

Karol Selwat

Wprowadzenie do systemu L^AT_EX

Skrypt dla studentów
Instytutu Matematycznego
Uniwersytetu Wrocławskiego

Instytut Matematyczny
Uniwersytetu Wrocławskiego

Spis treści

Od autora	4
1 Wiadomości wstępne	5
1.1 Rozkazy i otoczenia	5
1.2 Struktura dokumentu \LaTeX -owego. Pakiety dodatkowe	6
1.3 Style stron	7
2 Podstawowe elementy składu tekstu	8
2.1 Wiersze, akapity, strony	8
2.2 Dodatkowe odstępy	8
2.3 Znaki specjalne	9
2.4 Spacja nierozdzielająca	9
2.5 Niektóre znaki przestankowe	10
2.6 Komentarze	10
3 Style i wielkości pisma	11
3.1 Wyróżnienia	11
3.2 Symulacja maszynopisu	11
4 Logiczny podział dokumentu	13
4.1 Strona tytułowa	13
4.2 Spis treści	14
4.3 Bibliografia	14
Nasz spis publikacji	15
4.4 Odwołania czyli odsyłacze	15
4.5 Notki	16
5 Przykłady formatowania tekstu	17
5.1 Wyrównywanie poziome	17
5.2 Wykazy czyli listy	17
5.2.1 Wykaz nieuporządkowany	17
5.2.2 Wykaz uporządkowany	18
5.2.3 Wykaz definicyjny	19
5.3 Tabele	19
6 Wstawianie plików graficznych	21
7 Wyrażenia matematyczne	23
7.1 Indeksy, wykładniki i pochodne	23
7.2 Odstępy	24

7.3	Ułamki	24
7.4	Pierwiastki	24
7.5	Sumy, iloczyny i całki	24
7.6	Funkcje	25
7.7	Skalowalne nawiasy	25
7.8	Macierze	26
7.9	Zwykły tekst we wzorach	26
7.10	Podkreślenia i inne ozdobniki	27
7.11	Matematyczne kroje pisma	27
7.12	Akcenty	27
7.13	Litery greckie	28
7.14	Niektóre symbole matematyczne	28
7.15	Wzory numerowane	29
7.16	Wzory wielowierszowe	30
7.17	Struktury tekstowe	30
7.18	Parę słów na zakończenie	31
8	Obiekty ruchome	32
9	Praca z dużym dokumentem	34
10	Modyfikowanie \LaTeX-a	35
11	Często Zadawane Pytania	36
	Bibliografia	38
	Indeks	39

Od autora

Skrypt ten powstał na podstawie zajęć prowadzonych przeze mnie w laboratoriach komputerowych Instytutu Matematycznego Uniwersytetu Wrocławskiego. Dlatego też przytaczane przykłady są przystosowane do tamtejszych instalacji L^AT_EX-a opartych na dystrybucji **Web2c** w systemach linuksowych. Używanie L^AT_EX-a w innych instalacjach może wymagać pewnych, najczęściej bardzo drobnych, modyfikacji kodów przykładów.

Karol Selwat

Wersja 1.3
Copyright © 2003 – 2007 by Karol Selwat
Wszystkie prawa zastrzeżone.

1 Wiadomości wstępne

\LaTeX (czyt. „la-tech” lub „lej-tech”) jest systemem poleceń, który ułatwia skład publikacji elektronicznych przy pomocy programu \TeX . Twórcą programu \TeX jest Donald E. Knuth [4], a pierwszej wersji \LaTeX -a – Leslie Lamport [5]. Obecna wersja \LaTeX -a to $\LaTeX 2_{\epsilon}$ i tym właśnie pakietem się zajmujemy. Więcej o filozofii i historii systemu $\LaTeX 2_{\epsilon}$ można przeczytać w rozdziale 1 pracy [1].

Plik źródłowy \LaTeX -a jest zwykłym plikiem tekstowym i można go edytować dowolnym edytorem tekstowym np. edytorem **emacs** w systemie Linux lub edytorem **Notatnik** w systemie MS Windows. Pliki źródłowe mają rozszerzenie (przyrostek) **.tex**.

