ksiazka-kucharska}
\usepackage{xcolor}

\begin{document}
\pagecolor{lightgray}
\textcolor{white}{Co z~oczu, to z~\colorbox{gray}{serca}.}
\end{document}
```

Standardowy L<sup>A</sup>T<sub>E</sub>X niestety nie zna się na kolorach; na szczęście już dawno dostrzeżono tę niedogodność i zaradzono jej, najpierw za pomocą pakietu `color`, a następnie za pomocą rozszerzającego jego możliwości pakietu `xcolor`.

Podstawowym poleceniem pakietu `xcolor` jest `\textcolor`, którego pierwszy parametr to określenie (np. nazwa) koloru, a drugi – tekst, który ma być nim złożony. Standardowo dostępne (zdefiniowane przez pakiet `xcolor`) nazwy kolorów to: `red`, `green`, `blue`, `cyan`, `magenta`, `yellow`, `black`, `gray`, `white`, `darkgray`, `lightgray`, `brown`, `lime`, `olive`, `orange`, `pink`, `purple`, `teal`, `violet` (zamiast tłumaczyć nazwy te na język polski, co osobom mniej biegłym w nazywaniu kolorów i tak nie pomogłoby zbyt wiele, polecamy samodzielnie je wypróbować).

Możemy również użyć danego koloru jako tła, korzystając z polecenia `\colorbox`. Również i ono przyjmuje dwa parametry, określenie koloru i tekst, lecz tym razem podany kolor będzie kolorem tła. Uwaga: jak sama na-

- Istnieje jeszcze pakiet `xxcolor`, rozszerzający możliwości pakietu `xcolor`, ale omawianie tego rozszerzenia wykracza poza zakres naszej książeczki.

```
\textcolor
```

```
\colorbox
```

zwa wskazuje, polecenie to umieszcza podany tekst w „pudełku”, co oznacza m.in., że nie może on być złamany między wierszami.

Trzecim z podstawowych poleceń pakietu `xcolor` jest `\pagecolor`, które zmienia kolor tła strony. Poza bardzo szczególnego rodzaju dokumentami, jak np. ulotki reklamowe, nie widać powodu, dla którego jego użycie miałyby sens w dokumentach przeznaczonych do wydruku, a nie do oglądania na ekranie.

`\pagecolor`

## Mieszanie farb

Co z oczu, to z serca.

```
% kolor-mieszanie.tex
\documentclass{mwart}
\usepackage{latex-ksiazka-kucharska}
\usepackage{xcolor}
\definecolor{bookgreen}{RGB}{0,226,181}

\begin{document}
\textcolor{blue!30!black}{Co
 z~\colorbox{bookgreen}{oczu}, to
 z~\colorbox{gray!20}{serca}.}
\end{document}
```

Może się zdarzyć, że proponowany przez pakiet `xcolor` zestaw kolorów okaże się niewystarczający. Możemy sobie z tym problemem poradzić na dwa sposoby. Pierwszy z nich to definiowanie własnego koloru przy użyciu polecenia `\definecolor`. Podajemy mu: nazwę definiowanego koloru, mo-

`\definecolor`

def kolorów oraz liczbowe określenie koloru. Pakiet `xcolor` pozwala korzystać z całkiem pokaźnej ilości modeli kolorów; pewnie najprostszym w użyciu jest tzw. model *addytywny* RGB, w którym kolor jest definiowany przy użyciu trzech składowych – czerwonej (ang. *red*), zielonej (ang. *green*) i niebieskiej (ang. *blue*). W tym modelu każda ze składowych jest liczbą całkowitą z przedziału  $[0, 255]$ ; trzy maksymalne wartości odpowiadają kolorowi białemu, trzy zera – czarnemu, pozostałe kombinacje – różnym proporcjom trzech barw podstawowych.

Bardzo wygodną cechą pakietu `xcolor` (i zarazem jedną z różnic między nim a prostszym pakietem `color`) jest możliwość mieszania kolorów. W podanym przykładzie mamy tekst w kolorze składającym się w 30% z koloru niebieskiego i w 70% z koloru czarnego oraz tło składające się w 20% z koloru szarego i w 80% z koloru białego.

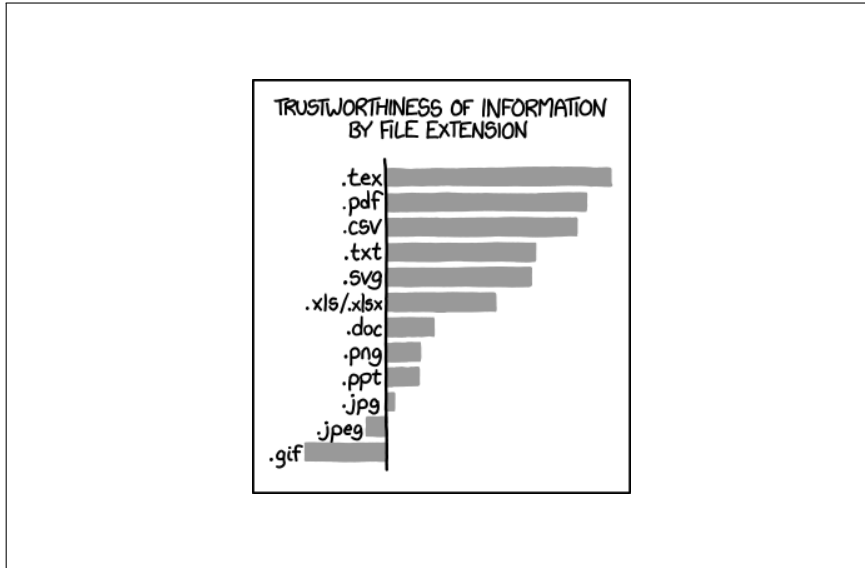


# Grafiki

W wielu typach dokumentów zachodzi potrzeba użycia takich elementów, jak zdjęcia, wykresy, diagramy czy rysunki, słowem – ilustracje. Jak się okazuje, [L<sup>A</sup>T<sub>E</sub>X może nam pomóc w przygotowaniu niektórych typów ilustracji](#), jak diagramy czy wykresy; inne, jak zdjęcia, będą pochodzić z innych źródeł. W tym rozdziale nauczymy się włączać do dokumentu L<sup>A</sup>T<sub>E</sub>Xowego ilustracje zapisane w oddzielnych plikach.

Temat tworzenia grafik wykracza poza ramy tej książki, jednak zainteresowanych Czytelników zachęcamy do zbadania możliwości pakietu `tikz`.

## Dołączanie grafiki



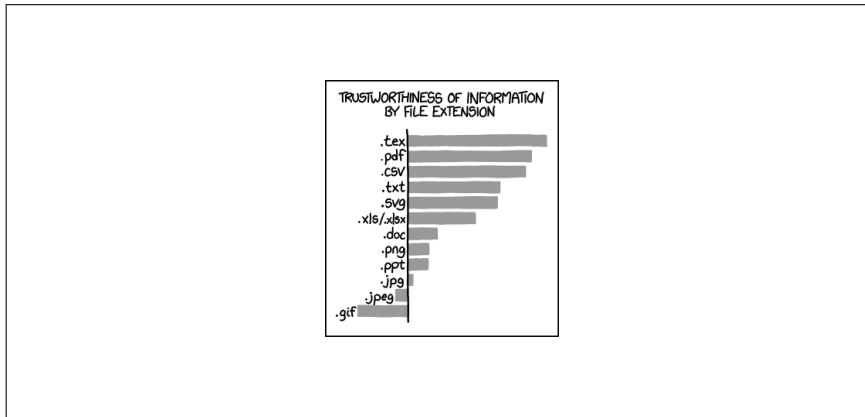
```
% grafiki-dolaczanie.tex
\documentclass{mwart}
\usepackage{latex-ksiazka-kucharska}
\usepackage{graphicx}

\begin{document}
\begin{center}
\includegraphics[width=5cm]{xkcd-1301}
\end{center}
\end{document}
```

Powyższy przykład ilustruje bodaj najprostszy sposób włączania grafiki do tekstu w  $\text{\LaTeX}$ u. Zamiast podawać szerokość (ang. *width*), można podać wy-

sokość (ang. *height*); drugi wymiar zostanie automatycznie dostosowany tak, by zachować proporcje obrazu.

Gdy używamy do kompilacji programu `pdf $\LaTeX$`  lub `x $\LaTeX$` , można włączać grafiki następujących typów: `pdf`, `png`, `jpg` (i kilku innych, rzadziej używanych). Gdyby oprócz pliku `xkcd-1301.png` znajdowały się w bieżącym folderze np. pliki `xkcd-1301.pdf` i/lub `xkcd-1301.jpg`, jako pierwszy byłby użyty plik PDF. Pod jego nieobecność  $\LaTeX$  próbuje użyć pliku `png`, a na końcu `jpg`. Oznacza to w szczególności, że próba dołączenia do dokumentu pliku o nazwie takiej samej, jak nazwa pliku  $\LaTeX$ owego, wymaga podania rozszerzenia – inaczej  $\LaTeX$  będzie próbował dołączyć właśnie tworzony plik `pdf`, z opłakanymi skutkami.



```
% grafiki-dolaczanie.tex
\documentclass{mwart}
\usepackage{latex-ksiazka-kucharska}
\usepackage{graphicx}

\begin{document}
\begin{center}
\includegraphics[width=0.75\textwidth,
```

```

height=8\baselineskip,
keepaspectratio]{xkcd-1301}
\end{center}
\end{document}

```

Dołączając grafikę, można też podać zarówno wysokość, jak i szerokość; oczywiście, jeżeli proporcje podanych wymiarów będą inne niż oryginalne, będzie to skutkowało zniekształceniem obrazu. Możemy tego uniknąć podając dodatkowy parametr `keepaspectratio`. Wówczas podane wysokość i szerokość będą traktowane jako maksymalne i – o ile podane wymiary miały inne proporcje niż oryginalne rozmiary obrazu – jeden z wymiarów w dokumencie będzie mniejszy niż podany.

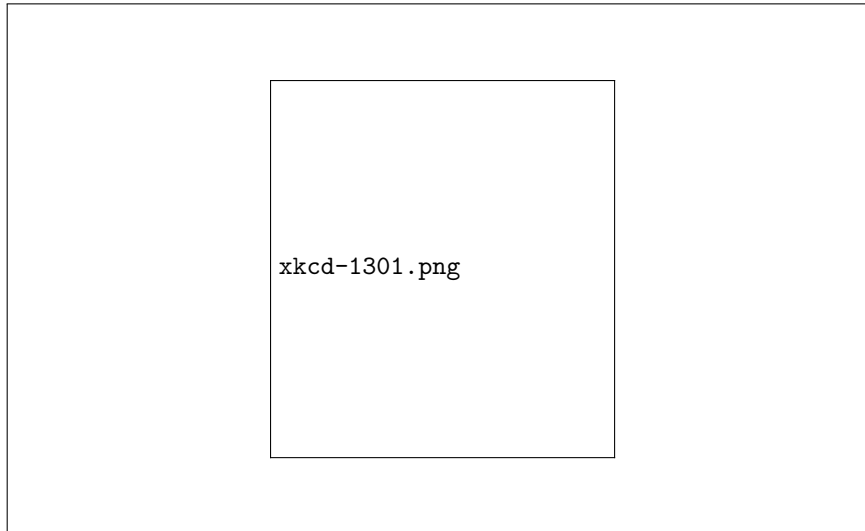
Powyższy przykład ilustruje też ważną technikę: podawanie wymiarów w stosunku do innych wymiarów charakteryzujących dokument. Oprócz szerokości kolumny tekstu (`\textwidth`), można wykorzystać np. wysokość kolumny (`\textheight`), a także np. podać wysokość jako wielokrotność odległości dwóch sąsiednich linii (`\baselineskip`).

```

\textwidth
\textheight
\baselineskip

```

## Wersje testowe dokumentu



```
% grafiki-szkic.tex
\documentclass{mwart}
\usepackage{latex-ksiazka-kucharska}
\usepackage[draft]{graphicx}

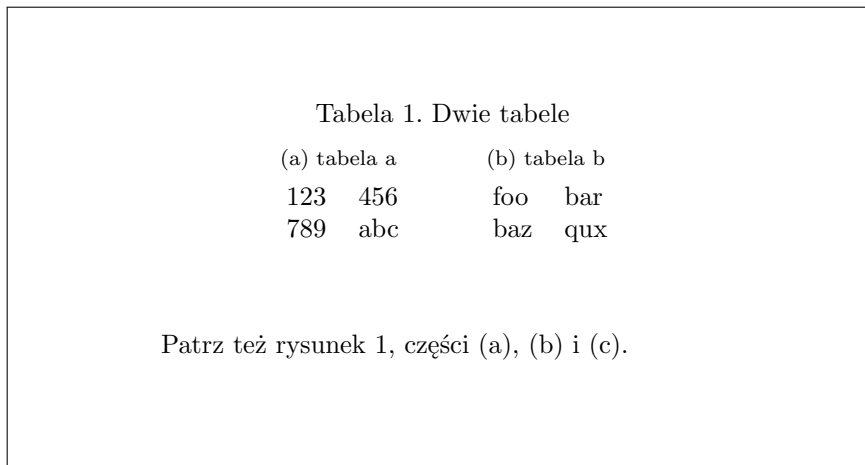
\begin{document}
\begin{center}
\includegraphics[width=5cm,
 height=5cm,
 keepaspectratio]{xkcd-1301}
\end{center}
\end{document}
```

Dołączanie grafik (szczególnie zdjęć w wyższej rozdzielczości) powoduje zarówno wzrost rozmiaru pliku PDF, jak i wydłużenie czasu jego generowania. W przypadku pliku z większą liczbą zdjęć może to być kłopotliwe – szcze-

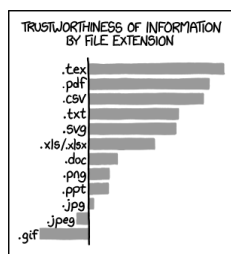
gólnie, jeżeli mamy zwyczaj częstej kompilacji dokumentu. Wówczas przydaje się opcja `draft` pakietu `graphics`. Powoduje ona, że w miejsce grafik pojawiają się jedynie ramki z nazwą pliku.

Można też podać opcję `draft` przy wywołaniu polecenia `\includegraphics` – wówczas dotyczyć będzie ona tej konkretnej grafiki, a nie wszystkich zdjęć w dokumencie. Wygodniej jest jednak użyć opcji pakietu – wówczas wystarczy wprowadzić jedną zmianę w całym pliku, żeby wygenerować docelową wersję ze wszystkimi obrazami.

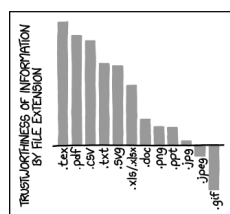
## Grupowanie grafik



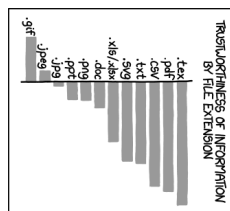
Rysunek 14. Wyciąg z dokumentu wynikowego (strona 1)



(a) Pierwszy rysunek



(b) Drugi rysunek (obrócony)



(c)

Rysunek 1. Trzy rysunki obok siebie

```
% grafiki-grupowanie.tex
\documentclass{mwart}
\usepackage[utf8]{inputenc}
\usepackage{polski}
\usepackage{latex-ksiazka-kucharska}
\usepackage{graphicx}
\usepackage{subfig}
\captionsetup{labelseparator=period}
\captionsetup{justification=centerlast}

\begin{document}
\begin{figure}
 \centering
 \subfloat[Pierwszy rysunek]{%
 \includegraphics[width=3cm]{xkcd-1301}%
 \label{rys:pierwszy}}
 \quad
 \subfloat[Drugi rysunek (obrócony)]{%
 \includegraphics[angle=90,width=3cm]{xkcd-1301}%
 \label{rys:drugi}}
 \\
 \subfloat[]{%
 \includegraphics[angle=-90,width=3cm]{xkcd-1301}%
 \label{rys:trzeci}}
 \caption{Trzy rysunki obok siebie}
 \label{rys:potrojny-rysunek}
\end{figure}

Patrz też rysunek~\ref{rys:potrojny-rysunek},
części~\subref{rys:pierwszy}, \subref{rys:drugi}
i~\subref{rys:trzeci}.

\begin{table}
 \centering
 \caption{Dwie tabele}
```



```

\subfloat[tabela a]{
 \begin{tabular}{ll}
 123 & 456 \\
 789 & abc
 \end{tabular}}
\quad
\subfloat[tabela b]{
 \begin{tabular}{ll}
 foo & bar \\
 baz & qux
 \end{tabular}}
\end{table}
\end{document}

```

Zdarza się, że zachodzi potrzeba połączenia kilku grafik, diagramów czy tabel w jedną „grupę”. W takiej sytuacji z pomocą przychodzi pakiet `subfig`, który pozwala na połączenie kilku np. rysunków w jedną „pływającą” całość. Podstawowym poleceniem jest `\subfloat`, w argumencie którego należy umieścić grafikę. Okazuje się jednak, że sprawa nie jest taka prosta.

Po pierwsze, „grupa” może mieć numer i podpis (oba wstawione przy użyciu polecenia `\caption` – numer automatycznie, podpis oczywiście należy podać w jego argumencie), ale każdy z rysunków w grupie może mieć własny „numer” (zwykle będzie to oznaczenie literowe, jak w podanym przykładzie) i podpis. Podpis ten podaje się w argumencie opcjonalnym polecenia `\subfloat`. Pusty podpis oznacza, że rysunek będzie miał „numer” (literkę), ale nie będzie miał podpisu.

Po drugie, skoro rysunki w „grupie” są automatycznie numerowane, możemy chcieć powoływać się na nie przy użyciu poleceń `\label` oraz `\ref`. W takiej sytuacji polecenie `\label` należy umieścić *wewnątrz argumentu* polecenia `\subfloat`, zaś zamiast polecenia `\ref` użyć polecenia `\subref`.

Po trzecie, aby rysunki nie stykały się bokami, należy wstawić między nie odstęp (na przykład długości dwóch fioletów, [uzyskiwany poleceniem](#) `\quad`). Aby umieścić rysunek w nowym wierszu, stosujemy polecenie `\\`.

Po czwarte, zwróćmy uwagę, że podpis (`\caption`) znajduje się *pod* rysunkiem (lub rysunkami), ale *nad* tabelą (lub tabelami). Zasady tej należy prze-

`\subfloat`

`\subref`

*fiolet* – jednostka długości równa aktualnie używanemu stopniowi pisma.

`\quad`

`\\`

strzeżać, w przeciwnym wypadku może się okazać, że niechcący otworzymy portal, przez który dostaną się do naszego świata okropne potwory.

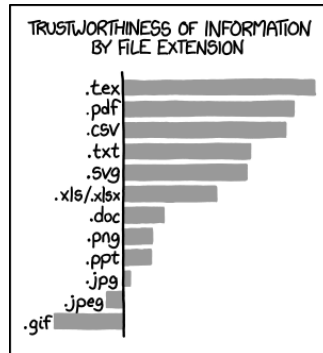
Po piąte, pakiet `subfig` nadpisuje kropkę jako symbol kończący nagłówki podpisów tabel i rysunków, przywracając standardowy,  $\LaTeX$ owy (i angielski) dwukropek. Aby z powrotem nauczyć  $\LaTeX$ a polskich zwyczajów, należy wydać polecenie `\captionsetup{labelseparator=period}`.

Po szóste, pakiet `subfig` ma *mnóstwo* opcji konfiguracyjnych, szczegółowo opisanych w dokumentacji. Ponadto, nie jest jedynym narzędziem pozwalającym łączyć rysunki, tabele itp. w grupy (inne to np. pakiety `subcaption` czy `floatrow`). Jeżeli pakiet `subfig` nie zaspakają naszych potrzeb, warto zerknąć do dokumentacji innych, podobnych pakietów. Tutaj dla przykładu zademonstrowaliśmy opcję `justification=centerlast`. Proponujemy usunąć wiersz ją zawierający i zwrócić uwagę na różnicę w wyglądzie pliku wynikowego.

Po siódme (i ostatnie), zwróćmy uwagę na znaki procenta na końcach niektórych wierszy. Zgodnie z zasadami, o których była mowa w *Elementarzu* (por. s. 16),  $\LaTeX$  zamienia znaki końca wiersza na spacje. Oznacza to, że jeśli chcemy przejść w pliku źródłowym do nowego wiersza nie wprowadzając spacji, musimy jakoś kazać  $\LaTeX$ owi zignorować znak końca wiersza. Standardowym sposobem jest użycie znaku komentarza (czyli procenta), który nakazuje  $\LaTeX$ owi zignorować wszystko do najbliższego znaku końca wiersza, łącznie z tym znakiem.

Bardziej techniczny powód jest podany w dokumentacji pakietu `subfig`.

## Pływające grafiki



Rysunek 1. Przykładowy rysunek

Rysunek 1 na stronie 1 sugeruje, że plik z rozszerzeniem `.tex` nie może nas wprowadzić w błąd.

### Spis rysunków

1. Przykładowy rysunek . . . . . 1

```
% grafiki-plywajace.tex
\documentclass{mwart}
\usepackage[utf8]{inputenc}
\usepackage{polski}
\usepackage{latex-ksiazka-kucharska}
```

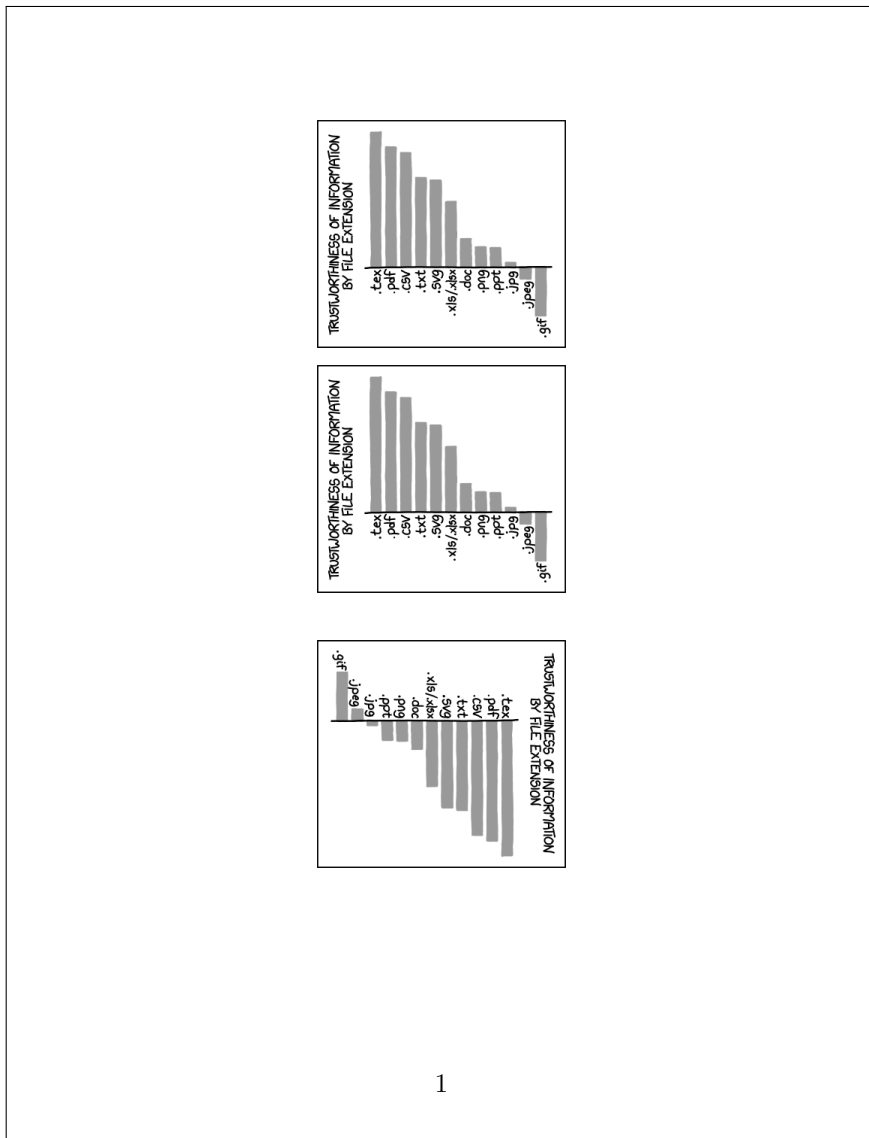
```
\usepackage{graphicx}

\begin{document}
Rysunek \ref{rys:przykladowy-rysunek} na stronie
\pageref{rys:przykladowy-rysunek} sugeruje, że plik
z~rozszerzeniem \texttt{.tex} nie może nas wprowadzić
w~błąd.

\begin{figure}
 \centering
 \includegraphics[width=0.5\textwidth]{xkcd-1301}
 \caption{Przykładowy rysunek}
 \label{rys:przykladowy-rysunek}
\end{figure}

\listoffigures
\end{document}
```

W innej części książki (por. s. 108) pokazaliśmy, jak „oderwać” tabelę od miejsca, w którym została ona zdefiniowana w pliku źródłowym, i pozwolić  $\text{\TeX}$ owi znaleźć dla niej najlepsze (przynajmniej według niego) miejsce – zwykle na górze strony (bieżącej lub następnej). Możemy uczynić to samo z grafiką; służy do tego środowisko `figure`. Wszystko, co powiedzieliśmy o „pływających” tabelach (podpisy, mechanizm powołań itd.) odnosi się również do grafik, poprzestaniemy więc na prostym przykładzie.



Rysunek 16. Wyciąg z dokumentu wynikowego

## Obracanie grafik

Wyciąg z dokumentu wynikowego pokazano na rysunku 16, s. 133.

```
% grafiki-obracanie.tex
\documentclass{mwart}
\usepackage{latex-ksiazka-kucharska}
\usepackage{graphicx}

\begin{document}
\begin{center}
 \includegraphics[width=3cm,angle=90]{xkcd-1301}
\end{center}

\begin{center}
 \includegraphics[angle=90,height=3cm]{xkcd-1301}
\end{center}

\begin{center}
 \includegraphics[angle=-90,totalheight=3cm]{xkcd-1301}
\end{center}
\end{document}
```

Obrócone grafiki można uzyskać, korzystając z opcji `angle`. Trzeba jednak uważać – kryje ona w sobie pewne pułapki.

Po pierwsze, kolejność, w jakiej podajemy wymiary i kąt obrotu ma znaczenie: jeżeli najpierw napiszemy `width=3cm`, a potem `angle=90`, to  $\LaTeX$  przeskaluje obraz do szerokości 3 cm, a później obróci o  $90^\circ$  przeciwie do ruchu wskazówek zegara. W ten sposób otrzymamy obraz o wysokości 3 cm. Jeżeli z kolei napiszemy najpierw `angle=90`, a następnie `height=3cm`,  $\LaTeX$  najpierw obróci obraz o  $90^\circ$ , a następnie przeskaluje go do wysokości 3 cm; efekt będzie oczywiście identyczny jak poprzednio.

Po drugie, jeżeli chcemy obrócić grafikę o  $90^\circ$ , ale zgodnie z ruchem wskazówek zegara, i jeżeli chcemy podać wysokość grafiki po obróceniu, nie możemy użyć opcji `height`, tylko `totalheight`. Przyczyna tego problemu jest

następująca: obrót o  $90^\circ$  zgodnie z ruchem wskazówek zegara wokół domyślnego środka obrotu, czyli lewego dolnego narożnika grafiki, daje w efekcie zerową wysokość *ponad* linię podstawową, czyli poziomą linię przechodzącą przez ów narożnik, i *głębokość* równą szerokości grafiki przed obrotem. Parametr *totalheight* pozwala zaś podać sumę wysokości i głębokości obrazu.

Sposób obracania grafik ma istotne znaczenie z punktu widzenia składu tekstu. Dobry zwyczaj mówi, że obiekty w dokumencie należy obracać tylko wtedy, gdy jest to konieczne, np. jeśli zależy nam na jak najdokładniejszym przedstawieniu fotografii w proporcjach 4:3 lub 16:9. Po podjęciu decyzji, że spełniamy te wymagania, musimy pamiętać, żeby takie obiekty obracać *przeciwnie* do ruchu wskazówek zegara, niezależnie od tego, czy dokument jest jedno- czy dwustronny.

# Układ strony

## Wymuszenie przejścia do nowej strony

Domyślnie  $\LaTeX$  stara się wypełnić każdą stronę. Czasem jednak zechodzi potrzeba kontynuowania tekstu na nowej stronie, niezależnie od tego, czy poprzednia zdążyła się zapełnić. Standardowo jest tak np. w przypadku rozpoczęcia rozdziału w książce, choć tym przypadkiem zajmuje się oczywiście polecenie `\chapter`. Jest kilka poleceń, które mogą nam pomóc w pozostałych przypadkach.

Podstawowym poleceniem służącym do zmiany strony jest `\pagebreak`. Podane bez żadnego argumentu wymusza zmianę strony w miejscu, w którym występuje. [Przyjmuje ono jeden parametr opcjonalny – liczbę całkowitą od 0 do 4](#) – im wyższa wartość, tym bardziej  $\LaTeX$  będzie się starał złamać stronę w podanym miejscu (`\pagebreak` bez argumentu jest równoważne `\pagebreak[4]`). Wariant z parametrem równym np. 3 ma istotne zastosowanie: jeżeli z jakichś powodów uznamy, że warto zmienić stronę w jakimś miejscu (położonym blisko dolnego marginesu), dobrze jest zrobić to poleceniem `\pagebreak[3]`. W ten sposób, gdy przeredagujemy dokument tak, że miejsce to wypadnie np. w połowie strony, polecenie to zostanie zignorowane. Dzieje się tak dlatego, że  $\LaTeX$  musiałby zbyt mocno zwiększyć odstępy pionowe, żeby wyrównać niepełną stronę do górnego i dolnego marginesu.

Jak widać z powyższego akapitu, polecenie `\pagebreak` zachowuje pionowe wyrównanie materiału (do górnego i dolnego marginesu). Nieraz chcemy jednak pozostawić dół strony pusty (tak jest np. w przypadku rozpoczęcia nowego rozdziału). Wówczas można zastosować polecenie `\newpage`, które

`\pagebreak`

Nie wnikając w szczegóły techniczne, powiedzmy tyle, że wydanie tego polecenia *wewnątrz* akapitu może dać nieoczekiwane rezultaty. Nie jest to błąd, choć być może bezpieczniejszym jest podawać je jedynie na początku nowego akapitu, wyliczenia itp. struktur, czyli np. po pustej linii w pliku źródłowym.

`\newpage`



wymusza złamanie strony w miejscu, w którym została wydana, pozostawiając przestrzeń od tego miejsca do końca strony pustą.

Dla pełności wspomnijmy jeszcze o komendzie `\clearpage`, która robi to, co `\newpage`, ale dodatkowo powoduje, że zaraz na następnej stronie pojawią się wszystkie „pływające” tabele i grafiki (patrz paragrafy na stronach 108 i 131), dla których dotąd  $\text{\LaTeX}$  nie znalazł miejsca. ↑ W trybie dwustronnym, o którym mowa w innym paragrafie tego rozdziału, podobną rolę pełni polecenie `\cleardoublepage`, które w razie potrzeby wstawia pustą stronę, aby dalszy ciąg tekstu zaczął się na prawej (nieparzystej) stronie.

`\clearpage`

Polecenie to jest np. używane przez polecenia rozpoczynające rozdziały czy części.

`\cleardoublepage`

## Zakaz przejścia do nowej strony

daes, ent ipsum pnaetra ngua, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet var-

2

ius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus.

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero.

```
% strona-przejscie.tex
\documentclass{mwbk}
```

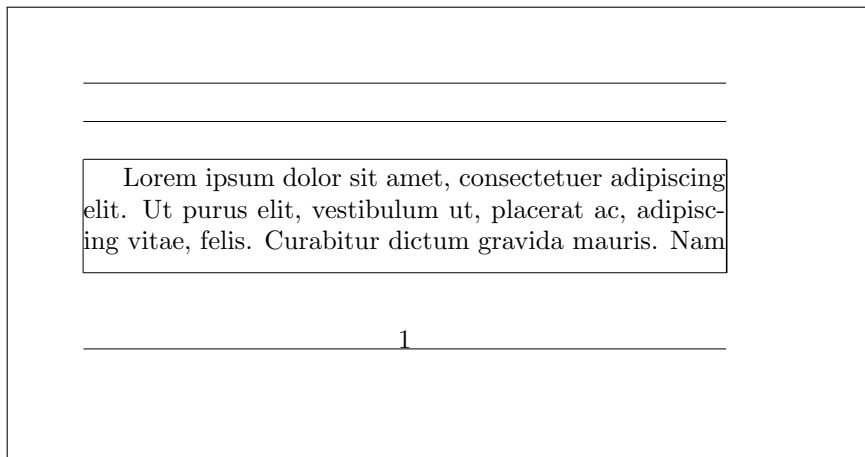
```
\usepackage{latex-ksiazka-kucharska}
\usepackage{lipsum}

\begin{document}
\lipsum[8-9]
\nopagebreak
\lipsum[10-12]
\end{document}
```

Może się zdarzyć również sytuacja, w której zależy nam, by w pewnym miejscu *nie* została złamana strona. Domyślnie  $\LaTeX$  unika łamania strony np. po pierwszym lub przed ostatnim wierszem akapitu, a także w kilku innych, podobnego typu sytuacjach. Gdyby jednak zachodziła potrzeba wyraźnego zabronienia złamania strony w pewnym miejscu, można użyć polecenia `\nopagebreak`. Warto dokładnie porównać powyższy przykład z dokumentem, który uzyskuje się po usunięciu polecenia `\nopagebreak`.

`\nopagebreak`

## Zmiana rozmiaru strony i marginesów



```
% strona-marginsy.tex
\documentclass{mwart}
\usepackage[paperwidth=115mm,paperheight=60mm,
 lmargin=1cm,rmargin=2cm,
 tmargin=2cm,bmargin=2.5cm,
 headsep=0.5cm,headheight=0.5cm,
 footskip=1cm,
 showframe]{geometry}
\usepackage{lipsum}