1.1 Rozkazy i otoczenia

Rozkazy \LaTeX -a rozpoczynają się znakiem „backslash” (\backslash), po którym następuje nazwa rozkazu. Nazwa rozkazu składa się z ciągu liter (np. $\backslash\text{LaTeX}$) lub dokładnie jednego znaku nie będącego literą (np. $!$). Zazwyczaj rozkazy posiadają jeszcze argumenty, bądź obowiązkowe, bądź opcjonalne. Argumenty obowiązkowe umieszczamy w nawiasach klamrowych (każdy argument w osobnej parze nawiasów), zaś opcjonalne – w nawiasach prostokątnych (każdy argument oddzielamy przecinkiem). Zatem składnia typowego polecenia jest następująca:

$$\backslash\textit{nazwa} [\textit{opcja1}, \textit{opcja2}, \dots] \{ \textit{argument obowiązkowy} \}.$$

Szczególnymi rozkazami są $\backslash\textit{begin}$ oraz $\backslash\textit{end}$. Służą one do wyznaczania otoczeń (środowisk). Otoczenia pozwalają np. stosować określone formatowanie do całych fragmentów tekstu. Otoczenie rozpoczyna się od $\backslash\textit{begin}\{\textit{nazwa}\}$, a kończy się na $\backslash\textit{end}\{\textit{nazwa}\}$, gdzie *nazwa* jest nazwą otoczenia. Pomędzy powyższymi rozkazami umieszczamy formatowany tekst. Dla przykładu: środowisko **center** umieszcza fragment tekstu na środku strony (w poziomie) – przykład powyżej.

\LaTeX jest systemem wrażliwym na wielkość liter w nazwach rozkazów i otoczeń. Na przykład: rozkaz $\backslash\text{LaTeX}$ jest napisany poprawnie (wygeneruje logo systemu); napisy typu $\backslash\textit{latex}$, $\backslash\text{Latex}$, $\backslash\textit{laTeX}$ spowodują błędy przy kompilacji.

1.2 Struktura dokumentu \LaTeX -owego. Pakiety dodatkowe

Plik źródłowy składa się z trzech części: prologu, preambuły oraz części właściwej. Na naszym poziomie zaawansowania będziemy pomijać prolog.

W preambule deklarujemy klasę (typ) dokumentu, ustawienia globalne (dotyczące całości dokumentu) i używane pakiety dodatkowe. Standardowe klasy dokumentów to: `article`, `report`, `book`, `letter` oraz `slides`. Ich szczegółowe omówienie można znaleźć np. w rozdziale 8 książki [8]. My będziemy używali klasy `article`, służącej głównie do tworzenia niedużych artykułów. Klasę dokumentu deklarujemy rozkazem `\documentclass`, który należy umieścić na samym początku preambuły. W naszych dokumentach będziemy stosowali czcionkę wielkości 12pt oraz papier formatu A4, czemu damy wyraz w opcjach rozkazu `\documentclass`. Zatem pierwsza linia naszej preambuły przedstawia się następująco:

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

Możliwości standardowego \LaTeX -a poszerzane są za pomocą tzw. pakietów dodatkowych. Umożliwiają one m.in. dołączanie plików graficznych, stosowanie języków narodowych bądź też używanie różnych standardów kodowania znaków. Chęć użycia dodatkowego pakietu deklarujemy w preambule rozkazem `\usepackage`. My zadeklarujemy parę pakietów, aby móc wygodnie (i ładnie) pisać w języku polskim. Nasza standardowa preambuła wyglądać będzie następująco:

```
\documentclass[12pt,a4paper]{article}
\usepackage[polish]{babel}
\usepackage[T1]{fontenc}
\usepackage{polski}
\usepackage[latin2]{inputenc}
```

Pakiety `babel` z opcją `polish` oraz `polski` określają język publikacji i m.in. wprowadzają polskie nazewnictwo części dokumentu. Pakiet `fontenc` ustala układ znaków (u nas: nowy system T1). Pakiet `inputenc` podaje sposób kodowania znaków, użyty w pliku źródłowym (`latin2` to standardowy wg. Polskiej Normy system ISO-8859-2, używany w systemach linuxowych; w systemach MS Windows standardem kodowania jest `cp1250`). Dzięki temu pakietowi polskie litery możemy wprowadzać z klawiszem Alt (np. `ą=Alt+a`).

Część właściwa dokumentu \LaTeX -owego umieszczona jest w ramach otoczenia `document`, tzn. zaczyna się od rozkazu `\begin{document}`, a kończy się na rozkazie `\end{document}`. Pomiędzy tymi rozkazami umieszczamy całą

treść dokumentu, łącznie z (lokalnymi) rozkazami formatującymi ją. Polecenia `\begin{document}` oraz `\end{document}` mogą wystąpić w pliku źródłowym co najwyżej jednokrotnie (najczęściej występują dokładnie jeden raz).

1.3 Style stron

Standardowa strona drukowanego dokumentu dzieli się na trzy części: tekst właściwy, nagłówek (paginę górną) ponad tym tekstem oraz stopkę (paginę dolną) poniżej tekstu głównego. W systemie *L^AT_EX* zawartość pagin jest określana za pomocą rozkazu `\pagestyle{styl}`, gdzie argument *styl* może przyjmować cztery wartości: `plain` (wartość domyślna), `empty`, `headings` lub `myheadings`. W stylu strony `plain` nagłówek jest pusty, a stopka zawiera wycentrowany numer bieżącej strony. Podanie argumentu `empty` spowoduje, że obie paginy będą puste. W stylu `headings` stopka jest pusta, natomiast nagłówek zawiera tzw. żywą paginę, zazwyczaj numer strony oraz tytuł bieżącego rozdziału. Jeśli chcemy samodzielnie ustalić treść żywej paginy, należy użyć stylu strony `myheadings`. Zainteresowanych opisem przygotowania własnej żywej paginy odsyłamy np. na strony 96–97 książki [8]. Polecenie `\pagestyle` powinno być umieszczone w preambule dokumentu.

Do zmiany stylu bieżącej strony służy rozkaz `\thispagestyle{styl}`, gdzie wartości argumentu *styl* są takie same jak w przypadku `\pagestyle`.

Przy definiowaniu żywej paginy dla tego skryptu autor posłużył się pakietem `fancyhdr` autorstwa Pieta van Oostruma. Parę słów na temat użycia tego pakietu można przeczytać w rozdziale 4.4 opracowania [1].

2 Podstawowe elementy składu tekstu

Formatowanie dokumentów przy pomocy \LaTeX -a nie jest formatowaniem typu WYSIWYG („Co widzisz, to dostaniesz”, jak np. w MS Word), lecz formatowaniem logicznym. Wprowadzamy tekst, podajemy rozkazy go formatujące, np. mówimy że w danym miejscu ma się rozpocząć nowy rozdział i nie martwimy się o wygląd wyjściowego dokumentu. Dzięki temu, że system \LaTeX ma wbudowane pewne mechanizmy (zasady) typograficzne, nasz tekst zostanie złożony na poziomie niemalże profesjonalnym.

2.1 Wiersze, akapity, strony

\LaTeX rozmieszcza podany w pliku źródłowym tekst w taki sposób, że kiedy zapełni nim cały wiersz, automatycznie przechodzi do zapełniania wiersza następnego (często dzieląc ostatni wyraz w wierszu zgodnie z regułami dzielenia wyrazów). Dlatego też nie musimy się martwić, aby jakieś wyrażenie nie wystawało poza prawy margines tekstu (przynajmniej w większości wypadków). Ponadto kolejno występujące po sobie znaki odstępu są przez \LaTeX -a traktowane jako jeden znak. Zatem nie musimy się martwić o ilość spacji międzywyrazowych. Rozmieszczany tekst jest domyślnie wyrównywany do lewego marginesu.

Jeśli chcemy wymusić rozpoczęcie nowego akapitu, w pliku źródłowym umieszczamy pusty wiersz (lub używamy rozkazu `\par`). Na wydruku nowe akapity rozpoczynają się wcięciem akapitowym. Jeśli chcielibyśmy się go pozbyć, to możemy użyć rozkazu `\noindent`. Do wymuszenia złamania wiersza (ale wciąż w ramach tego samego akapitu) służy rozkaz `\\` (lub równoważnie `\newline`).

Po zapełnieniu tekstem całej strony \LaTeX przechodzi automatycznie do zapełniania strony następnej. Jeśli chcemy wymusić wcześniejsze przejście do nowej strony, to należy użyć polecenia `\newpage`.

2.2 Dodatkowe odstępy

Do umieszczania dodatkowych odstępow pionowych między kolejnymi akapitami stosujemy polecenie `\vspace`. Jako argument obowiązkowy podajemy wielkość odstępu, np. `1cm`, `0.2mm`, `3ex` (`1ex` to wielkość równa wysokości litery x w bieżącym kroju czcionki). Pionowy odstęp między kolejnymi wierszami tego samego akapitu uzyskamy dodając do rozkazu złamania wiersza argument opcjonalny, np. `\\[0.5cm]`.

Do uzyskania odstępow poziomych służy polecenie `\hspace`, analogiczne do `\vspace`.

Jeśli dodatkowe odstępy mają wystąpić od początku nowej strony (odstęp pionowy) lub od początku nowego wiersza (odstęp poziomy), to należy użyć postaci „gwiazdkowych” powyższych poleceń, tzn. `\vspace*` i `\hspace*` odpowiednio.