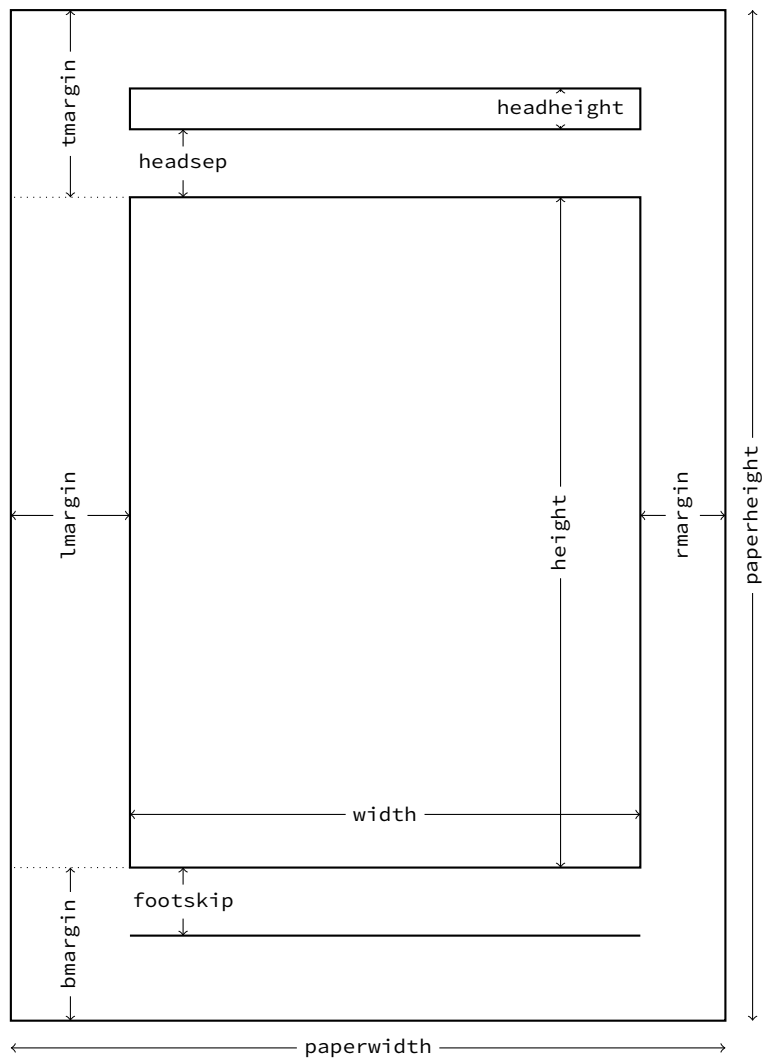
\begin{document}
\lipsum[1]
\end{document}
```

Bardzo istotną kwestią z punktu widzenia projektu typograficznego jest układ strony. Specjaliści sugerują, że szerokość kolumny tekstu powinna być równa ok. 66 znaków. W wielu przypadkach takie ustawienia (w przypadku klas `mwart` i `mwbk` są to strona formatu A4 i spore marginesy) nie muszą

nam odpowiadać. Aby to zmienić (co w zasadzie odradzamy), można postąpić na dwa sposoby. Prościej jest podanie opcji `[wide]` klasie `mwart` czy `mwbk`. Wówczas lewy i prawy margines będą wynosiły zaledwie 2 cm. Nieco bardziej skomplikowanym, ale też pozwalającym na dużo większą dowolność, jest użycie pakietu `geometry`.

Jak widać w powyższym przykładzie, w opcjach wywołania pakietu `geometry` można zawrzeć całkiem sporo informacji o tym, jakie mają być wymiary strony, kolumny tekstu i innych elementów składu. Parametrów tych jest jednak o wiele więcej niż to wynika z przykładu. W wielu przypadkach można też uprościć sobie życie, pisząc np. `hmargin=2cm`, gdy lewy i prawy margines są takie same. Po szczegóły odsyłamy do dokumentacji pakietu. Nas interesuje w tej chwili kwestia ustawienia rozmiarów strony i kolumny tekstu oraz nagłówka (w którym znajduje się często np. tytuł bieżącego rozdziału) i stopki (w której znajduje się często np. pagina, czyli numer strony). Do tego wystarczy zestaw opcji przedstawionych na rysunku 17.

Format A4 oznacza wymiary strony 210×297 mm, co oznacza wymiary ok. czterokrotnie mniejsze od formatu A0 – 841×1189 mm. Czytelnikom ciekawym przyczyny takich właśnie wartości polecamy wyznaczyć stosunek boków prostokąta, którego połowa jest podobna do całości, i znalezienie wymiarów prostokąta o takim właśnie stosunku boków i polu powierzchni równym 1 m<sup>2</sup>.



Rysunek 17. Opcje pakietu geometry

Oprócz opcji przedstawionych na diagramie, warto wspomnieć o jeszcze kilku. Jedną z nich jest opcja `showframe`, która powoduje, że  $\LaTeX$  rysuje ramki ilustrujące obowiązujące w danym dokumencie wymiary. Wysokość kolumny można podać w wierszach tekstu opcją `lines=\langle liczba całkowita \rangle`. Można też, po podaniu wysokości kolumny w „normalny” sposób (tj. podając jeden lub dwa z parametrów: `tmargin`, `margin`, `height`), użyć opcji `heightrounded`, która *zaokrągla* podaną wysokość do wielokrotności wysokości wiersza (tym samym powodując, że na każdej stronie zmieści się całkowita liczba wierszy).

Warto również wiedzieć, że nie trzeba podawać *wszystkich* wymiarów; pakiet `geometry` potrafi dość inteligentnie uzupełnić sobie „brakujące dane”. Przykładowo, gdy dane są wymiary papieru i oba marginesy poziome (lewy i prawy), nie trzeba (a nawet nie powinno się [tego robić](#)) podawać dodatkowo szerokości kolumny tekstu, która jest równa różnicy szerokości papieru i łącznej szerokości marginesów. Okazuje się jednak, że można np. podać jedynie szerokość papieru i szerokość kolumny; wówczas marginesy będą miały równą szerokość (jeżeli używamy opcji `oneside`, czyli stosujemy druk jednostronny) lub margines zewnętrzny będzie o 50% szerszy niż wewnętrzny (jeżeli stosujemy druk dwustronny). Podobne zasady obowiązują, gdy „zapomnimy” podać innych zestawów parametrów; szczególne są opisane w dokumentacji pakietu `geometry`, w paragrafie *Auto-completion*.

Gdy podane wymiary spowodują powstanie sprzecznego układu warunków, pakiet `geometry` po cichu zignoruje niektóre z parametrów.

1

## 1. Lorem ipsum

### 1.1. Dolor sit amet

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consetetur.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

2

*1. Lorem ipsum*

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetur at, consectetur sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetur a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollici-

*1.1. Dolor sit amet*

3

tudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consetetuer. Nullam elementum, urna vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus.

Rysunek 20. Wyciąg z dokumentu wynikowego (strona 3)



## Standardowe style stopkek i nagłówek

Wyciąg z dokumentu wynikowego pokazano na *rysunkach 18–20*, s. 142–144.

```
% strona-stopki.tex
\documentclass[twoside]{mwart}
\usepackage{lipsum}
\usepackage{latex-ksiazka-kucharska}
\pagestyle{headings}

\begin{document}
\section{Lorem ipsum}
\subsection{Dolor sit amet}
\lipsum[5-9]
\end{document}
```

W większości książek i czasopism strony są numerowane. Ponadto (najczęściej w nagłówku, czyli nad kolumną tekstu) umieszcza się często tzw. „żywą paginę”, czyli tytuł bieżącego rozdziału (na stronie parzystej, czyli lewej) i paragrafu (na stronie nieparzystej, czyli prawej).

Sytuację utrudnia fakt, że na stronach otwierających rozdział nie powinno się stosować paginy górnej, zaś na stronach kończących rozdział – paginy dolnej. Ponieważ reguły te są odmienne od zasad panujących w typografii anglosaskiej,  $\LaTeX$  ich nie przestrzega. Z pomocą przychodzą klasy  $m\omega^*$ , które – sprytnie modyfikując  $\LaTeX$ owy mechanizm tzw. „stylu stron” – pozwalają automatycznie przestrzegać powyższych zasad.

Polecenie `\pagestyle{<nazwa stylu>}` pozwala wybrać jeden z kilku dostępnych w klasach  $m\omega^*$ , gotowych „stylu stron”. W stylu `plain` (domyślnym w klasach `mwart` i `mwrap`) numery stron umieszczone są w stopkach na środku, zaś nagłówki są puste. W stylu `outer` nagłówki również są puste, zaś numery stron umieszczone są po zewnętrznej stronie stopki. W stylu `headings` stopki są puste, zaś w nagłówkach znajduje się numer strony (po zewnętrznej stronie) i tzw. *żywa pagina*, czyli tytuł (i numer) bieżącego rozdziału, paragrafu lub podparagrafu (w klasie `mwbk` – tytuł rozdziału na

`\pagestyle`

lewych, a paragrafu na prawych stronach; w klasie `mwrap` – tytuł rozdziału po obu stronach, zaś w klasie `mwart` – tytuł paragrafu na lewych, a podparagrafu na prawych stronach). Styl `uheadings` różni się od poprzedniego wyłącznie tym, że pod nagłówkiem znajduje się pozioma linia. W stylu `empty` – jak nietrudno się domyślić – zarówno nagłówki, jak i stopki są całkiem puste. Wreszcie style `myheadings` i `myuheadings` są podobne do `headings` i `uheadings`, ale pozwalają na samodzielne określenie tekstu umieszczanego w nagłówku na lewych i prawych stronach.

## Zmiana układu stopek i nagłówków na pojedynczej stronie

Polecenie `\pagestyle` zmienia styl strony dla wszystkich stron dokumentu (poczynając od tej, na której zostało wydane). Czasem jednak zachodzi potrzeba zmiany stylu tylko na jednej stronie. Przykładowo, pisząc list możemy nie chcieć, żeby na pierwszej jego stronie wystąpił jej numer, natomiast począwszy od drugiej, strony powinny być numerowane. Można oczywiście wydać polecenie `\pagestyle` dwukrotnie – na wybranej i następnej stronie – lecz nie jest to zbyt wygodne (ani eleganckie). Ponadto, może to prowadzić do subtelnych błędów: gdy między dwoma wystąpieniami polecenia `\pagestyle` dopiszemy więcej tekstu (lub skasujemy jego fragment), może się okazać, że odstęp między nimi wyniesie dwie strony – lub że wystąpią na tej samej stronie.

W takich sytuacjach na ratunek przychodzi polecenie `\thispagestyle`, które zmienia styl wyłącznie na stronie, na której zostało wydane.

`\thispagestyle`

Czasem nie jest oczywiste, co to znaczy „na stronie, na której zostało wydane”, przynajmniej dopóki nie sprawdzi się w wynikowym pliku PDF, na której stronie wypadły odpowiednie fragmenty.

*Lorem ipsum – Dolor sit amet*

1

## **1. Lorem ipsum**

### **1.1. Dolor sit amet**

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio

2

*Nazwisko autora*

elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetur at, consectetur sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetur a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetur. Nullam elementum, urna vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus.

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagit-

## Ręczna zmiana tekstu stopek i nagłówków

Wyciąg z dokumentu wynikowego pokazano na rysunkach 21–22, s. 147–148.

```
% strona-reczna-zmiana.tex
\documentclass[twoside]{mwart}
\usepackage{polski}
\usepackage{latex-ksiazka-kucharska}
\usepackage{lipsum}

\begin{document}
\pagestyle{myheadings}
\markboth{Nazwisko autora}
 {Lorem ipsum\ppauza Dolor sit amet}
\section{Lorem ipsum}
\subsection{Dolor sit amet}

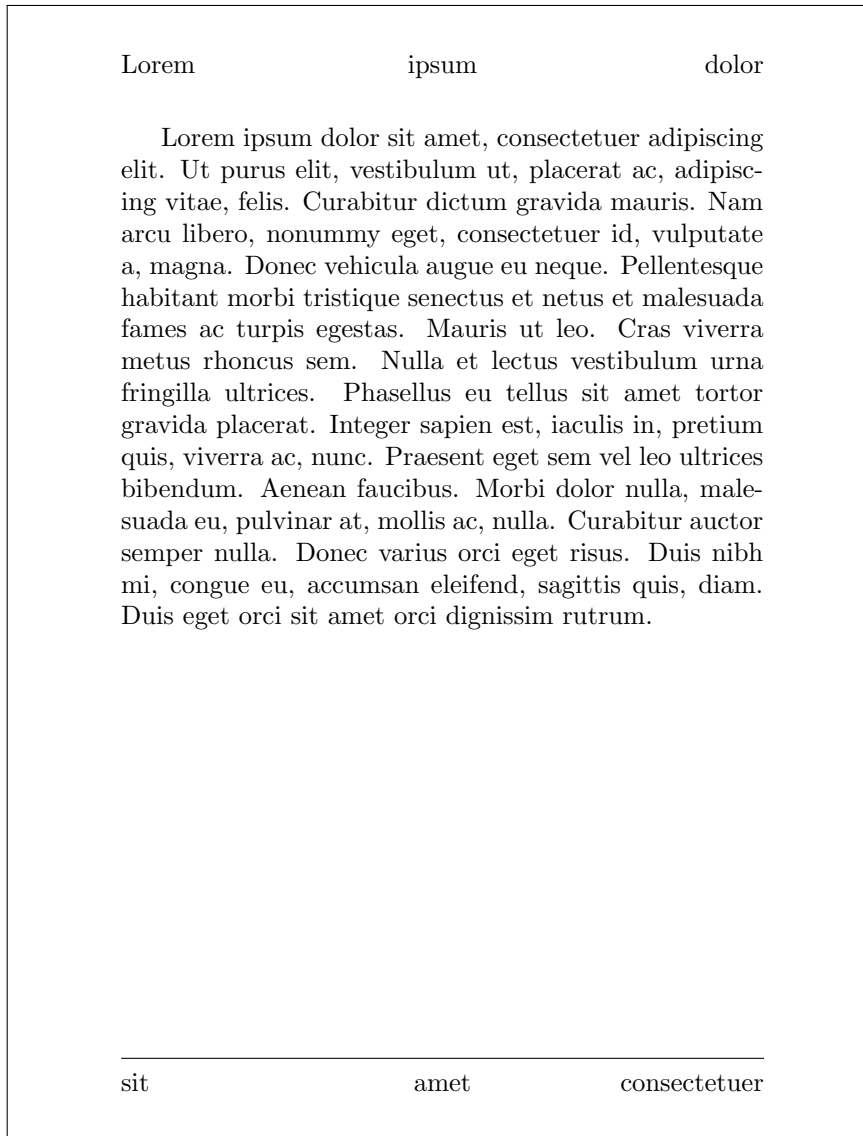
\lipsum[6-10]
\end{document}
```

Może się zdarzyć, że zdefiniowane w klasach `mw*` style stron, odpowiedzialne za zawartość stopek i nagłówków, okażą się niewystarczające. W takiej sytuacji mamy dwie możliwości. Jeżeli potrzebujemy prostej modyfikacji stylu `headings` (lub `uheadings`), w którym np. zamiast numeru i tytułu paragrafu/podparagrafu (w przypadku klasy `mwart`) będziemy chcieli umieścić np. nazwisko autora i tytuł artykułu (lub cokolwiek innego), możemy użyć stylu `myheadings` (lub `myuheadings`) i polecenia `\markboth` do zadania tekstów, które mają się znaleźć w nagłówku. Przykład zastosowania tej metody widoczny jest powyżej.

Warto dodać, że istnieje również polecenie `\markright`, które przyjmuje jeden parametr obowiązkowy, i zmienia tekst w nagłówku wyłącznie na stronach prawych (nieparzystych).

`\markboth`

`\markright`



Rysunek 23. Wyciąg z dokumentu wynikowego

## Nagłówki niestandardowe

Wyciąg z dokumentu wynikowego pokazano na *rysunku 23*, s. 150.

```
% strona-niestandardowe.tex
\documentclass{mwart}
\usepackage{latex-ksiazka-kucharska}
\usepackage{fancyhdr}
\lhead{Lorem}
\chead{ipsum}
\rhead{dolor}
\lfoot{sit}
\cfoot{amet}
\rfoot{consectetuer}
\renewcommand{\headrulewidth}{0pt}
\renewcommand{\footrulewidth}{0.4pt}
\pagestyle{fancy}
\usepackage{lipsum}

\begin{document}
\lipsum[1]
\end{document}
```

W przypadku, gdy nie wystarczą nam standardowe możliwości zmiany nagłówków, udostępniane domyślnie, możemy użyć pakietu `fancyhdr`. Definiuje on nowy „styl strony” o nazwie `fancy`, w którym możemy dowolnie zmienić zarówno nagłówek, jak i stopkę, a także zdecydować, czy chcemy, by były one oddzielone od tekstu głównego poziomą linią.

Najprostszy przykład użycia pakietu `fancyhdr` wygląda jak powyżej. Jak widać, zdefiniowaliśmy lewą, środkową i prawą część nagłówka i stopki oraz podaliśmy grubości linii oddzielających. Domyślnie linia pomiędzy nagłówkiem a kolumną ma szerokość 0,4 pt, zaś linia między stopką a kolumną ma zerową szerokość, czyli jest niewidoczna.

## Ciekawy dokument

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat,



Jan Nowak

congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

2

Rysunek 25. Wyciąg z dokumentu wynikowego (strona 2)

## Różne nagłówki i stopki dla stron parzystych i nieparzystych

Wyciąg z dokumentu wynikowego pokazano na *rysunkach 24–25*, s. 152–153.

```
% strona-rozne.tex
\documentclass[twoside]{mwart}
\usepackage{latex-ksiazka-kucharska}
\usepackage{lipsum}
\usepackage{fancyhdr}
\fancyhf{}
\fancyhead[CE]{Jan Nowak}
\fancyhead[CO]{Ciekawy dokument}
\fancyfoot[LE,RO]{\thepage}
\renewcommand{\headrulewidth}{0pt}
\pagestyle{fancy}

\begin{document}
\lipsum[1-3]
\end{document}
```

Gdy tworzymy dokument dwustronny, często chcielibyśmy, aby zawartość nagłówków i stopek na stronach lewych i prawych różniła się od siebie. Uzyskanie tego efektu przy użyciu pakietu `fancyhdr` jest łatwe.

W przykładzie podaliśmy opcję klasy `twoside`, żeby włączyć tryb druku dwustronnego. Po drugie, zaraz po załadowaniu pakietu `fancyhdr`, poleceniem `\fancyhf` wyłączyliśmy wszystkie domyślne treści w nagłówkach i stopkach (w praktyce oznacza to wyłączenie numeru strony na środku stopki). Następnie ustawiliśmy treść w środku nagłówku (C jak ang. *center* – środek) na stronach parzystych (E jak ang. *even* – parzysty) na „Jan Nowak” oraz treść w środku nagłówku na stronach nieparzystych na „Ciekawy dokument”. W kolejnej linii ustawiliśmy numer strony (polecenie `\thepage`) w lewej stopce na stronach parzystych i jednocześnie prawej stopce na stronach nieparzystych. Wreszcie wyłączyliśmy poziomą linię oddzielającą nagłówki od kolumny tekstu i włączyliśmy pracowicie utworzony styl strony poleceniem `\pagestyle{fancy}`.

`\fancyhf`

`\thepage`

## Rozdział 1

# Rozdział pierwszy

### 1.1. Podrozdział pierwszy

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu,

## Rozdział pierwszy

accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum

## Podrozdział pierwszy

dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commo-

## Rozdział pierwszy

do pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

## Rozdział 2

# Rozdział drugi

### 2.1. Podrozdział drugi

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetur at, consectetur sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

## O nagłówkach w książkach

Wyciąg z dokumentu wynikowego pokazano na *rysunkach 26–30*, s. 155–159.