Czasami zachodzi potrzeba równomiernego rozmieszczenia w jednym wierszu paru grup wyrazów, przy czym pierwsza grupa powinna być wyrównana do lewego marginesu, a ostatnia grupa – do marginesu prawego. Taki efekt osiągniemy dzięki rozkazowi `\hfill` (skrót od `\hspace{\fill}`), który wstawia odstęp poziomy o maksymalnej możliwej w wierszu szerokości (tzw. „rozporę poziomą”). Np. kod `Początek wiersza \hfill koniec wiersza` skompiluje się jako

Początek wiersza

koniec wiersza

Maksymalnie wysoki odstęp pionowy (tzw. „rozporę pionową”) uzyskujemy za pomocą polecenia `\vfill`.

2.3 Znaki specjalne

Parę znaków ma dla \LaTeX -a specjalne znaczenie. Te znaki to:

$$\backslash \$ \& \% \# _ \{ \} \sim \wedge$$

Dla przykładu: w nawiasach klamrowych umieszcza się argumenty obowiązkowe polecenia. Ponadto nawiasy klamrowe służą do grupowania: znak `{` otwiera grupę wyrazów (wyrażeń), a znak `}` ją zamyka. Czasami para `{}` służy do wyraźnego zaznaczenia końca rozkazu po to, aby nie zlewał się on z dalszym tekstem (np. `\LaTeX{}`).

Znaki specjalne umieszczone wprost w pliku źródłowym nie pojawią się na wydruku. Można je wstawić do dokumentu poprzedzając je znakiem `\` (ang. *backslash*): zapis `\$ \& \% \# _ \{ \} \sim \wedge` w źródle daje na wydruku `$ & % # _ { } ~ ^`. Aby wydrukować znak `\` stosujemy rozkaz `\backslash$`.

2.4 Spacja nierozdzielająca

Regułą języka polskiego jest niepozostawianie jednoliterowych spójników oraz przyimków na końcu linii. Aby taki spójnik lub przyimek powiązać z następnym wyrazem stosujemy tzw. nierozdzielającą (twardą) spację, otrzymaną za pomocą symbolu `~`. Na przykład pisząc `z~głową` nakazujemy \LaTeX -owi by nie złamał wiersza między wyrazami `z` oraz `głową`, przy jednoczesnym zachowaniu odstępu między nimi.

2.5 Niektóre znaki przestankowe

Słowa można ujmować w cudzysłowy wedle zasad amerykańskich lub polskich. Obie reguły definiują cudzysłów zamykający jako ” (’). Amerykański cudzysłów otwierający to “ (‘), zaś polski to „ (,).

Przy składzie dokumentów używamy trzech różnych typów myślników, stosowanych do różnych celów.

Pauza (---) używana jest do wyróżnienia wtrąconej części zdania np. „Używałem — albo raczej próbowałem używać — matematyki do rozwiązywania różnych zagadnień”.

Półpauzę (--) stosujemy np. do zapisu „w latach 1939–1945”, „twierdzenie Naviera–Stokesa” itp.

Dywiz (-) pojawia się przy podziale i przenoszeniu wyrazów do następnego wiersza.

Wielokropek ... otrzymamy poleceniem `\ldots`.

2.6 Komentarze

Często przydatne jest umieszczenie w pliku źródłowym własnego komentarza, który nie będzie widoczny na wydruku. Do tego celu służy znak %. Umieszczenie go w pliku źródłowym powoduje, że \LaTeX podczas kompilacji ignoruje wszystkie znaki począwszy od % aż do końca linii, w której on się znajduje. Na przykład kod

```
„II wojna światowa rozpoczęła się w roku 1939, % sprawdzić  
a skończyła się w roku 1945.” % także sprawdzić
```

kompiluje się jako

```
„II wojna światowa rozpoczęła się w roku 1939, a skończyła się w roku 1945.”
```

3 Style i wielkości pisma

W systemie \LaTeX dostępnych jest kilka krojów (stylów) pisma:

```
\textrm – Roman,
\textbf – Bold Face,
\texttt – Typewriter,
\textit – Italic,
\textsl – Slanted,
\textsc – SMALL CAPS,
\textsf – Sans Serif.
```

Rozkazy udostępniające kroje pisma mają składnię standardową, na przykład `\textbf{...}` (jako argument podajemy oczywiście fragment tekstu, który ma posiadać określony krój czcionki). Natomiast polecenia zmieniające wielkość pisma są rozkazami bezargumentowymi i stosujemy je do już pogrupowanych wyrazów, np. `{\tiny Fragment pisany pismem drobnutkim}`.

W \LaTeX -u istnieją następujące wielkości pisma:

```
\tiny – Drobnutkie
\scriptsize – Bardzo ma"le
\footnotesize – Dla notek
\small – Ma"le
\normalsize – Normalne
\large – Du"re
\Large – Wi"eksze
\LARGE – Wielkie
\huge – Olbrzymie
\Huge – Gigantyczne
```

3.1 Wyróżnienia

Wyróżnień fragmentów tekstu dokonujemy rozkazem `\emph`, na przykład zapis `\textrm{Tekst \emph{wyróżniony}}` daje po przetworzeniu Tekst *wyróżniony*.