```
% strona-ksiazki.tex
\documentclass[twoside]{mwbk}
\usepackage[utf8]{inputenc}
\usepackage{polski}
\usepackage{latex-ksiazka-kucharska}
\usepackage{lipsum}

\usepackage{fancyhdr}
\renewcommand{\headrulewidth}{0pt}
\renewcommand{\footrulewidth}{0pt}
\fancypagestyle{plain}{
 \fancyhf{}
 \fancyhead[CE]{\leftmark}
 \fancyhead[CO]{\rightmark}
 \fancyfoot[LE,R0]{\thepage}
}
\pagestyle{plain}
\fancypagestyle{closing}{
 \fancyhf{}
 \fancyhead[CE]{\leftmark}
 \fancyhead[CO]{\rightmark}
}
\fancypagestyle{opening}{
 \fancyhf{}
 \fancyfoot[LE,R0]{\thepage}
}

\begin{document}
\chapter{Rozdział pierwszy}
\markboth{Rozdział pierwszy}{}
\section{Podrozdział pierwszy}
```



```
\markright{Podrozdział pierwszy}
\lipsum[1-7]

\chapter{Rozdział drugi}
\section{Podrozdział drugi}
\lipsum[8]
\end{document}
```

Jeżeli używamy klasy `mwbk`, stopki i nagłówki na stronach rozpoczynających i kończących rozdziały wymagają szczególnego traktowania. Zgodnie z zasadami polskiej typografii, nagłówki na stronie otwierającej rozdział powinny być pusty; to samo dotyczy stopki na stronie kończącej rozdział. Klasa `mwbk` realizuje te zalecenia poprzez specjalne style strony: `open ing`, włączany na pierwszej stronie rozdziału, i `close ing`, włączany na stronie ostatniej. Domyślnie pierwszy z nich ma pusty nagłówek, a drugi – pustą stopkę i te ich cechy oczywiście warto (a nawet należy) zachować.

Zwróćmy uwagę na miejsce, w którym wywołano w preambule polecenie `\pagestyle{plain}`. Z uwagi na szczegóły techniczne dotyczące implementacji mechanizmu stylów w klasach `mw*`, umieszczenie go w innym miejscu może spowodować błędny wygląd dokumentu.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisi. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

## Poziome strony

Wyciąg z dokumentu wynikowego pokazano na *rysunkach 31–33*, s. 162–164.

```
% strona-ksiazki.tex
\documentclass[twoside]{mwbk}
\usepackage[utf8]{inputenc}
\usepackage{polski}
\usepackage{latex-ksiazka-kucharska}
\usepackage{lipsum}

\usepackage{fancyhdr}
\renewcommand{\headrulewidth}{0pt}
\renewcommand{\footrulewidth}{0pt}
\fancypagestyle{plain}{
 \fancyhf{}
 \fancyhead[CE]{\leftmark}
 \fancyhead[CO]{\rightmark}
 \fancyfoot[LE,R0]{\thepage}
}
\pagestyle{plain}
\fancypagestyle{closing}{
 \fancyhf{}
 \fancyhead[CE]{\leftmark}
 \fancyhead[CO]{\rightmark}
}
\fancypagestyle{opening}{
 \fancyhf{}
 \fancyfoot[LE,R0]{\thepage}
}

\begin{document}
\chapter{Rozdział pierwszy}
\markboth{Rozdział pierwszy}{}
\section{Podrozdział pierwszy}
```

```
\markright{Podrozdział pierwszy}
\lipsum[1-7]

\chapter{Rozdział drugi}
\section{Podrozdział drugi}
\lipsum[8]
\end{document}
```

Czasami zachodzi potrzeba obrócenia *pojedynczej strony*. Jest tak na przykład wówczas, gdy chcemy umieścić w książce niezbyt długą, ale za to szeroką tabelę. Możemy wówczas użyć pakietu `pdf\landscape`. Definiuje on środowisko `landscape`, które umieszcza swoją zawartość na osobnych stronach ułożonych poziomo.

## Otoczenia numerowane

W wielu pracach, szczególnie o charakterze matematycznym, zachodzi potrzeba zdefiniowania pewnych otoczeń, które w ramach tej książki będziemy nazywać *otoczeniami numerowanymi* lub po prostu *twierdzeniami*, chociaż mogą to być zarówno twierdzenia, jak i – przykładowo – definicje, lematy, przykłady, uwagi itd. Zbioru narzędzi przydatnych do zarządzania takimi środowiskami dostarcza pakiet `ams thm`.

Zainteresowanych Czytelników zachęcamy do zapoznania się również z innymi pakietami o podobnych zastosowaniach, np. `ntheorem`.

## Tworzenie otoczeń numerowanych

### 1. Pierwsza sekcja

**Twierdzenie 1.** *To jest pierwsze twierdzenie.*

**Przykład.** *To jest pierwsza definicja.*

**Definicja 1.** *To jest pierwsza definicja.*

### 2. Druga sekcja

**Twierdzenie 2.** *To jest drugie twierdzenie.*

```
% otoczenia-numerowane.tex
\documentclass{mwart}
\usepackage[utf8]{inputenc}
\usepackage{polski}
\usepackage{latex-ksiazka-kucharska}

\usepackage{amsthm}
\newtheorem{tw}{Twierdzenie}
\newtheorem{defi}{Definicja}
\newtheorem*{przy}{Przykład}

\begin{document}
\section{Pierwszy paragraf}

\begin{tw}
To jest pierwsze twierdzenie.
```



```
\end{tw}

\begin{przy}
To jest pierwsza definicja.
\end{przy}

\begin{defi}
To jest pierwsza definicja.
\end{defi}

\section{Drugi paragraf}
\begin{tw}
To jest drugie twierdzenie.
\end{tw}
\end{document}
```

Do tworzenia otoczeń numerowanych służy polecenie `\newtheorem`, będące częścią pakietu `amsthm`. Umieszcza się je w preambule i przyjmuje ono dwa argumenty obowiązkowe – pierwszy z nich określa nazwę referencyjną, której będziemy używać, aby odwołać się do otoczenia, drugi zaś określa nazwę środowiska wypisywaną w tekście dokumentu.

Domyślnie otoczenia o różnych nazwach referencyjnych są numerowane w sposób niezależny od siebie i ciągły. Zdefiniowanie otoczenia przy użyciu polecenia `\newtheorem*` powoduje, że nie towarzyszy mu numer.

`\newtheorem``\newtheorem*`

## Numeracja zależna

### 1. Pierwszy paragraf

**Twierdzenie 1.** *To jest pierwsze twierdzenie.*

**Definicja 2.** *To jest pierwsza definicja.*

### 2. Drugi paragraf

**Twierdzenie 3.** *To jest drugie twierdzenie.*

**Przykład 1.** *To jest pierwszy przykład.*

```
% otoczenia-zalezna.tex
\documentclass{mwart}

\usepackage[utf8]{inputenc}
\usepackage{polski}
\usepackage{latex-ksiazka-kucharska}

\usepackage{amsthm}

\newcounter{licznik}

\newtheorem{tw}[licznik]{Twierdzenie}
\newtheorem{defi}[licznik]{Definicja}
\newtheorem{prz}{Przykład}

\begin{document}
```

```
\section{Pierwszy paragraf}

\begin{tw}
To jest pierwsze twierdzenie.
\end{tw}

\begin{defi}
To jest pierwsza definicja.
\end{defi}

\section{Drugi paragraf}

\begin{tw}
To jest drugie twierdzenie.
\end{tw}

\begin{prz}
To jest pierwszy przykład.
\end{prz}
\end{document}
```

Niekiedy chcemy, aby pewne otoczenia, choć na pozór niezależne, korzystały z tego samego licznika. W praktyce, często wiąże się w ten sposób umieszczane w dokumentach matematycznych lematy i twierdzenia. W tej książce zdefiniowaliśmy (wykorzystywane w pierwszej części) otoczenia *Pytanie* i *Ćwiczenie*, które korzystają z tej samej własności.

Aby zastosować numerację zależną w praktyce, należy, przy użyciu polecenia `\newcounter`, zdefiniować licznik – polecenie przyjmuje jeden argument obowiązkowy, będący jego nazwą. Nazwę tę umieszcza się następnie jako argument opcjonalny (pomiędzy nazwą referencyjną a nazwą drukowaną) przy każdym otoczeniu, które ma z niego korzystać. Każdorazowe wywołanie otoczenia korzystającego z danego licznika powoduje zwiększenie jego wartości o jeden, co pozwala uzyskać oczekiwany efekt.

`\newcounter`

## Numeracja hierarchiczna

### 1. Pierwszy paragraf

**Twierdzenie 1.1.** *To jest pierwsze twierdzenie.*

**Definicja 1.** *To jest pierwsza definicja.*

### 2. Drugi paragraf

**Twierdzenie 2.1.** *To jest drugie twierdzenie.*

**Definicja 2.** *To jest druga definicja.*

```
% otoczenia-hierarchiczna.tex
\documentclass{mwart}

\usepackage[utf8]{inputenc}
\usepackage{polski}
\usepackage{latex-ksiazka-kucharska}

\usepackage{amsthm}

\newtheorem{tw}{Twierdzenie}[section]
\newtheorem{defi}{Definicja}

\begin{document}
\section{Pierwszy paragraf}

\begin{tw}
```

```
To jest pierwsze twierdzenie.
\end{tw}

\begin{defi}
To jest pierwsza definicja.
\end{defi}

\section{Drugi paragraf}

\begin{tw}
To jest drugie twierdzenie.
\end{tw}

\begin{defi}
To jest druga definicja.
\end{defi}
\end{document}
```

Wielokrotnie zachodzi potrzeba modyfikacji sposobu numeracji twierdzeń tak, aby wewnątrz danego poziomu hierarchii dokumentu (rozdziału, paragrafu, podparagrafu itd.) zaczynała się ona od nowa. Taki efekt można osiągnąć, wzbogacając wywołanie polecenia `\newtheorem` o argument opcjonalny, umieszczany w nawiasach kwadratowych za argumentami obowiązkowymi, zawierający nazwę odpowiedniego poziomu (`chapter`, `section`, `subsection` itd.). Oczywiście, aby nie doprowadzić do sytuacji, w której w dokumencie będą występować dwa twierdzenia o tym samym numerze, poprzedza się je numerem bieżącego poziomu. W przykładzie, numer przy *Twierdzeniu 2.1* oznacza, że jest to pierwsze twierdzenie w paragrafie drugim.

## Numeracja hierarchiczna i zależna

### 1. Pierwszy paragraf

**Twierdzenie 1.1.** *To jest pierwsze twierdzenie.*

**Definicja 1.2.** *To jest pierwsza definicja.*

### 2. Drugi paragraf

**Twierdzenie 2.1.** *To jest drugie twierdzenie.*

**Przykład 1.** *To jest pierwszy przykład.*

```
% otoczenia-hierarchiczna-zalezna.tex
\documentclass{mwart}

\usepackage[utf8]{inputenc}
\usepackage{polski}
\usepackage{latex-ksiazka-kucharska}

\usepackage{amsthm}

\newtheorem{tw}{Twierdzenie}[section]
\newtheorem{defi}[tw]{Definicja}
\newtheorem{prz}{Przykład}

\begin{document}
\section{Pierwszy paragraf}
```

```
\begin{tw}
To jest pierwsze twierdzenie.
\end{tw}

\begin{defi}
To jest pierwsza definicja.
\end{defi}

\section{Drugi paragraf}

\begin{tw}
To jest drugie twierdzenie.
\end{tw}

\begin{prz}
To jest pierwszy przykład.
\end{prz}
\end{document}
```

Nie jest możliwe jednoczesne stosowanie numerowania hierarchicznego i zależnego przy użyciu zewnętrznego licznika. Wynika to z faktu, że polecenie `\newtheorem` może przyjmować co najwyżej jeden parametr opcjonalny na raz. Czasem jednak chcielibyśmy, aby numeracja twierdzeń była współdzielona pomiędzy kilka otoczeń, a jednocześnie aby była zależna od poziomu hierarchii.

Na szczęście, dla każdego tworzonego otoczenia numerowanego,  $\LaTeX$  tworzy automatycznie licznik o nazwie takiej samej, jak jego nazwa referencyjna. To oznacza, że możemy odwołać się (przy użyciu odpowiedniego parametru opcjonalnego) do tego licznika i w ten sposób uniknąć sytuacji, w której musielibyśmy korzystać z dwóch parametrów opcjonalnych jednocześnie.

## Nazwy twierdzeń

**Twierdzenie 1** (Zasadnicze twierdzenie). *To jest pierwsze twierdzenie.*

```
% otoczenia-nazwy.tex
\documentclass{mwart}

\usepackage[utf8]{inputenc}
\usepackage{polski}
\usepackage{latex-ksiazka-kucharska}

\usepackage{amsthm}

\newtheorem{tw}{Twierdzenie}

\begin{document}
\begin{tw}[Zasadnicze twierdzenie]
To jest pierwsze twierdzenie.
\end{tw}
\end{document}
```

Jeśli chcemy nadać twierdzeniu nazwę, możemy umieścić ją jako argument opcjonalny wywołanego środowiska. Tak zdefiniowane nazwy nie tylko są pomocne czytelnikom, ale także mogą służyć do tworzenia odpowiednich spisów.



## Spisy otoczeń numerowanych

### 1. Twierdzenia

**Twierdzenie 1.1** (Zasadnicze twierdzenie). *To jest pierwsze twierdzenie.*

**Definicja 1.** *To jest pierwsza definicja.*

**Przykład 1.** *To jest przykład.*

### Lista twierzeń i definicji

|     |                                                |   |
|-----|------------------------------------------------|---|
| 1.1 | Twierdzenie (Zasadnicze twierdzenie) . . . . . | 1 |
| 1   | Definicja . . . . .                            | 1 |

```
% otoczenia-spis.tex
\documentclass{mwart}

\usepackage[utf8]{inputenc}
\usepackage{polski}
\usepackage{latex-ksiazka-kucharska}

\usepackage{amsthm}

\usepackage{thmtools}
\renewcommand{\listtheoremname}{Lista twierzeń i definicji}

\newtheorem{tw}{Twierdzenie}[section]
\newtheorem{defi}{Definicja}
\newtheorem{prz}{Przykład}
```

```

\begin{document}
\section{Twierdzenia}

\begin{tw}[Zasadnicze twierdzenie]
To jest pierwsze twierdzenie.
\end{tw}

\begin{defi}
To jest pierwsza definicja.
\end{defi}

\begin{prz}
To jest przykład.
\end{prz}

\listoftheorems[ignoreall,show={tw,defi}]
\end{document}

```

Jak wiemy,  $\LaTeX$  pozwala w dość łatwy sposób tworzyć spisy treści, tabel i rysunków. Tworzenie listy twierdzeń umożliwia pakiet `thmtools`, który należy dołączyć w preambule.

Warto ponownie zdefiniować polecenie `\listtheoremname` przy użyciu instrukcji `\renewcommand` – odpowiada ono za to, jaki tytuł będzie miał spis. Po tej operacji wystarczy w wybranym miejscu dokumentu (z reguły na końcu) umieścić polecenie `\listoftheorems`. Polecenie to domyślnie wypisuje listę wszystkich otoczeń numerowanych występujących w dokumencie. Jeśli chcemy ograniczyć się jedynie do niektórych rodzajów, możemy je wywołać z odpowiednimi parametrami opcjonalnymi. Zastosowany w przykładzie parametr `ignoreall` wskazuje, że nie chcemy wypisywać żadnego twierdzenia, zaś przy pomocy parametru `show` definiujemy listę nazw referencyjnych otoczeń, które mają być uwzględnione w spisie.

Pakiet `thmtools` daje o wiele większe możliwości, do zapoznania się z którymi zachęcamy ambitniejszych Czytelników.

```

\listtheoremname
\renewcommand
\listoftheorems

```

## Dowody

**Twierdzenie 1.** *To jest pierwsze twierdzenie.*

*Dowód.* To jest dowód twierdzenia. □

**Twierdzenie 2.** *To jest drugie twierdzenie.*

*Dowód.* Dowód wynika wprost ze wzoru

$$a^2 + b^2 = c^2. \quad \square$$

```
% otoczenia.dowody.tex
\documentclass{mwart}

\usepackage[utf8]{inputenc}
\usepackage{polski}
\usepackage{latex-ksiazka-kucharska}

\usepackage{amsthm}

\newtheorem{tw}{Twierdzenie}

\begin{document}
\begin{tw}
To jest pierwsze twierdzenie.
\end{tw}

\begin{proof}
To jest dowód twierdzenia.
\end{proof}
```

```

\begin{tw}
To jest drugie twierdzenie.
\end{tw}

\begin{proof}
Dowód wynika wprost ze wzoru $[a^2 + b^2 = c^2.\qedhere\]$
\end{proof}
\end{document}

```

Pakiet `amsthm` dostarcza nam otoczenie `proof`, służące do umieszczania w tekście dowodów matematycznych. Jego cechą charakterystyczną jest nie tylko słowo *Dowód* na początku, ale także symbol końca dowodu (kwadrat), umieszczany automatycznie jako ostatni element w otoczeniu. Jeśli jednak dowód kończy się wzorem wystawionym lub wyliczeniem, należy wska-

W praktyce zasada ta dotyczy wszystkich otoczeń stosowanych wewnątrz dowodu.

znać miejsce umieszczenia symbolu końca dowodu ręcznie, stosując polecenie `\qedhere`. Sprawia to, że symbol ten umieszczany jest na wysokości ostatniego wiersza dowodu, a nie pod nim. Dowody nie zostaną umieszczone w spisach twierdzeń, nawet jeśli umiemy nazwę referencyjną `proof` na liście będącej wartością parametru opcjonalnego show polecenia `\listoftheorems`, opisanego wcześniej.

`\qedhere`

`\listoftheorems`

## Zmiana symbolu końca dowodu

**Twierdzenie 1.** *To jest pierwsze twierdzenie.*

*Dowód.* To jest dowód twierdzenia. Q.E.D.

```

% otoczenia-symbol.tex
\documentclass{mwart}

\usepackage[utf8]{inputenc}
\usepackage{polski}
\usepackage{latex-ksiazka-kucharska}

\usepackage{amsthm}
\newtheorem{tw}{Twierdzenie}

\renewcommand{\qedsymbol}{Q.E.D.}

\begin{document}
\begin{tw}
To jest pierwsze twierdzenie.
\end{tw}

\begin{proof}
To jest dowód twierdzenia.
\end{proof}
\end{document}

```

Aby zmienić symbol końca dowodu, należy zdefiniować ponownie polecenie `\qedsymbol` przy użyciu znanej nam już instrukcji `\renewcommand`. Wywołanie polecenia `\renewcommand{\qedsymbol}{}` spowoduje, że stosowany będzie znak pusty.

```

\qedsymbol
\renewcommand

```

## Rodzaje otoczeń numerowanych

**Twierdzenie 1** (Zasadnicze twierdzenie). *To jest pierwsze twierdzenie.*

**Definicja 1.** To jest pierwsza definicja.

*Przykład 1.* To jest przykład.

*Uwaga 1.* To jest uwaga.

```
% otoczenia-rodzaje.tex
\documentclass{mwart}

\usepackage[utf8]{inputenc}
\usepackage{polski}
\usepackage{latex-ksiazka-kucharska}

\usepackage{amsthm}

\theoremstyle{plain}
\newtheorem{tw}{Twierdzenie}

\theoremstyle{definition}
\newtheorem{defi}{Definicja}

\theoremstyle{remark}
\newtheorem{prz}{Przykład}
\newtheorem{uwa}{Uwaga}

\begin{document}
\begin{tw}[Zasadnicze twierdzenie]
```

```
To jest pierwsze twierdzenie.
\end{tw}

\begin{defi}
To jest pierwsza definicja.
\end{defi}

\begin{prz}
To jest przykład.
\end{prz}

\begin{uwa}
To jest uwaga.
\end{uwa}
\end{document}
```

Tworząc otoczenia numerowane wewnątrz dokumentu możemy stosować jeden z trzech dostępnych domyślnie stylów: `plain`, `definition` oraz `remark`. Wywołanie polecenia `\theoremstyle` z jednym parametrem obowiązkowym, będącym nazwą stylu, powoduje, że dla wszystkich otoczeń numerowanych zdefiniowanych później stosowany będzie wskazany styl. Jeśli nie wskazano inaczej, stosowany jest styl `plain` (porównaj z wcześniejszymi przykładami).

Pakiet `amsthm` daje bardzo szerokie możliwości tworzenia własnych stylów twierdzeń. Zainteresowanych Czytelników zachęcamy do zapoznania się z dokumentacją pakietu, szczegółowo opisującą wiele parametrów, które można ustawiać.

`\theoremstyle`

## Zamiana numeru otoczenia i jego nazwy miejscami

1. **Twierdzenie.** *To jest pierwsze twierdzenie.*
2. **Twierdzenie.** *To jest drugie twierdzenie.*
3. **Definicja.** *To jest pierwsza definicja.*

```
% otoczenia-zamiana.tex
\documentclass{mwart}

\usepackage[utf8]{inputenc}
\usepackage{polski}
\usepackage{latex-ksiazka-kucharska}

\usepackage{amsthm}
\makeatletter
\def\swappedhead#1#2#3{%
 \thmnumber{#2.}%
 \thmname{\@ifnotempty{#2}{~}#1}%
 \thmnote{ {\the\thm@notefont(#3)}}}
\makeatother
\swapnumbers

\newtheorem{tw}{Twierdzenie}
\newtheorem{defi}[tw]{Definicja}

\begin{document}
\begin{tw}
To jest pierwsze twierdzenie.
\end{tw}
```



```
\begin{tw}
To jest drugie twierdzenie.
\end{tw}

\begin{defi}
To jest pierwsza definicja.
\end{defi}
\end{document}
```

Niekiedy, z różnych przyczyn, chcielibyśmy, aby numer twierdzenia znajdował się przed jego nazwą. Aby osiągnąć taki efekt, należy wywołać w preambule polecenie `\swapnumbers` związane z pakietem `amsthm`.

Niestety, zwyczajem anglosaskim jest w takiej sytuacji pisanie numeru twierdzenia bez kropki. W języku polskim po liczebnikach porządkowych winno się kropkę stawiać. Pakiet `amsthm` nie pozwala na łatwe osiągnięcie takiego efektu – oczywiście nie oznacza to, że nie da się go uzyskać wcale. W powyższym przykładzie, zaraz po zadeklarowaniu, że będziemy korzystać z pakietu `amsthm`, tak przeddefiniowaliśmy jedno z wewnętrznych poleceń tego pakietu, by kropki pojawiły się tam, gdzie trzeba. Zauważmy również przy okazji, że zastosowaliśmy wspólną numerację twierdzeń i definicji – jak się zdaje, jest to rozsądne rozwiązanie przy „odwróconej” numeracji.

`\swapnumbers`

# Wzory i formuły matematyczne

Jak to już zostało wspomniane w *Elementarzu*, jedną z mocnych stron  $\TeX$ a (a przez to i  $\LaTeX$ a) są zaawansowane możliwości składu wzorów i formuł matematycznych. Teraz nadszedł moment, w którym będziemy mogli przekonać się o tym na własne oczy.

## Prosty wzór

Ponieważ skład tekstu w języku naturalnym i skład matematyczny różnią się znacząco, w  $\TeX$ u używa się do tego ostatniego osobnego tzw. *trybu matematycznego*. Jak się okaże, niektóre polecenia działają wyłącznie w tym trybie, inne zaś w ogóle w nim nie działają. Dzięki temu łatwiej wychwycić częsty błąd polegający na tym, że zapomnieliśmy włączyć (lub wyłączyć) tryb matematyczny – najczęściej w takich sytuacjach  $\LaTeX$  prędzej czy później trafia na „nielegalne” polecenie i zgłasza błąd.

Najprostszym sposobem włączenia trybu matematycznego jest użycie symbolu dolara ( $\$$ ). Fragmenty pomiędzy takimi symbolami będą składane jako wzory. Niektórym nie odpowiada to, że ten sam symbol służy zarówno do *włączania*, jak i *wyłączania* trybu matematycznego; dla tych osób przewidziano polecenia  $\left($  i  $\right)$ .

$2+2=4$   $\quad$   $\$2+2=4\$$   
 $\quad$   $\quad$   $\left(2 + 2 = 4\right)$

|                                    |
|------------------------------------|
| $2+2=4$ $2 + 2 = 4$<br>$2 + 2 = 4$ |
|------------------------------------|

Zauważmy, że prosty wzór złożony jako *tekst* (czyli bez znaczników trybu matematycznego) wygląda bardzo kiepsko: między liczbami a znakiem dodawania (i znakiem równości) nie ma w ogóle odstępów. (W przypadku bardziej

$\$$

$\left($   
 $\right)$

skomplikowanych wzorów bywa jeszcze gorzej.) Ten sam wzór wpisany pomiędzy znakami dolara (lub poleceniami `\(...\)`) wygląda już zupełnie inaczej. W przykładzie występuje też polecenie `\quad`, o którym jest mowa na stronie 204.

`\quad`

Jak można się domyślić z powyższego przykładu, w trybie matematycznym  $\LaTeX$  ignoruje spacje (oczywiście, nadal są one potrzebne, aby oddzielić polecenie  $\LaTeX$ a od następującego po nim tekstu), samodzielnie dbając o właściwe odstępy między elementami formuł.

## Wzory wycentrowane

Wzór w tekście (np.  $\frac{1}{2} - \frac{1}{3} = \frac{1}{6}$ ) możemy umieścić dość łatwo.

Czasem chcemy, żeby wzór był wycentrowany:

$$\frac{1}{2} - \frac{1}{3} = \frac{1}{6}. \quad (1)$$

A oto ponownie wzór (1), tym razem bez numeru:

$$\frac{1}{2} - \frac{1}{3} = \frac{1}{6}.$$

Można go zapisać równoważnie w postaci

$$\frac{1}{2} - \frac{1}{3} = \frac{1}{6}.$$

```
% matematyka-wycentrowany.tex
\documentclass{mwart}
```

```

\usepackage[utf8]{inputenc}
\usepackage{polski}
\usepackage{latex-ksiazka-kucharska}
\usepackage{amsmath}
\usepackage{amssymb}

\begin{document}
Wzór w tekście (np. $\frac{1}{2}-\frac{1}{3}=\frac{1}{6}$)
możemy umieścić dość łatwo.

Czasem chcemy, żeby wzór był wycentrowany:
\begin{equation}
\label{ulamki}
\frac{1}{2}-\frac{1}{3}=\frac{1}{6}\text{.}
\end{equation}

A~oto ponownie wzór~\eqref{ulamki}, tym razem bez
numera:
\begin{equation*}
\frac{1}{2}-\frac{1}{3}=\frac{1}{6}\text{.}
\end{equation*}
Można go zapisać równoważnie w postaci
\[\frac{1}{2}-\frac{1}{3}=\frac{1}{6}\text{.}\]
\end{document}

```

Wzory otoczone symbolami dolara lub poleceniami  $\backslash(\dots\backslash)$  są składane wewnątrz akapitu. Wzory wycentrowane składa się w otoczeniu `equation`, jeżeli potrzebujemy ich numeracji (do której oczywiście możemy się odwoływać przy użyciu poleceń `\eqref` lub `\ref`), lub `equation*`, jeżeli nie chcemy ich numerować. Zamiast środowiska `equation*` można też użyć równoważnej mu, ale krótszej konstrukcji `\[...\]`.

Zwróćmy uwagę, że niektóre wyrażenia matematyczne (np. ułamki) wyglądają inaczej we wzorach w tekście lub np. pod znakiem pierwiastka, a inaczej w wycentrowanych: ogólnie rzecz ujmując, formuły matematyczne w tekście są składane przez  $\text{\LaTeX}$  tak, by zajmowały możliwie mało miejsca

`\eqref`

• Polecenie `\eqref` działa tak samo, jak polecenie `\ref`, ale pozwala na automatyczne umieszczenie w tekście odwołania otoczonego nawiasami okrągłymi.

`\ref`

w pionie (w przeciwnym wypadku powodowałyby zwiększenie odstępu między wierszami, co nie wygląda zbyt estetycznie). Możemy to zmienić przy użyciu poleceń `\limits` i `\nolimits`:

```
\sum\nolimits_{n=0}^{\infty}
\sum\limits_{n=0}^{\infty}
```

$$\sum_{n=0}^{\infty} \sum_{n=0}^{\infty}$$

```
\limits
\nolimits
```

Zgodnie z amerykańskimi zwyczajami typograficznymi,  $\LaTeX$  składa granice całkowania obok symbolu całki (zapisywanego przy użyciu polecenia `\int`) zamiast pod i nad nim. Aby to zmienić, można pisać `\int\limits` zamiast `\int`. O wiele łatwiej jest jednak wywołać pakiet `amsmath` z opcją `intlimits`, to znaczy `\usepackage[intlimits]{amsmath}`.

```
\int
```

Wreszcie, należy pamiętać o tym, że jeżeli wzór (również wycentrowany) znajduje się w środku zdania, należy go traktować jako jego część (na przykład wzór (1) z przykładu jest zdaniem podrzędnym, w którym rolę orzeczenia pełni symbol  $=$ ), co w szczególności oznacza poprawne użycie znaków interpunkcyjnych. W przypadku wzorów w tekście znaki te nie należą do wzoru, zatem należy je pisać poza znacznikami `$...$` (lub `\(...\)`). W przypadku wzorów wycentrowanych musimy włączyć interpunkcję do nich (inaczej trafiłaby ona do osobnego wiersza, co byłoby błędem), ale należy umieścić je jako argument polecenia `\text` (więcej o nim można przeczytać w paragrafie *Teksty we wzorach* na stronie 205).

Pakiety `amssymb` (zwiększający znacząco liczbę dostępnych symboli) i `amsmath` (zwiększający możliwości tworzenia m.in. wielowierszowych ciągów formuł) nie są co prawda konieczne do składania najprostszyc wzorów, ale przydają się na tyle często, że włączanie ich do dokumentu zawsze, gdy pojawiają się w nim treści matematyczne, wydaje się nie być pozbawione sensu. **Uwaga.** W tym rozdziale w większości przypadków odstąpimy od dotychczasowego zwyczaju, w którym przykłady były kompletnymi dokumentami. Przykłady składu formuł matematycznych, niebędące samodzielnyimi dokumentami, można będzie traktować jako wpisane do pewnego ogólnego szablonu. W trosce o czytelność przykładów będziemy najczęściej pomijać znaczniki początku i końca trybu matematycznego; zdecydowana większość przykładów będzie składana tak, jakby była umieszczona wewnątrz akapitu.

Na przykład, poniższy fragment powinien być utożsamiany z następującym po nim dokumentem.

$$2+2=4$$

$$2 + 2 = 4$$

```
\documentclass{mwart}
\usepackage{latex-ksiazka-kucharska}
\usepackage{amsmath}
\usepackage{amssymb}