3.2 Symulacja maszynopisu

Aby zasymulować pismo maszynowe (w ramach jednego wiersza), używamy rozkazu `\verb+tekst+`. Znak `+` jest tutaj ogranicznikiem argumentu `tekst`, który zostanie wydrukowany tak, jakby był pisany na maszynie do pisania.

Zamiast znaku `+` jako ogranicznika można użyć innego znaku, za wyjątkiem litery, spacji oraz gwiazdki.

Do składu pismem maszynowym większych fragmentów tekstu (np. kodów programów) stosujemy otoczenie `verbatim`. Zachowuje ono układ tekstu, odstępy i zmiany wiersza.

Rozkaz `\verb` oraz środowisko `verbatim` nie mogą być używane jako argumenty innych poleceń. Są natomiast często wykorzystywane do składania przykładów kodu \LaTeX -a, gdyż wewnątrz rozkazu `\verb` oraz otoczenia `verbatim` nie są wykonywane żadne rozkazy tego systemu.

4 Logiczny podział dokumentu

Publikacje, zwłaszcza te większe, są zazwyczaj dzielone na mniejsze części logiczne: rozdziały, podrozdziały, paragrafy itd. *L^AT_EX* bardzo ułatwia pracę z takim podziałem logicznym m.in. automatycznie formatuje nagłówki rozdziałów itp., a ponadto w łatwy sposób tworzy odpowiednio sformatowany spis treści.

W klasie `article` mamy następujące rozkazy podziału dokumentu na części: `\part` (część), `\section` (sekcja, czasami utożsamiana z rozdziałem), `\subsection` (podsekcja, podrozdział), `\subsubsection`, `\paragraph` (paragraf) oraz `\subparagraph`. Domyślnie nagłówki pierwszych czterech części zostaną opatrzone odpowiednim numerem i pojawią się w spisie treści. Składnia w.w. rozkazów jest następująca:

$$\backslash\text{rozkaz} [\textit{wpis}] \{\textit{nagłówek}\},$$

gdzie *wpis* pojawi się w spisie treści, a *nagłówek* pojawi się (odpowiednio sformatowany) na wydruku w miejscu użycia rozkazu. Jeśli napisy *wpis* oraz *nagłówek* mają być identyczne, to można pominąć argument opcjonalny. Na przykład efekt użycia rozkazu `\section` jest widoczny u góry tej strony (i oczywiście w spisie treści).

Polecenia podziału dokumentu mają swoje odpowiedniki „z gwiazdką” (np. `\section*`), które wygenerują nagłówki nieopatrzone numerem i niepojawiające się w spisie treści. Aby je tam dodać posługujemy się rozkazem `\addcontentsline` o następującej postaci:

$$\backslash\text{addcontentsline}\{\text{toc}\}\{\textit{rozkaz}\}\{\textit{wpis}\}.$$

Dla przykładu:

```
\section*{Sekcja}
\addcontentsline{toc}{section}{Sekcja}
```

4.1 Strona tytułowa

Strona tytułowa jest generowana przez bezargumentowy rozkaz `\maketitle`, umieszczany na początku części właściwej dokumentu. Dane potrzebne do utworzenia strony tytułowej deklarujemy w preambule poleceniami `\author`, `\title` oraz (opcjonalnie) `\date`. Jako ich argumenty obowiązkowe podajemy odpowiednio autora, tytuł oraz datę artykułu. Jeśli jest paru autorów to ich nazwiska oddzielamy rozkazem `\and`. Bieżącą datę wygeneruje rozkaz `\today`.

Jeśli chcemy utworzyć własną, niestandardową stronę tytułową, to należy ją umieścić w otoczeniu `titlepage`.

W klasie `article` część tytułowa nie będzie stanowić oddzielnej strony. Aby otrzymać osobną stronę tytułową należy zadeklarować opcję `titlepage` rozkazu `\documentclass`.

4.2 Spis treści

Spis treści dokumentu tworzymy poleceniem `\tableofcontents`. Standardowo zostaną w nim umieszczone wpisy (a w razie ich braku – nagłówki) poleceń `\part`, `\section`, `\subsection` oraz `\subsubsection`. Ponadto w spisie zawartości pojawią się ewentualne wpisy dokonane przy pomocy rozkazu `\addcontentsline`. Wszystkie w.w. wpisy będą odpowiednio sformatowane i opatrzone numerami odpowiednich stron.

Spis treści umieszczony na początku wymaga trzykrotnej kompilacji dokumentu.