\begin{document}
\ (2+2=4\)
\end{document}
```

## Ułamki, potęgi i indeksy dolne. Symbol Newtona.

$$\frac{1}{2}(a_1+a_2)^2$$

$$\frac{1}{2}(a_1 + a_2)^2$$

Powyższy przykład wymaga kilku komentarzy. Odnotujmy, że ułamek „tekstowy” (pisany mniejszym pismem) możemy umieścić w dokumencie przy użyciu polecenia `\tfrac`, a ułamek „duży” (taki, jak we wzorach wycentrowanych) – przy użyciu polecenia `\dfrac`.

Może się też zdarzyć, że będziemy potrzebować indeksów górnych i dolnych jednocześnie – można je wówczas podać w dowolnej kolejności (dla  $\text{\LaTeX}$ a nie ma różnicy między  $x_{\{0\}}^{\{2\}}$  a  $x^{\{2\}}_{\{0\}}$ , choć pierwszy wariant jest bliższy sposobowi, w jaki odczytalibyśmy taką formułę). Niekiedy też jest potrzeba zapisania indeksów piętrowych. Konstrukcja  $x_{\{n\}}_{\{k\}}$  jest niejednoznaczna (nie wiadomo, czy indeks  $\{k\}$  stoi przy całym wyrażeniu  $x_{\{n\}}$ , czy przy indeksie  $\{n\}$ ) i spowoduje błąd kompilacji; w takiej sytuacji należy napisać na przykład  $x_{\{n\}_{\{k\}}}$ .

Kolejną kwestią związaną z indeksami jest oszczędność miejsca i wysiłku: jeżeli indeks jest *pojedynczym znakiem* (lub pojedynczym poleceniem  $\text{\LaTeX}$ a),

`\tfrac`  
`\dfrac`

nie ma potrzeby otaczania go nawiasami klamrowymi – można np. napisać po prostu  $(a_1+a_2)^2$ .

`\binom{n}{k} = \binom{n}{n-k}`

$$\binom{n}{k} = \binom{n}{n-k}$$

Skoro mowa o ułamkach, odnotujmy również, że symbol Newtona, typograficznie pokrewny symbolowi ułamka, składa się w  $\text{\LaTeX}$  przy użyciu polecenia `\binom` (przez analogię do `\frac` i `\dfrac` mamy też `\tbinom` i `\dbinom`):

`\binom`  
`\tbinom`  
`\dbinom`

## Nawiasy

Już w szkole podstawowej dowiadujemy się, że działania arytmetyczne mają swoją hierarchię, i że możemy modyfikować kolejność wykonywania działań we wzorach przy użyciu *nawiasów*. Przy pisaniu w  $\text{\LaTeX}$  istotne są dwie kwestie: jak wpisywać same nawiasy, i jak zmienić ich wielkość.

Co do pierwszej kwestii, sprawa jest prosta: nawiasy okrągłe i kwadratowe wpisujemy „zwyczajnie”, nawiasy klamrowe (które mają szczególne znaczenie dla  $\text{\LaTeX}$ a) poprzedzamy znakiem odwróconego ukośnika, natomiast bardziej wyrafinowane rodzaje nawiasów wpisujemy, korzystając z odpowiednich poleceń.

`\langle [ ( \lvert x`  
`\rvert ) ] \rangle \rangle`

$$\langle \langle \langle \langle x \rangle \rangle \rangle \rangle$$

Kwestia wielkości nawiasów jest subtelniejsza. Możemy poprzedzić nawiasy poleceniami `\left` i `\right` – wówczas  $\text{\LaTeX}$  dobierze ich wielkość tak, by objęły całe wyrażenie znajdujące się w środku.

`\left[\left(x^2\right)^2\right]`

$$\left[ \left( x^2 \right)^2 \right]$$

Często się jednak zdarza, że automatycznie dobrana wielkość nawiasów nie jest właściwa. Na przykład, powyższe wyrażenie wygląda trochę lepiej, gdy wszystkie nawiasy są zwykłej wielkości, lub gdy wyłącznie nawiasy kwadratowe są nieco większe:

Kreski oznaczające wartość bezwzględną można uzyskać też pisząc po prostu np.  $|x|$ , choć wówczas  $\text{\LaTeX}$  może nie być w stanie zorientować się, która z nich jest „lewa”, a która „prawa”; w razie nieprawidłowych odstępów należy wówczas napisać `\lvert x\rvert`.

`\left`  
`\right`

`[(x^2)^2] \quad`  
`\left[(x^2)^2\right]`

$$[(x^2)^2] \quad [(x^2)^2]$$

W przypadku, gdy chcemy mieć pełną kontrolę nad rozmiarem nawiasów, z pomocą przychodzą polecenia składające nawiasy żądanej wielkości.

`\Biggl(\biggl(\Bigl(\bigl((x`  
`)\bigr)\Bigr)\biggr)\Biggr)`

$$\left(\left(\left(\left(x\right)\right)\right)\right)$$

Właściwe dobranie wielkości nawiasów bywa niełatwe, ale warto włożyć w nie trochę wysiłku, by nasze wzory wyglądały bardzo profesjonalnie. W poniższych przykładach użycie komend `\left` i `\right` powoduje, że kreski pionowe są zbyt małe (w pierwszym wzorze) lub nawiasy zbyt duże (w drugim wzorze). O komendzie `\!` jest mowa na stronie 203.

`\left|\left|x-1\right|+1\right|`

$$||x-1|+1|$$

`\bigl|\lvert x-1\rvert+1\bigr|`

$$|x-1|+1|$$

`\left(\sum_{n=1}^{\infty}`  
`\lvert x_n\rvert^p\right)^{\!\!\!1/p}`

$$\left(\sum_{n=1}^{\infty}|x_n|^p\right)^{1/p}$$

`\biggl(\sum_{n=1}^{\infty}`  
`\lvert x_n\rvert^p\biggr)^{\!\!\!1/p}`

$$\left(\sum_{n=1}^{\infty}|x_n|^p\right)^{1/p}$$

Zauważmy przy okazji, że choć – jak to zostało wspomniane wcześniej – lepiej jest używać poleceń `\lvert` i `\rvert` zamiast symbolu `|`, gdyż ten ostatni jest niejednoznaczny, to jednak w przypadku użycia go po takich poleceniach, jak `\bigl` czy `\right` problem znika – te komendy jednoznacznie określają bowiem, czy mamy do czynienia z kreską otwierającą, czy zamykającą wyrażenie.

`\lvert`  
`\rvert`



## Symbole działań, relacji, zbiorów itp.

Przez kilka wieków rozwoju notacji algebraicznej matematycy zdołali wy-  
myślić całkiem sporo różnych symboli oznaczających różne rzeczy. Każdy  
system składu tekstów matematycznych powinien pozwolić umieścić przy-  
najmniej większość z nich w dokumencie. Nie inaczej jest z  $\text{\LaTeX}$ em. Nie bę-  
dziemy tu przytaczać *in extenso* tabel z komendami produkującymi te symbo-  
le (na to jest miejsce w *Dodatku A*), ale poprzestaniemy na kilku przykładach  
ilustrujących szczególnie często pojawiające się oznaczenia.

|                                                     |                                         |
|-----------------------------------------------------|-----------------------------------------|
| <code>x \in (0,1) \cup (1,2)</code>                 | $x \in (0,1) \cup (1,2)$                |
| <code>\pi \approx 3{,}1416</code>                   | $\pi \approx 3,1416$                    |
| <code>\sqrt{2} \approx 2{,}4142</code>              | $\sqrt{2} \approx 2,4142$               |
| <code>\exists y \geq 0; \forall x : x \leq y</code> | $\exists y \geq 0 \forall x : x \leq y$ |
| <code>f \colon X \to Y</code>                       | $f : X \rightarrow Y$                   |
| <code>f'(x) = x^2</code>                            | $f'(x) = x^2$                           |
| <code>(20 \pm 1) \cdot (20 \mp 1)</code>            | $(20 \pm 1) \cdot (20 \mp 1)$           |

Zwróćmy uwagę na pewne subtelności. Po pierwsze, przecinek (i kilka in-  
nych symboli, np. średnik) jest traktowany przez  $\text{\LaTeX}$ a jako *znak przestanko-  
wy*, czyli w trybie matematycznym pojawia się po nim wąska spacja. Jeżeli  
chcemy tego uniknąć, należy wziąć taki symbol w nawiasy klamrowe.

Po drugie, zdarza się, że trzeba podpowiedzieć  $\text{\LaTeX}$ owi, jakie powinien  
wstawiać odstępy. Tak jest w przypadku wzoru z kwantyfikatorami, w któ-  
rym wstawiliśmy odstęp poleceniem `\;`. Podobnie jest w przypadku znaku  
dwukropka, który – gdy jest używany jako symbol oddzielający nazwę funk-  
cji od jej dziedziny i przeciwdziedziny – wstawia się poleceniem `\colon`.

Takich subtelności jest więcej; zainteresowanym czytelnikom polecamy  
lekturę książki  *$\text{\TeX}$ . Przewodnik użytkownika* autorstwa Donalda E. Knutha,  
a pozostałych nie będziemy już zanudzać.

Można też napisać  
`\usepackage{icomma}`  
w preambule i odróżnić  
przecinek dziesiętny od  
znaku przestankowego  
umieszczaniem bądź nie  
spacji po nim w pliku źró-  
dłowym.

Więcej o odstępach w try-  
bie matematycznym powie-  
my na s. 203.

`\;`  
`\colon`

## Popularne funkcje

W trybie matematycznym litery alfabetu łacińskiego składane są kursywą (a jeżeli font jest w takową wyposażony, to nawet specjalną *kursywą matematyczną*, która od zwykłej różni się głównie odstępami międzyliterowymi). Zgodnie z tradycją, nazwy funkcji zapisuje się z kolei pismem prostym – w ten sposób można odróżnić np. iloczyn czterech czynników  $s, i, n, x$  ( $sinx$ ) od sinus liczby  $x$  ( $\sin x$ ). Spora liczba często spotykanych funkcji ma swoje polecenia  $\LaTeX$ owe:

$$\backslash \sin ^2 x + \backslash \cos ^2 x = 1$$

$$\sin ^2 x + \cos ^2 x = 1$$

$$\backslash \sup \backslash \operatorname{tg} \backslash \log x$$

$$\sup \operatorname{tg} \log x$$

W przypadku, gdy w standardowym zestawie brak oznaczenia jakiejś rzadziej używanej funkcji, można użyć komendy  $\backslash \operatorname{operatorname}$ .

$$\backslash \operatorname{operatorname}\{\operatorname{conv}\}A$$

$$\operatorname{conv} A$$

Jeżeli oznaczenie takie będzie używane częściej, warto zdefiniować nowe polecenie. Jeżeli w preambule wpisujemy linijkę

$$\backslash \operatorname{DeclareMathOperator}\{\operatorname{conv}\}\{\operatorname{conv}\}$$

będziemy mogli używać dalej polecenia  $\backslash \operatorname{conv}$  tak, jakby było wbudowane w  $\LaTeX$ a.

Na zakończenie odnotujmy, że niektóre z funkcji nazywanych kilkoma literami miewają indeksy dolne pisane obok nazwy funkcji (na przykład logarytm), a niektóre – pod tą nazwą (na przykład granica). W przypadku formuł

$$\backslash \log _b \backslash \lim _{x \rightarrow 0} b^x$$

$$\log _b \lim _{x \rightarrow 0} b^x$$

umieszczonych wewnątrz akapitu nie ma to znaczenia, gdyż  $\LaTeX$  i tak umieszcza wszystkie indeksy obok symboli, których dotyczą (ma to na celu uniknięcie zwiększania odstępów między wierszami), jednak przy formułach wycenrowanych jest to istotne. Aby zdefiniować nową funkcję, której indeksy powinny być umieszczone pod nią, należy zamiast  $\backslash \operatorname{DeclareMathOperator}$  użyć polecenia  $\backslash \operatorname{DeclareMathOperator*}$ .

$$\backslash \operatorname{operatorname}$$

$$\backslash \operatorname{DeclareMathOperator}$$

$$\backslash \operatorname{DeclareMathOperator*}$$

## Alfabet matematyczne

W wielu tekstach matematycznych pojawia się potrzeba użycia różnych krojów pisma. Oprócz zwykłej kursywy matematycznej pojawiają się litery pisane (kaligraficzne), pisane podwójną kreską (tradycyjnie tak oznacza się zbiory liczbowe), rzadziej wytłuszczone, a także fraktura. Wszystkie te alfabet

Fraktura zwana jest popularnie, acz niezbyt poprawnie *pismem gotyckim*.

`R\mathcal{R}\mathbb{R}`  
`\mathbf{R}\mathfrak{R}`

$\mathcal{R}\mathbb{R}\mathfrak{R}$

Oczywiście, mamy również do dyspozycji komplet liter greckich, a także kilka początkowych znaków alfabetu hebrajskiego.

`\alpha\beta\gamma\dots\omega`  
`\quad \aleph\beth\gimel`

$\alpha\beta\gamma\dots\omega \aleph\beth$

## Akcenty nad literami

Polecenia wstawiające akcenty w trybie tekstowym nie działają w trybie matematycznym (i *vice versa*). Z łatwością można jednak używać rozmaitych akcentów również w składzie matematycznym. Warto pamiętać o poleceniach `\imath` i `\jmath`, składających litery „i” oraz „j” bez kropki kursywą matematyczną.

`\hat{a} + \dot{x}`  
`+ \vec{\imath} + \bar{z}`

$\hat{a} + \dot{x} + \vec{i} + \bar{z}$

`\imath`  
`\jmath`

## Funkcje dane kilkoma wzorami

Całkiem często definiuje się w matematyce funkcję kilkoma wzorami, zależnie od tego, w jakim podzbiore dziedziny znajduje się argument. (Klasycznym przykładem jest definicja wartości bezwzględnej.) Do składu takich definicji używa się otoczenia `cases`.

```
\lvert x\rvert=\begin{cases}
x\text{,} & \text{gd}y \ $x\ge 0$ \\
-x\text{,} & \text{gd}y \ $x<0$
\end{cases}
```

$$|x| = \begin{cases} x, & \text{gd}y \ x \geq 0 \\ -x, & \text{gd}y \ x < 0 \end{cases}$$

Powyższy kod może się wydawać dość skomplikowany, ale w rzeczywistości jest całkiem prosty. Kolejne wiersze (poza ostatnim) kończą się oznaczeniem końca wiersza `\\`. Dwie części każdego wiersza oddzielone są znakiem `&`. Fragmenty pisane w języku naturalnym umieszcza się w argumencie polecenia `\text`. Wreszcie, formuły matematyczne wewnątrz argumentu tego polecenia tradycyjnie otacza się symbolami dolara (oczywiście, można również użyć poleceń `\(...\)`).

## Układy równań

Układy równań, tradycyjnie obejmowane klamrą po lewej stronie, składa się podobnie, jak wartości funkcji „sklejanych” z kilku wzorów. Możemy do tego celu wykorzystać środowisko `aligned`.

```
\left\{
\begin{aligned}
Ax + By + C &= 0 \\
(x-a)^2 + (y-b)^2 &= r^2
\end{aligned}
\right.
```

$$\begin{cases} Ax + By + C = 0 \\ (x - a)^2 + (y - b)^2 = r^2 \end{cases}$$

Zwróćmy uwagę na dwie kwestie. Po pierwsze, ponownie (por. wcześniejszy paragraf) użyliśmy symboli `&`, aby zaznaczyć miejsca, które powinny być wyrównane w obu wierszach, i polecenia `\\`, aby oznaczyć koniec wiersza. Po

drugie, otoczyliśmy nasz układ równań nawiasami: klamrowym po lewej stronie (`\left\{`) i pustym po prawej (`\right.`) – komendy `\left` i `\right` muszą występować parami!

## Ciągi równości

Dłuższe ciągi równoważnych formuł, ilustrujące bardziej złożone przekształcenia, wyglądają również podobnie do poprzedniego przykładu. Jest jednak istotna różnica: jeżeli chcemy ponumerować kolejne etapy przekształcenia (i np. odwoływać się do nich później przy użyciu poleceń `\label` oraz `\ref`), niekoniecznie będziemy chcieli zagnieździć w otoczeniu `equation` (odpowiedzialnym za numerowane wzory) otoczenia `aligned`, gdyż wówczas mielibyśmy *jeden* numer całego ciągu przekształceń. W takiej sytuacji zamiast środowiska `equation` należy użyć środowiska `align`. Należy również pamiętać, że do odwoływania się do wzorów używa się polecenia `\eqref`, które (w odróżnieniu od `\ref`) otacza numer nawiasami oraz składa

`\eqref`

$$(x + y)^2 = (x + y)(x + y) = \tag{1}$$

$$= x^2 + xy + yx + y^2 = \tag{2}$$

$$= x^2 + 2xy + y^2 \tag{3}$$

Najpierw podnieśliśmy dane wyrażenie do kwadratu (1), potem wymnożyliśmy nawiasy (2), a na końcu zredukowaliśmy wyrazy podobne (3).

```

% matematyka-ciag-1.tex
\documentclass{mwart}
\usepackage[utf8]{inputenc}
\usepackage{polski}
\usepackage{latex-ksiazka-kucharska}
\usepackage{amsmath}
\usepackage{amssymb}

\begin{document}
\begin{align}
(x+y)^2 &= (x+y)(x+y) = && \text{\label{kwadrat}} \\
&= x^2 + xy + yx + y^2 = && \text{\label{nawiasy}} \\
&= x^2 + 2xy + y^2 && \text{\label{redukcja}}
\end{align}
Najpierw podnieśliśmy dane wyrażenie do
kwadratu~\eqref{kwadrat}, potem wymnożyliśmy
nawiasy~\eqref{nawiasy}, a~na końcu zredukowaliśmy
wyrazy podobne~\eqref{redukcja}.
\end{document}

```

Może się jednak okazać, że zamiast numerować kolejne wiersze i odwoływać się do nich później, zechcemy opisać przekształcenia wraz ze wzorami. Możemy wówczas wykorzystać otoczenie `align*`, które nie numeruje wierszy, a zamiast numerów dodać krótkie opisy.

$$\begin{array}{rcl}
 (x + y)^2 & = & (x + y)(x + y) = & \text{kwadrat} \\
 & = & x^2 + xy + yx + y^2 = & \text{nawiasy} \\
 & = & x^2 + 2xy + y^2 & \text{redukcja}
 \end{array}$$

```
% matematyka-ciag-2.tex
\documentclass{mwart}
\usepackage{latex-ksiazka-kucharska}
\usepackage{amsmath}
\usepackage{amssymb}

\begin{document}
\begin{align*}
(x+y)^2 &=& (x+y)(x+y) = && && \& \text{\text{kwadrat}}\\
&=& x^2 + xy + yx + y^2 = && && \& \text{\text{nawiasy}}\\
&=& x^2 + 2xy + y^2 && && \& \text{\text{redukcja}}
\end{align*}
\end{document}
```

Wreszcie, może zająć potrzeba ponumerowania tylko niektórych wierszy w ciągu przekształceń. Wówczas można skorzystać z polecenia `\notag`.

`\notag`

$$\begin{aligned}
 (x + y)^2 &= (x + y)(x + y) = & (1) \\
 &= x^2 + xy + yx + y^2 = \\
 &= x^2 + 2xy + y^2 & (2)
 \end{aligned}$$

Najpierw podnieśliśmy dane wyrażenie do kwadratu (1), potem wymnożyliśmy nawiasy, a na końcu zredukowaliśmy wyrazy podobne (2).

```
% matematyka-ciag-3.tex
\documentclass{mwart}
\usepackage[utf8]{inputenc}
```

```

\usepackage{polski}
\usepackage{latex-ksiazka-kucharska}
\usepackage{amsmath}
\usepackage{amssymb}

\begin{document}
\begin{align}
(x+y)^2 &= (x+y)(x+y) = && \label{kwadrat} \\
&= x^2 + xy + yx + y^2 = && \notag \\
&= x^2 + 2xy + y^2 && \label{redukcja}
\end{align}
Najpierw podnieśliśmy dane wyrażenie do
kwadratu~\eqref{kwadrat}, potem wymnożyliśmy
nawiasy, a~na końcu zredukowaliśmy
wyrazy podobne~\eqref{redukcja}.
\end{document}

```

## Wzory związane ze sobą

Opisany w poprzednim paragrafie sposób nadaje się do wzorów, które powinny być wyrównane np. na znaku równości. Zdarza się jednak, że nie zależy nam na wyrównaniu (lub zgoła go nie chcemy). Tak może być m.in. wówczas, gdy chcemy złożyć kilka formuł, jedna pod drugą, przy czym niektóre z nich są o wiele krótsze niż inne. W takiej sytuacji przydaje się środowisko `gather` (lub jego odpowiednik `gather*`, w którym wzory nie są numerowane).

```

\begin{gather}
(x-a)^2+(y-b)^2=r^2 \\
y=px+q
\end{gather}

```

$$(x - a)^2 + (y - b)^2 = r^2 \quad (1)$$

$$y = px + q \quad (2)$$



Nietrudno wyobrazić sobie, jak dziwnie wyglądałyby powyższe formuły, gdyby użyć środowiska `align` i wyrównać je na znakach równości.

Jeżeli powyższe sposoby składu formuł rozciągających się na więcej niż jeden wiersz nie wystarczają dla naszych potrzeb, najlepiej skorzystać z bogactwa możliwości kryjących się w pakiecie `amsmath`, którego dokumentacja podaje szereg przykładów.

## Macierze

Jak wiadomo, macierze to po prostu tabelki zawierające wyrażenia matematyczne. [Nic więc dziwnego, że opis macierzy w pliku  \$\LaTeX\$ owym jest podobny do opisu tabeli.](#)

```
\begin{pmatrix}
 a & b \\
 c & d
\end{pmatrix}
```

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

Autorzy spieszą donieść, że znają poprawną definicję, ale nie jest ich celem wywoływanie u czytelników koszmarów sennych.

Litera `p` w nazwie środowiska `pmatrix` pochodzi od angielskiego *parentheses*, czyli nawiasy; analogiczne środowiska `bmatrix`, `vmatrix` składają tabelki otoczone nawiasami kwadratowymi lub pionowymi kreskami, [zaś](#) w przypadku konieczności użycia jakichś nietypowych nawiasów mamy do dyspozycji środowisko `matrix`, składające macierz bez żadnych nawiasów, i polecenia `\left` i `\right`, wstawiające *dowolne* nawiasy. Zdarza się też, że wygodnie jest umieścić niewielką macierz we wzorze wewnątrz akapitu; wówczas można wykorzystać środowisko `smallmatrix`:

```
\bigl(\begin{smallmatrix}
 1 & 0 \\
 0 & 1
\end{smallmatrix}\bigr)
```

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Jest takich środowisk jeszcze kilka – wszystkie opisane są w dokumentacji pakietu `amsmath`



Jeżeli wolimy tradycję anglosaską – kropki nieco bardziej od siebie oddalone i umieszczone na wysokości połowy małej litery, gdy znajdują się np. między symbolami dodawania – należy użyć polecenia `\dots`, a w rzadkich przypadkach, gdy wybiera ona kropki na niewłaściwej wysokości – poleceń `\ldots` i `\cdots`.

$$1 + 2 + \dots + n = \frac{n(n + 1)}{2}$$

$$1 + 2 + \dots + n = \frac{n(n+1)}{2}$$

$$\mathbb{N} = \{ 1, 2, 3, \dots \}$$

$$\mathbb{N} = \{1, 2, 3, \dots\}$$

`\dots`

`\ldots`

`\cdots`

`\ldots` (*lowered dots*) składa kropki na wysokości linii podstawowej, zaś `\cdots` (*centered dots*) na wysokości linii środkowej. W istocie, zalecane jest używanie jeszcze innych poleceń; po szczegóły odsyłamy do dokumentacji pakietu `amsmath` oraz na s. 215.

## Odstępy w trybie matematycznym

Mimo, że algorytm  $\text{\LaTeX}$ a odpowiedzialny za wstawianie właściwej wielkości odstępów między częściami składowymi formuł matematycznych jest starannie zaprojektowany, by robił to poprawnie w zdecydowanej większości przypadków, zdarza się, że trzeba te odstępy ręcznie modyfikować. Bywa również, że (głównie we wzorach wycentrowanych) zachodzi potrzeba oddzielenia kilku podformuł większym odstępem. W takich przypadkach przydają się polecenia wstawiające odstępy różnych rozmiarów, zarówno dodatnich, jak i ujemnych (te ostatnie pozwalają *zmniejszyć* odległość między elementami formuły). Polecenia te zebraliśmy w *tabeli 5*.

Tabela 5. Polecenia wstawiające odstępy poziome

| Polecenie                                | Odstęp               | Polecenie                                   | Odstęp              |
|------------------------------------------|----------------------|---------------------------------------------|---------------------|
| <code>\,</code> <code>\thinspace</code>  | $\rightarrow \kappa$ | <code>\!</code> <code>\negthinspace</code>  | $\leftarrow \kappa$ |
| <code>\:</code> <code>\medspace</code>   | $\rightarrow \kappa$ | <code>\!</code> <code>\negmedspace</code>   | $\leftarrow \kappa$ |
| <code>\;</code> <code>\thickspace</code> | $\rightarrow \kappa$ | <code>\!</code> <code>\negthickspace</code> | $\leftarrow \kappa$ |
| <code>\quad</code>                       | $\rightarrow \kappa$ |                                             |                     |
| <code>\qquad</code>                      | $\rightarrow \kappa$ |                                             |                     |

Wiedząc już, jak wstawić odstępy, popatrzmy, kiedy je wstawiać. Oto kilka przykładów sytuacji, gdy trzeba wspomóc  $\LaTeX$ a sugestią zwiększenia (bądź zmniejszenia) odstępu.

`\int_0^\infty f(x)\, \text{textup}{d}x`

$$\int_0^\infty f(x) dx$$

`\dfrac{n!}{k!(n-k)!}`

$$\frac{n!}{k!(n-k)!}$$

`[\,0,1) \quad \sqrt{x}`  
`\quad x^2!/2`

$$[0,1) \quad \sqrt{x} \quad x^2/2$$

We wszystkich tych przypadkach niewstawienie odstępu nie byłoby może błędem, ale pogorszyłoby wygląd formuł.

Zauważmy również, że w pierwszym z powyższych równań litera „d” (oznaczająca operator różniczkowy) jest złożona pismem prostym (użyliśmy do tego polecenia `\textup`). Nie wszyscy są zgodni co do tej konwencji, ale nam wydaje się ona rozsądna.

Jak już wspomnieliśmy, drugim typem sytuacji wymagającym ręcznej ingerencji w domyślne odstępy jest sytuacja, w której chcemy oddzielić kilka podformuł odstępem większych rozmiarów, co mocno polepsza czytelność tekstu. Jak można się domyślić z powyższych przykładów (i tabeli 5), służą do tego polecenia `\quad`, wstawiające odstęp szerokości *fired*, oraz `\qquad`, wstawiające odstęp szerokości dwóch *fired*. Nie ma ogólnych norm, które pozwalałyby podjąć decyzję, czy odstęp jednego, czy dwóch *fired* będzie właściwy; wydaje się, że rozsądną regułą jest wstawianie odstępu jednego *fired* i dopiero, gdy będzie on niewystarczający, zastosowanie odstępu większego. Obejrzyjmy to na przykładach.

`\binom{n}{k} = \frac{n!}{k!(n-k)!},`  
`\quad 0 \leq k \leq n`

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}, \quad 0 \leq k \leq n$$

`\textup`

Nie jest to książka o matematyce, więc pozwolimy sobie nie wnikać zbyt głęboko w szczegóły.

`\quad`

Jeden *fired* to kwadrat o boku równym stopniowi pisma, zatem przy składzie rozmiarem 10 pt polecenie `\quad` wstawi odstęp 10 pt.

`\qquad`



strukturę formuły: wewnątrz formuły mamy tu fragment tekstu, a wewnątrz tego fragmentu tekstu mamy dwie kolejne formuły.

Innym przypadkiem, gdy możemy potrzebować fragmentu tekstu wewnątrz formuły, są wzory ze słowami w indeksie dolnym. W takim przypadku sprawa jest prosta.

$$v_{\{\text{średnia}\}} = \frac{s}{t}$$

$$v_{\text{średnia}} = \frac{s}{t}$$

## Matematyka i tekst – zgrabne połączenie

Doświadczenie uczy, że jedną z najtrudniejszych rzeczy w składzie tekstu z zawartością matematyczną jest rozwiązywanie problemów *na styku* matematyki i typografii. W takich przypadkach trzeba polegać na wyczuciu poprawności językowej i w ogóle estetyce. Poniżej zebraliśmy kilka reguł pomagających ulepszyć skład tekstów o matematyce.

- Zdania nie powinno się rozpoczynać od symbolu matematycznego. Zamiast pisać

...takie same.  $f$   
jest więc injekcją.

...takie same.  $f$  jest więc  
injekcją.

lepiej jest napisać

...takie same. Funkcja  $f$   
jest więc iniekcją.

...takie same. Funkcja  $f$   
jest więc iniekcją.

(zauważmy, że zadbaliliśmy przy okazji o to, by kolejny wiersz nie rozpoczął się od oznaczenia „ $f$ ”).

- Dwa wzory występujące jeden obok drugiego powinny być oddzielone przynajmniej jednym pełnym słowem. Zamiast

...funkcja  $f(x)$ ,  
 $x \neq 0$ , jest...

...funkcja  $f(x)$ ,  $x \neq 0$ ,  
jest...

lepiej napisać

...funkcja  $f(x)$ ,  
gdzie  $x \neq 0$ , jest...

...funkcja  $f(x)$ , gdzie  
 $x \neq 0$ , jest...

a jeszcze lepiej

...funkcja  $f(x)$ ,  
gdzie  $x \neq 0$ , jest...

...funkcja  $f(x)$ ,  
gdzie  $x \neq 0$ , jest...

- Dobrze jest unikać przesadnej liczby symboli matematycznych. Zamiast

$$\begin{aligned} & \forall \epsilon > 0 \setminus ; \\ & \exists N \in \mathbb{N} : \\ & \left| a_n \right| < \epsilon \end{aligned}$$

$$\forall \epsilon > 0 \exists N \in \mathbb{N} : |a_n| < \epsilon$$

lepiej napisać

dla dowolnej liczby  $\epsilon > 0$   
istnieje taka liczba  $N \in \mathbb{N}$ ,  
że  $\left| a_n \right| < \epsilon$ ,

dla dowolnej liczby  $\epsilon > 0$  istnieje taka liczba  $N \in \mathbb{N}$ , że  $|a_n| < \epsilon$ ,

albo nawet

dla dowolnej liczby dodatniej  $\epsilon$   
istnieje taka liczba naturalna  $N$ ,  
że  $\left| a_n \right| < \epsilon$ .

dla dowolnej liczby dodatniej  $\epsilon$  istnieje taka liczba naturalna  $N$ ,  
że  $|a_n| < \epsilon$ .

- Nie należy mieszać zapisu słownego i symbolicznego w zdaniu. Zamiast

zakładając, że rozwiązanie równania jest  $> 0$

zakładając, że rozwiązanie równania jest  $> 0$

lepiej napisać

zakładając, że rozwiązanie równania jest dodatnie

zakładając, że rozwiązanie równania jest dodatnie

## Wielkości z jednostkami

Jakkolwiek w samej matematyce (teoretycznej) nieczęsto stosujemy wielkości z jednostkami, to w *zastosowaniach* matematyki jest to zjawisko powszechne. W  $\text{\LaTeX}$ u mamy kilka możliwości składu wielkości z jednostkami. Prawdopodobnie najbardziej zaawansowaną z nich jest wykorzystanie pakietu `siunitx`, który zawiera bardzo zaawansowaną maszynę pozwalającą składać najrozmaitsze jednostki. My jednak poprzestaniemy na prostszym pakiecie `units`. Zawiera on m.in. polecenie `\unit`. W najprostszej wersji – z jednym parametrem (obowiązkowym) – składa ono symbol jednostki:

`\unit{mi}` to oznaczenie mili

mi to oznaczenie mili

Zwróćmy uwagę, że polecenie to działa w trybie tekstowym; jak łatwo sprawdzić, działa ono również prawidłowo w trybie matematycznym.

Częściej przydaje się jego forma z parametrem opcjonalnym, zawierającym faktyczną wielkość

`\unit[3]{km}=\unit[300\,000]{cm}`

3 km = 300 000 cm

Zauważmy, że – dla poprawienia czytelności dużej liczby – wykorzystaliśmy wąski odstęp do oddzielenia tysięcy od setek.

Pakiet `units`, mimo swej prostoty, jest nader użyteczny. Stosując go, warto pamiętać o pewnej pułapce: w przypadku takich jednostek, jak centymetry kwadratowe czy metry sześciennie, potrzebujemy trybu matematycznego, by złożyć odpowiedni wykładnik.

`\unit[50]{cm^2}`

50 cm<sup>2</sup>

`\unit`



## Polecane lektury

Na tej książce świat  $\LaTeX$ a się nie kończy. Jest wiele lektur, które można polecić, dlatego postaraliśmy się wybrać kilka najistotniejszych.

### Dla nie czujących się jeszcze zbyt pewnie:

- *Nie za krótkie wprowadzenie do systemu  $\LaTeX 2_{\epsilon}$* , autorstwa Tobiasa Oetikera i innych, dostępne za darmo w internecie w postaci elektronicznej.
- *Praca magisterska i dyplomowa z programem  $\LaTeX$* , autorstwa Tomasza Przechlewskiego, wydana przez wydawnictwo Helion.

### Dla pewniejszych siebie:

- *$T_{\epsilon}X$ . Przewodnik użytkownika* autorstwa Donalda E. Knutha.
- *Digital Typography Using LaTeX*, autorstwa Apostolosa Syropoulosa i innych, w języku angielskim.
- *The LaTeX Companion*, autorstwa Franka Mittelbacha, Michela Goossensa i innych, w języku angielskim.

Wszystkie wymienione powyżej pozycje wyczerpująco wprowadzają w tematykę  $\LaTeX$ a. Jedno jest jednak prawdą – gdy mowa o sposobach wykorzystania pakietów, ciężko o lepsze źródło niż ich dokumentacja. Jeśli pakiet jest umieszczony w oficjalnym repozytorium, jego dokumentację można znaleźć na stronie <https://www.ctan.org/pkg/>.

**Dodatki**

# Symbole

W tym dodatku zebraliśmy najczęściej wykorzystywane symbole specjalne. O ile nie powiedziano inaczej, można je wykorzystywać jedynie w trybie matematycznym. Jeśli jednak znajdują się osoby, dla których ten zbiór okaże się niewystarczający, warto zapoznać się z najbardziej rozbudowaną z opublikowanych list, dostępną na serwerach ctan<sup>9</sup>.

## Podstawowe symbole specjalne

|                       |                                    |                             |                                  |
|-----------------------|------------------------------------|-----------------------------|----------------------------------|
| $\hat{\phantom{x}}$   | <code>\textasciicircum</code>      | $\text{\textcircled{O}}$    | <code>\textordmasculine</code>   |
| $\tilde{\phantom{x}}$ | <code>\textasciitilde</code>       | $\cdot$                     | <code>\textperiodcentered</code> |
| *                     | <code>\textasteriskcentered</code> | $\hat{\phantom{x}}$         | <code>\textasciicircum</code>    |
| $\backslash$          | <code>\textbackslash</code>        | $\text{\textquestiondown}$  | <code>\textquestiondown</code>   |
|                       | <code>\textbar</code>              | $\text{\textquotedblleft}$  | <code>\textquotedblleft</code>   |
| •                     | <code>\textbullet</code>           | $\text{\textquotedblright}$ | <code>\textquotedblright</code>  |
| —                     | <code>\textemdash</code>           | $\text{\textquotelleft}$    | <code>\textquoteleft</code>      |
| -                     | <code>\textendash</code>           | $\text{\textquoteright}$    | <code>\textquoteright</code>     |
| ¡                     | <code>\textexclamdown</code>       | $\text{\textregistered}$    | <code>\textregistered</code>     |
| >                     | <code>\textgreater</code>          | $\text{\texttrademark}$     | <code>\texttrademark</code>      |
| <                     | <code>\textless</code>             | $\text{\textvisiblespace}$  | <code>\textvisiblespace</code>   |
| <sup>a</sup>          | <code>\textordfeminine</code>      |                             |                                  |

---

<sup>9</sup> <http://tug.ctan.org/info/symbols/comprehensive/symbols-a4.pdf>

Powyższe symbole dostępne są jedynie w trybie tekstowym. Poniższe z kolei można używać zarówno w trybie tekstowym, jak i matematycznym.

|    |                         |     |                      |   |                    |
|----|-------------------------|-----|----------------------|---|--------------------|
| \$ | <code>\\$</code>        | _   | <code>\_</code>      | ‡ | <code>\ddag</code> |
| {  | <code>\{</code>         | }   | <code>\}</code>      | ¶ | <code>\¶</code>    |
| ©  | <code>\copyright</code> | ... | <code>\dots</code>   | § | <code>\§</code>    |
| †  | <code>\dag</code>       | £   | <code>\pounds</code> |   |                    |

Symbole waluty euro występujące w pakiecie eurosym

|   |                       |   |                             |   |                             |
|---|-----------------------|---|-----------------------------|---|-----------------------------|
| € | <code>\geneuro</code> | € | <code>\geneuronarrow</code> | € | <code>\officiaaleuro</code> |
|---|-----------------------|---|-----------------------------|---|-----------------------------|

## Alfabet grecki

|             |                        |            |                       |               |                          |
|-------------|------------------------|------------|-----------------------|---------------|--------------------------|
| $\alpha$    | <code>\alpha</code>    | $\beta$    | <code>\beta</code>    | $\gamma$      | <code>\gamma</code>      |
| $\delta$    | <code>\delta</code>    | $\epsilon$ | <code>\epsilon</code> | $\varepsilon$ | <code>\varepsilon</code> |
| $\zeta$     | <code>\zeta</code>     | $\eta$     | <code>\eta</code>     | $\theta$      | <code>\theta</code>      |
| $\vartheta$ | <code>\vartheta</code> | $\iota$    | <code>\iota</code>    | $\kappa$      | <code>\kappa</code>      |
| $\lambda$   | <code>\lambda</code>   | $\mu$      | <code>\mu</code>      | $\nu$         | <code>\nu</code>         |
| $\xi$       | <code>\xi</code>       | $\pi$      | <code>\pi</code>      | $\varpi$      | <code>\varpi</code>      |
| $\rho$      | <code>\rho</code>      | $\varrho$  | <code>\varrho</code>  | $\sigma$      | <code>\sigma</code>      |
| $\varsigma$ | <code>\varsigma</code> | $\tau$     | <code>\tau</code>     | $\upsilon$    | <code>\upsilon</code>    |
| $\phi$      | <code>\phi</code>      | $\varphi$  | <code>\varphi</code>  | $\chi$        | <code>\chi</code>        |
| $\psi$      | <code>\psi</code>      | $\omega$   | <code>\omega</code>   |               |                          |

|           |                      |            |                       |          |                     |
|-----------|----------------------|------------|-----------------------|----------|---------------------|
| $\Gamma$  | <code>\Gamma</code>  | $\Delta$   | <code>\Delta</code>   | $\Theta$ | <code>\Theta</code> |
| $\Lambda$ | <code>\Lambda</code> | $\Xi$      | <code>\Xi</code>      | $\Pi$    | <code>\Pi</code>    |
| $\Sigma$  | <code>\Sigma</code>  | $\Upsilon$ | <code>\Upsilon</code> | $\Phi$   | <code>\Phi</code>   |
| $\Psi$    | <code>\Psi</code>    | $\Omega$   | <code>\Omega</code>   |          |                     |

## Symbole matematyczne

### Operatory

|                    |                               |                 |                       |                  |                             |
|--------------------|-------------------------------|-----------------|-----------------------|------------------|-----------------------------|
| $\pm$              | <code>\pm</code>              | $\mp$           | <code>\mp</code>      | $\times$         | <code>\times</code>         |
| $\div$             | <code>\div</code>             | $\cdot$         | <code>\cdot</code>    | $*$              | <code>\ast</code>           |
| $\star$            | <code>\star</code>            | $\dagger$       | <code>\dagger</code>  | $\ddagger$       | <code>\ddagger</code>       |
| $\amalg$           | <code>\amalg</code>           | $\cap$          | <code>\cap</code>     | $\cup$           | <code>\cup</code>           |
| $\uplus$           | <code>\uplus</code>           | $\sqcap$        | <code>\sqcap</code>   | $\sqcup$         | <code>\sqcup</code>         |
| $\vee$             | <code>\vee</code>             | $\wedge$        | <code>\wedge</code>   | $\oplus$         | <code>\oplus</code>         |
| $\ominus$          | <code>\ominus</code>          | $\otimes$       | <code>\otimes</code>  | $\circ$          | <code>\circ</code>          |
| $\bullet$          | <code>\bullet</code>          | $\diamond$      | <code>\diamond</code> | $\triangleleft$  | <code>\triangleleft</code>  |
| $\triangleright$   | <code>\triangleright</code>   | $\triangleleft$ | <code>\unlhd</code>   | $\triangleright$ | <code>\unrhd</code>         |
| $\oslash$          | <code>\oslash</code>          | $\odot$         | <code>\odot</code>    | $\bigcirc$       | <code>\bigcirc</code>       |
| $\triangleleft$    | <code>\triangleleft</code>    | $\diamond$      | <code>\Diamond</code> | $\triangleup$    | <code>\bigtriangleup</code> |
| $\bigtriangledown$ | <code>\bigtriangledown</code> | $\square$       | <code>\Box</code>     | $\triangleright$ | <code>\triangleright</code> |
| $\setminus$        | <code>\setminus</code>        | $\wr$           | <code>\wr</code>      |                  |                             |

### Relacje

|               |                          |             |                        |               |                          |
|---------------|--------------------------|-------------|------------------------|---------------|--------------------------|
| $\leq$        | <code>\le</code>         | $\geq$      | <code>\ge</code>       | $\neq$        | <code>\neq</code>        |
| $\sim$        | <code>\sim</code>        | $\ll$       | <code>\ll</code>       | $\gg$         | <code>\gg</code>         |
| $\doteq$      | <code>\doteq</code>      | $\simeq$    | <code>\simeq</code>    | $\subset$     | <code>\subset</code>     |
| $\supset$     | <code>\supset</code>     | $\approx$   | <code>\approx</code>   | $\asymp$      | <code>\asymp</code>      |
| $\subseteq$   | <code>\subseteq</code>   | $\supseteq$ | <code>\supseteq</code> | $\cong$       | <code>\cong</code>       |
| $\subset$     | <code>\smile</code>      | $\sqsubset$ | <code>\sqsubset</code> | $\sqsupset$   | <code>\sqsupset</code>   |
| $\equiv$      | <code>\equiv</code>      | $\frown$    | <code>\frown</code>    | $\sqsupseteq$ | <code>\sqsupseteq</code> |
| $\sqsupseteq$ | <code>\sqsupseteq</code> | $\propto$   | <code>\propto</code>   | $\bowtie$     | <code>\bowtie</code>     |
| $\in$         | <code>\in</code>         | $\ni$       | <code>\ni</code>       | $\prec$       | <code>\prec</code>       |
| $\succ$       | <code>\succ</code>       | $\vdash$    | <code>\vdash</code>    | $\dashv$      | <code>\dashv</code>      |
| $\preceq$     | <code>\preceq</code>     | $\succeq$   | <code>\succeq</code>   | $\models$     | <code>\models</code>     |
| $\perp$       | <code>\perp</code>       | $\parallel$ | <code>\parallel</code> | $\mid$        | <code>\mid</code>        |

W przypadku wielu z wymienionych wyżej symboli można uzyskać odpowiadające im symbole negacji poprzez poprzedzenie nazwy polecenia literą `n`, np. `\nsim` ( $\approx$ ) lub poprzedzenie polecenia instrukcją `\not`, np. `\not\in` ( $\notin$ ).

Zauważmy, że niektóre symbole (np. uzyskiwane przez użycie polecenia `\ge`) wyglądają inaczej w dokumentach, do których dołączono pakiet polski. Wynika to z zasad składu tekstów matematycznych zaimplementowanych w wymienionym pakiecie.

### Strzałki

|                       |                                  |                       |                                  |
|-----------------------|----------------------------------|-----------------------|----------------------------------|
| $\leftarrow$          | <code>\gets</code>               | $\rightarrow$         | <code>\to</code>                 |
| $\leftarrow$          | <code>\leftarrow</code>          | $\rightarrow$         | <code>\rightarrow</code>         |
| $\Leftarrow$          | <code>\Leftarrow</code>          | $\Rightarrow$         | <code>\Rightarrow</code>         |
| $\leftrightarrow$     | <code>\leftrightarrow</code>     | $\Leftrightarrow$     | <code>\Leftrightarrow</code>     |
| $\mapsto$             | <code>\mapsto</code>             | $\longmapsto$         | <code>\longmapsto</code>         |
| $\lharpoonup$         | <code>\lharpoonup</code>         | $\rharpoonup$         | <code>\rharpoonup</code>         |
| $\lharpoondown$       | <code>\lharpoondown</code>       | $\rharpoondown$       | <code>\rharpoondown</code>       |
| $\hookrightarrow$     | <code>\hookrightarrow</code>     | $\hookrightarrow$     | <code>\hookrightarrow</code>     |
| $\Lleftarrow$         | <code>\Lleftarrow</code>         | $\Rrightarrow$        | <code>\Rrightarrow</code>        |
| $\longleftarrow$      | <code>\longleftarrow</code>      | $\longrightarrow$     | <code>\longrightarrow</code>     |
| $\longleftrightarrow$ | <code>\longleftrightarrow</code> | $\Longleftrightarrow$ | <code>\Longleftrightarrow</code> |
| $\uparrow$            | <code>\uparrow</code>            | $\downarrow$          | <code>\downarrow</code>          |
| $\Uparrow$            | <code>\Uparrow</code>            | $\Downarrow$          | <code>\Downarrow</code>          |
| $\updownarrow$        | <code>\updownarrow</code>        | $\Updownarrow$        | <code>\Updownarrow</code>        |
| $\nearrow$            | <code>\nearrow</code>            | $\nwarrow$            | <code>\nwarrow</code>            |
| $\swarrow$            | <code>\swarrow</code>            | $\searrow$            | <code>\searrow</code>            |
| $\rightleftharpoons$  | <code>\rightleftharpoons</code>  | $\leadsto$            | <code>\leadsto</code>            |

### Akcenty

|             |                        |             |                        |                |                           |
|-------------|------------------------|-------------|------------------------|----------------|---------------------------|
| $\hat{x}$   | <code>\hat{x}</code>   | $\check{x}$ | <code>\check{x}</code> | $\dot{x}$      | <code>\dot{x}</code>      |
| $\breve{x}$ | <code>\breve{x}</code> | $\acute{x}$ | <code>\acute{x}</code> | $\ddot{x}$     | <code>\ddot{x}</code>     |
| $\grave{x}$ | <code>\grave{x}</code> | $\tilde{x}$ | <code>\tilde{x}</code> | $\mathring{x}$ | <code>\mathring{x}</code> |

|                 |                            |                 |                            |                   |                              |
|-----------------|----------------------------|-----------------|----------------------------|-------------------|------------------------------|
| $\bar{x}$       | <code>\bar{x}</code>       | $\vec{x}$       | <code>\vec{x}</code>       | $\overline{x}$    | <code>\overline{x}</code>    |
| $\not{x}$       | <code>\not{x}</code>       | $\widehat{xxx}$ | <code>\widehat{xxx}</code> | $\widetilde{xxx}$ | <code>\widetilde{xxx}</code> |
| $\underline{x}$ | <code>\underline{x}</code> |                 |                            |                   |                              |

**Symbole duże**

|          |                        |           |                         |           |                        |
|----------|------------------------|-----------|-------------------------|-----------|------------------------|
| $\Sigma$ | <code>\sum</code>      | $\prod$   | <code>\prod</code>      | $\coprod$ | <code>\coprod</code>   |
| $\oplus$ | <code>\bigoplus</code> | $\otimes$ | <code>\bigotimes</code> | $\odot$   | <code>\bigodot</code>  |
| $\cup$   | <code>\bigcup</code>   | $\cap$    | <code>\bigcap</code>    | $\uplus$  | <code>\biguplus</code> |
| $\sqcup$ | <code>\bigsqcup</code> | $\vee$    | <code>\bigvee</code>    | $\wedge$  | <code>\bigwedge</code> |
| $\int$   | <code>\int</code>      | $\oint$   | <code>\oint</code>      |           |                        |

Dodatkowo, pakiet `amsmath` udostępnia symbole `\iint`, `\iiint`, `\iiiiint` oraz `\idotsint`, odpowiednio dla całek podwójnych, potrójnych, poczwórnych i większej krotności.

**Kropki**

|         |                     |         |                     |         |                     |
|---------|---------------------|---------|---------------------|---------|---------------------|
| $\dots$ | <code>\dotsc</code> | $\dots$ | <code>\dotsb</code> | $\dots$ | <code>\dotsm</code> |
| $\dots$ | <code>\dotsi</code> | $\dots$ | <code>\dotso</code> |         |                     |

Wymienione w powyższej tabeli wielokropki powinny być wykorzystywane odpowiednio (wierszami): do wycień oddzielonych przecinkami, operatorami, do zapisu iloczynu, do użycia zaraz po symbolach dużych i w innych sytuacjach.

**Nawiasy**

|           |                      |           |                      |
|-----------|----------------------|-----------|----------------------|
| $\langle$ | <code>\langle</code> | $\rangle$ | <code>\rangle</code> |
| $\lceil$  | <code>\lceil</code>  | $\rceil$  | <code>\rceil</code>  |
| $\lfloor$ | <code>\lfloor</code> | $\rfloor$ | <code>\rfloor</code> |
| $\{$      | <code>\{</code>      | $\}$      | <code>\}</code>      |

## Inne symbole

|              |                         |                |                           |                   |                           |
|--------------|-------------------------|----------------|---------------------------|-------------------|---------------------------|
| $\infty$     | <code>\infty</code>     | $\triangle$    | <code>\triangle</code>    | $\sphericalangle$ | <code>\angle</code>       |
| $\aleph$     | <code>\aleph</code>     | $\hbar$        | <code>\hbar</code>        | $\imath$          | <code>\imath</code>       |
| $\jmath$     | <code>\jmath</code>     | $\ell$         | <code>\ell</code>         | $\wp$             | <code>\wp</code>          |
| $\Re$        | <code>\Re</code>        | $\Im$          | <code>\Im</code>          | $\mho$            | <code>\mho</code>         |
| $\prime$     | <code>\prime</code>     | $\emptyset$    | <code>\emptyset</code>    | $\nabla$          | <code>\nabla</code>       |
| $\surd$      | <code>\surd</code>      | $\partial$     | <code>\partial</code>     | $\top$            | <code>\top</code>         |
| $\perp$      | <code>\bot</code>       | $\vdash$       | <code>\vdash</code>       | $\dashv$          | <code>\dashv</code>       |
| $\forall$    | <code>\forall</code>    | $\exists$      | <code>\exists</code>      | $\neg$            | <code>\neg</code>         |
| $\flat$      | <code>\flat</code>      | $\natural$     | <code>\natural</code>     | $\sharp$          | <code>\sharp</code>       |
| $\backslash$ | <code>\backslash</code> | $\Box$         | <code>\Box</code>         | $\diamond$        | <code>\Diamond</code>     |
| $\clubsuit$  | <code>\clubsuit</code>  | $\diamondsuit$ | <code>\diamondsuit</code> | $\heartsuit$      | <code>\heartsuit</code>   |
| $\spadesuit$ | <code>\spadesuit</code> | $\Join$        | <code>\Join</code>        | $\blacksquare$    | <code>\blacksquare</code> |
| $\star$      | <code>\bigstar</code>   |                |                           |                   |                           |



# Pakiety

## Pakiet bamboleo

Poniżej przedstawiamy kod źródłowy pakietu o nazwie bamboleo. Definiuje on polecenie `\bamboleo`, które z każdym wywołaniem wyświetla losowy wers tekstu piosenki *Gipsy Kings* o tym samym tytule. Aby z niego skorzystać, umieść poniższy kod w pliku `bamboleo.sty`, zapisz go w katalogu ze swoim dokumentem, a w nim samym zadeklaruj użycie pakietu `bamboleo`.  
Dobrej zabawy!

`\bamboleo`