4.3 Bibliografia

Przy tworzeniu spisu literatury stosujemy środowisko `thebibliography`. Posiada ono jeden dodatkowy argument obowiązkowy `tekst`, którego znaczenie wyjaśnimy po podaniu przykładu spisu publikacji. Zatem otoczenie to rozpoczynamy od `\begin{thebibliography}{tekst}`. Niestety, ten rozkaz nie umieści automatycznie nagłówka „Bibliografia” w spisie treści. Zatem musimy zastosować polecenie `\addcontentsline`, najlepiej w następnej linii za powyższym rozkazem.

Każdy pozycja w spisie bibliograficznym rozpoczyna się od polecenia `\bibitem[etykieta]{klucz}`, gdzie `etykieta` pojawi się jako etykieta danej pozycji w bibliografii, a `klucz` będziemy stosowali w tekście przy odwołaniach do tej pozycji. Każdy klucz może składać się z liter (najlepiej z wyłączeniem polskich), cyfr oraz znaków przestankowych z wyjątkiem przecinka. Jeżeli w naszym spisie nie zastosujemy etykiet, to pozycje bibliograficzne zostaną ponumerowane kolejnymi liczbami naturalnymi.

W ramach tekstu do pozycji bibliograficznych odwołujemy się za pomocą rozkazu `\cite[adnotacja]{lista-kluczy}`, gdzie `adnotacja` jest opcjonalną adnotacją, natomiast `lista-kluczy` składa się z conajmniej jednego klucza opisu bibliograficznego (każdy klucz w liście oddzielamy przecinkiem).

Jako przykład podajemy następujący spis publikacji:

```
\begin{thebibliography}{Knuth}
\addcontentsline{toc}{subsubsection}{Nasz spis publikacji}
\bibitem[Diller]{wiersz}
Antoni Diller, \emph{\LaTeX} wiersz po wierszu}, ...
\bibitem[Knuth]{texbook}
Donald E. Knuth, \emph{The \TeX book}, ...
\end{thebibliography}
```

Spis ten po przetworzeniu da następujący rezultat (popatrz również na spis treści skryptu):

Literatura

[Diller] Antoni Diller, *LaTeX wiersz po wierszu*, ...

[Knuth] Donald E. Knuth, *The TeXbook*, ...

Do pozycji bibliograficznych tego spisu możemy się odwoływać np. za pomocą `\cite[str. 1]{wiersz}` lub `\cite{wiersz,texbook}`. Na wydruku otrzymamy odpowiednio: [Diller, str. 1] oraz [Diller, Knuth].

Wróćmy teraz do opisu argumentu *tekst* otoczenia `thebibliography`. W naszym przykładzie jego wartość wynosi `Knuth`. Jest to najdłuższa na wydruku etykieta spisu bibliograficznego. W przypadku gdy nie używamy etykiet argument *tekst* powinien być liczbą o tej samej ilości cyfr co liczba wpisów w bibliografii.

4.4 Odwołania czyli odsyłacze

W \LaTeX -u mechanizm automatyzacji odwołań działa nie tylko w odniesieniu do pozycji bibliograficznych. Możemy odwoływać się do wielu obiektów np. do logicznych części dokumentu, stron, równań matematycznych, twierdzeń, rysunków, tabel itp. Wystarczy w pliku źródłowym oznaczyć dany obiekt odpowiednim kluczem (etykietą) i następnie odwołać się do tego klucza w innym miejscu. Na wydruku w miejscu odwołania pojawi się numer porządkowy danego obiektu (w ramach hierarchii danego rodzaju obiektów) lub numer strony, na której on się znajduje.

Do oznaczania obiektów służy rozkaz `\label{klucz}`, gdzie *klucz* może składać się z liter (najlepiej z wyłączeniem polskich), cyfr oraz znaków interpunkcyjnych. Oczywiście każdy oznaczany obiekt powinien posiadać klucz

unikatowy, aby nie wywołać konfliktów przy kompilacji. Rozkaz `\label` najlepiej umieszczać zaraz po definicji oznaczanego elementu (np. zaraz za rozkazem `\section{...}`), a w przypadku otoczeń – natychmiast po otwierającym `\begin{...}`. Czasami polecenie `\label` umieszcza się „w ramach” definicji obiektu, np. `\caption{... \label{...}`.

Aby odwołać się do numeru porządkowego obiektu z etykietą *klucz* używamy polecenia `\ref{klucz}`. Natomiast jeśli w miejscu odwołania chcemy uzyskać numer strony, na której znajduje się dany obiekt, stosujemy rozkaz `\pageref{klucz}`.