```
\NeedsTeXFormat{LaTeX2e}[1994/06/01]
\ProvidesPackage{bamboleo}

\RequirePackage{pgf}

\pgfmathsetseed{\number\pdfandomseed}

\pgfmathdeclarerandomlist{bamboleo}{
 {Ese amor llega asi de esta manera}
 {No tiene la culpa}
 {Caballo de danza vana}
 {Porque es muy despreciado por eso}
 {No te perdona llorar}
 {Ese amor llega asi esta manera}
 {Amor de compra why venta}
 {Amor de en el pasado}
 {Bem, bem bem bem, bem bem bem}
```

```
{Bamboleo bambolea}
{Porque mi vida yo la prefiero vivir asi}
{No tienes perden de Dios}
{Tu eres mi vida la fortuna del destino}
{En el destino del desamparado}
{Lo mismo ya que ayer}
{Lo mismo soy yo}
{No te encuentro l'abandon}
{Eres imposible no te encuentro de verdad}
{Por eso un dia no encuentro si de nada}
{Lo mismo ya que ayer}
{Lo pienso en ti}
}

\def\bamboleo{\pgfmathrandomitem\z{bamboleo}\z}
\endinput
```

## Pakiet latex-ksiazka-kucharska

Pakiet `latex-ksiazka-kucharska` pozwala na składanie przykładów umieszczonych w tej książce w taki sposób, aby można je było zaprezentować w skali 1:1. Więcej na temat tego rozszerzenia napisano na s. 41.

```
\NeedsTeXFormat{LaTeX2e}[1994/06/01]
\ProvidesPackage{latex-ksiazka-kucharska}

\RequirePackage[paperwidth=11.5cm,
 paperheight=15cm,
 margin=1.5cm]{geometry}

\endinput
```

## Skorowidz poleceń i nazw

\(, 186  
\), 186  
\;, 193  
\!, 79, 106, 129

### A

\addbibresource, 102  
\aleph, 195  
align, 197, 201  
align\*, 198  
aligned, 196, 197  
ams, 22  
amsmath, 189, 201, 203, 215  
amsrefs, 101  
amssymb, 189  
amsthm, 167, 169, 180, 183, 185  
\approx, 193  
article, 19, 21, 22, 32, 34, 36, 42, 56

### B

balance, 88  
bamboleo, 217  
\bar, 195  
\baselineskip, 124  
beamer, 21  
\beth, 195  
biblatex, 102, 103  
\Biggl, 192  
\biggl, 192

\Biggr, 192  
\biggr, 192  
\Bigl, 192  
\bigl, 192  
\Bigr, 192  
\bigr, 192  
\bigskip, 78  
\binom, 191  
bmatrix, 201  
book, 21, 22, 32, 34, 36  
booktabs, 105, 106, 114  
\bottomrule, 106

### C

\caption, 109, 113  
\captiondelim, 113  
cases, 195  
ccaption, 113  
\cdot, 193  
\cdots, 203  
center, 79, 106, 109  
\centering, 109  
\chapter, 54  
\chapter\*, 58  
\cite, 102  
\cleardoublepage, 137  
\clearpage, 137  
\cmidrule, 114  
\colon, 193

`color`, 118, 120`\colorbox`, 118`\cos`, 194`\cup`, 205**D**`\dbinom`, 191`\DeclareMathOperator`, 194`\DeclareMathOperator*`, 194`\definecolor`, 119`\dfrac`, 190, 204`document`, 20, 24`\documentclass`, 18, 23`\dot`, 195`\dots`, 202, 203`\dywiz`, 67**E**`\emph`, 65`enumerate`, 42, 44, 51`enumerate*`, 51`enumitem`, 42, 45, 46, 49`\eqref`, 188, 197`equation`, 188, 197`equation*`, 188`eurosym`, 212`\exists`, 193**F**`fancyhdr`, 151, 154`\fancyhf`, 154`figure`, 132`floatrow`, 130`flushend`, 88`flushleft`, 79`flushright`, 79, 81`fontenc`, 29, 36`\footnote`, 69`\forall`, 193`\frac`, 18, 188**G**`gather`, 200`gather*`, 200`geometry`, 22, 139, 141`\geq`, 193`\gimel`, 195`graphicx`, 126**H**`\hat`, 195`\hdotsfor`, 202`hyperref`, 22, 70**I**`\i`, 71`\imath`, 195`\in`, 193`\indent`, 74`\index`, 96`\indexname`, 96`inputenc`, 23, 28`\int`, 189, 204`\item`, 43, 47`itemize`, 42--44, 46, 51`itemize*`, 51**J**`\j`, 71`\jmath`, 195**L**`\label`, 93, 188`landscape`, 166`latex-ksiazka-kucharska`, 41,

218

`\ldots`, 203  
`\left`, 191  
`\legend`, 113  
`\leq`, 193  
`letter`, 21, 32  
`\limits`, 189  
`lipsum`, 22, 23  
`listings`, 22  
`\listoffigures`, 99  
`\listoftables`, 99  
`\listoftheorems`, 178, 180  
`\listtheoremname`, 178  
`\log`, 194  
`\lvert`, 192

**M**

`makeidx`, 95  
`\makeindex`, 95  
`\maketitle`, 64  
`\markboth`, 149  
`\markright`, 149  
`\mathbb`, 195  
`\mathbf`, 195  
`\mathcal`, 195  
`\mathfrak`, 195  
`matrix`, 201  
`mdframed`, 91  
`\medskip`, 78  
`\medspace`, 203  
`\midrule`, 106  
`minimal`, 21  
`minipage`, 81  
`multicol`, 89  
`\multicolumn`, 114  
`multirow`, 115  
`mw*`, 78, 88, 145, 149, 161  
`mwart`, 36, 113, 139, 145, 146, 149  
`mwbk`, 36, 54, 64, 113, 139, 145, 161  
`mwcls`, 36, 51, 58  
`mwrep`, 36, 145, 146

**N**

`\negmedspace`, 203  
`\negthickspace`, 203  
`\negthinspace`, 203  
`\newcounter`, 171  
`\newpage`, 136  
`\newtheorem`, 169  
`\newtheorem*`, 169  
`\noindent`, 74  
`\nolimits`, 189  
`\nopagebreak`, 138  
`\notag`, 199  
`ntheorem`, 167

**O**

`\onecolumn`, 88  
`\operatorname`, 194

**P**

`\pagebreak`, 136  
`\pagecolor`, 119  
`\pageref`, 93  
`\pagestyle`, 145  
`\parbox`, 81  
`parskip`, 75  
`\part`, 62  
`pdfscape`, 166  
`\pi`, 193  
`pmatrix`, 201  
`polski`, 35, 36, 67, 96, 214  
`\ppauza`, 67  
`\printbibliography`, 102  
`\printindex`, 95  
`proof`, 180

**Q**

\qedhere, 180  
 \qedsymbol, 181  
 \qqquad, 203, 204  
 \quad, 129, 187, 203, 204  
 quotation, 84  
 quote, 84

**R**

\ref, 93, 188  
 \renewcommand, 96, 178, 181  
 report, 21, 32, 34, 36  
 \right, 191  
 \rvert, 192

**S**

\section, 56  
 \section\*, 58  
 \setenumerate, 50  
 \setitemize, 49  
 \sin, 194  
 siunitx, 208  
 slides, 21  
 smallmatrix, 201  
 \smallskip, 78  
 \sqrt, 193  
 subcaption, 130  
 subfig, 129, 130  
 \subfloat, 129  
 \subref, 129  
 \subsection, 56  
 \sup, 194  
 \swapnumbers, 185

**T**

table, 109, 110  
 \tableofcontents, 99  
 tabular, 106

\tbinom, 191  
 \text, 188, 205  
 \textasteriskcentered, 46  
 \textbackslash, 25  
 \textbf, 66  
 \textcolor, 118  
 \textheight, 124  
 \textit, 66  
 \textup, 204  
 \textwidth, 124  
 \tfrac, 190  
 \tg, 194  
 \theoremstyle, 183  
 \thepage, 154  
 \thickspace, 203  
 \thinspace, 203  
 \thispagestyle, 146  
 thmtools, 178  
 tikz, 121  
 tikzposter, 21  
 \to, 193  
 tocloft, 99  
 \toprule, 106  
 \twocolumn, 88

**U**

\unit, 208  
 units, 208  
 url, 70  
 \urldef, 70  
 \usepackage, 23

**V**

\vec, 195  
 verbatim, 16  
 verse, 84  
 \vfill, 78  
 vmatrix, 201

`\vspace,78`

`\vspace*,78`

**X**

`xcolor,118--120`

`xxcolor,118`