Na przykład bieżącą podsekcję oznaczyliśmy następująco:

```
\subsection{Odwołania czyli odsyłacze} \label{odwolania}
```

Teraz pisząc w dokumencie źródłowym

Bieżący podrozdział ma numer `\ref{odwolania}` i zaczyna się na stronie `\pageref{odwolania}`.

otrzymujemy:

Bieżący podrozdział ma numer 4.4 i zaczyna się na stronie 15.

Mechanizm automatycznego numerowania odwołań wymaga trzykrotnej kompilacji dokumentu.

4.5 Notki

Do umieszczania notek w stopce na dole strony służy polecenie `\footnote`. Na przykład fragment:

```
Umieszczamy przykładową\footnote{{} Przykładowa notka} notkę.
```

kompiluje się jako:

Umieszczamy przykładową¹ notkę.

¹ Przykładowa notka

5 Przykłady formatowania tekstu

W niniejszym rozdziale podamy parę przykładów najpopularniejszych w systemie \LaTeX otoczeń, służących do nieco bardziej zaawansowanego formatowania tekstu.

5.1 Wyrównywanie poziome

\LaTeX składa tekst domyślnie wyrównując go do lewego marginesu. W sytuacjach gdy chcemy podkreślić wyrównywanie do lewej strony, używamy otoczenia `flushleft`. Aby wyrównać fragment tekstu do marginesu prawego, zamykamy go w otoczeniu `flushright`. Do środkowania tekstu stosujemy środowisko `center`.

5.2 Wykazy czyli listy

W \LaTeX u możliwe jest stosowanie paru predefiniowanych rodzajów wykazów (z wykazem bibliograficznym zapoznaliśmy się już w podrozdziale 4.3). Ponadto można zdefiniować własne typy list. Zainteresowanych tą możliwością odsyłamy np. do rozdziału 4 książki [8].

Wspólną cechą różnych typów wykazów jest to, że każdy element wykazu rozpoczyna się rozkazem `\item`. Rozkaz ten może mieć argument opcjonalny *etykieta*, który pojawi się jako etykieta danego elementu w wykazie, zastępując etykiety domyślne dla danego typu wykazu. Tekst każdego elementu wykazu może posiadać dowolną długość i nie musi mieścić się w jednej linii.

Listy można zagłębiać, tzn. jeden wykaz może być podzbiorem innego wykazu. Niestety, zarówno wykaz nieuporządkowany, jak i wykaz uporządkowany (wyliczenie) mogą mieć co najwyżej cztery poziomy zagłębienia.

5.2.1 Wykaz nieuporządkowany

Listę nieuporządkowaną tworzymy za pomocą otoczenia `itemize`. W takiej liście domyślnymi etykietami na kolejnych poziomach zagłębienia są kolejno: `•`, `–`, `*` oraz `·`. Na przykład fragment

- Pierwszy poziom
 - Drugi poziom
 - * Trzeci poziom
 - @ Czwarty poziom ze zmienioną etykietą

powstał w wyniku kompilacji następującego kodu

```

\begin{itemize}
\item Pierwszy poziom
  \begin{itemize}
  \item Drugi poziom
    \begin{itemize}
    \item Trzeci poziom
      \begin{itemize}
      \item[@] Czwarty poziom ze zmienioną etykietą
      \end{itemize}
    \end{itemize}
  \end{itemize}
\end{itemize}
\end{itemize}

```

5.2.2 Wykaz uporządkowany

Do tworzenia wykazów uporządkowanych, czyli wyliczeń, służy otoczenie `enumerate`. Kolejne poziomy zagłębienia są tutaj etykietowane przez liczby arabskie (pierwszy poziom), małe litery alfabetu łacińskiego (poziom drugi), małe liczby rzymskie (poziom trzeci) lub duże litery alfabetu łacińskiego (poziom czwarty). Ponadto możliwe jest kolejne zastosowanie mechanizmu odsyłaczy (patrz podrozdział 4.4). Na przykład kod

```

\begin{enumerate}
\item Pierwszy poziom numerowany
  \begin{enumerate}
  \item \label{drugi} Drugi poziom numerowany
    \begin{enumerate}
    \item[*] Trzeci poziom numerowany ze zmienioną etykietą
      \begin{enumerate}
      \item Czwarty numerowany poziom
      \end{enumerate}
    \end{enumerate}
  \end{enumerate}
\end{enumerate}

```

Mój drugi poziom numerowany ma etykietę (`\ref{drugi}`).

kompiluje się jako:

1. Pierwszy poziom numerowany
 - (a) Drugi poziom numerowany
 - * Trzeci poziom numerowany ze zmienioną etykietą

A. Czwarty numerowany poziom

Mój drugi poziom numerowany ma etykietę (1a).

5.2.3 Wykaz definicyjny

Innym przykładem wykazu jest lista definicyjna, często stosowana w różnych słownikach. Tworzymy ją za pomocą otoczenia `description`. W tym otoczeniu argumentami opcjonalnymi kolejnych rozkazów `\item` są definiowane pojęcia. Domyślnie zostaną one złożone czcionką półgrubą (Bold Face). Przykładowy kod

```
\begin{description}
\item[\TeX] program (autorstwa D. Knutha), którego nie
będziemy opisywali w~tym skrypcie
\item[\LaTeX] system (autorstwa L. Lamporta), którego podstawy
opisujemy w~tym skrypcie
\end{description}
```

kompiluje się jako

TeX program (autorstwa D. Knutha), którego nie będziemy opisywali w tym skrypcie

LaTeX system (autorstwa L. Lamporta), którego podstawy opisujemy w tym skrypcie

5.3 Tabele

Do tworzenia tabel służy środowisko `tabular`. Posiada ono jeden dodatkowy argument obowiązkowy *specyfikacja*, który określa liczbę kolumn w tabeli oraz sposób ich wyrównywania poziomego (szerokość kolumn jest dobierana automatycznie, za wyjątkiem sytuacji gdy sami ją zadeklarujemy). Zatem środowisko to rozpoczynamy od `\begin{tabular}{specyfikacja}`. W specyfikacji dla każdej kolumny oddzielnie podajemy jedną z wartości: `l` (wyrównywanie do lewej), `r` (wyrównywanie do prawej) lub `c` (wycentrowanie). Jeśli chcemy zadeklarować kolumnę o ustalonej szerokości, wówczas używamy zapisu `p{szerokość}`. Oddzielenie kolumn pionową kreską otrzymamy wstawiając znak `|`. Jeśli pewien sposób formatowania powtarza się w paru sąsiednich kolumnach, to możemy użyć symbolu `*{k}{specyfikacja}`, który jest tożsamy z k -krotnym użyciem zapisu *specyfikacja*. Na przykład zapis `|*{3}{l|}` jest równoważny zapisowi `|l|l|l|`.

W ramach otoczenia `tabular` poszczególne komórki w wierszu rozdzielamy znakiem `&`, a wiersze tabeli – poleceniem `\\`. Po rozkazie `\\` można umieścić jeden lub więcej rozkazów `\hline`, które wstawiają do tabeli poziome linie między wierszami.

Rozkaz `\multicolumn{ilość}{specyfikacja}{tekst}` tworzy rubryki rozciągające się na kilka kolumn. Argument *ilość* określa liczbę połączonych kolumn. Argument *specyfikacja* ma takie samo znaczenie jak dla `tabular`. Natomiast *tekst* jest tekstem umieszczonym w rubryce.

Przykładowo, w wyniku kompilacji następującego kodu

```
\begin{tabular}{|*{2}{1}|c|p{2cm}|} \hline
\multicolumn{4}{|c|}{Indeks naszego studenta} \\ \hline \hline
Przedmiot & Prowadzący & Wymiar & Ocena \\ \hline
Analiza & prof. A. Analityczny & 30 godz. & 5.0 \\
Algebra & prof. T. Algebraiczny & 30 godz. & 5.0 \\
Geometria & prof. G. Geometryczny & 30 godz. & 5.0 \\ \hline
\end{tabular}
```

powstała następująca tabela

Indeks naszego studenta			
Przedmiot	Prowadzący	Wymiar	Ocena
Analiza	prof. A. Analityczny	30 godz.	5.0
Algebra	prof. T. Algebraiczny	30 godz.	5.0
Geometria	prof. G. Geometryczny	30 godz.	5.0

6 Wstawianie plików graficznych

\TeX i \LaTeX nie zostały stworzone z myślą o tworzeniu grafiki. W ramach systemu \LaTeX można wprawdzie kreować różne rysunki, ale albo możliwości są bardzo ograniczone (np. przy użyciu środowiska `picture`) albo metody tworzenia są dość skomplikowane (np. przy stosowaniu pakietu `PSTricks`). Osoby zainteresowane tworzeniem grafik w \LaTeX u odsyłamy np. do rozdziału 12 książki [8], rozdziału 9 książki [10] oraz rozdziału 4 książki [9].

Innym rozwiązaniem, umożliwiającym umieszczanie rysunków w tworzonych dokumentach, jest import gotowych plików graficznych spoza systemu \LaTeX . Importu takiego można dokonać na różne sposoby. Zainteresowanych odsyłamy np. do publikacji [2] lub [3]. Poniżej omówimy metodę najprostszą i najskuteczniejszą (wg. autora skryptu).

Sposób wstawiania plików graficznych do dokumentu jest zależny od tego, czy używamy kompilatora `latex`, a później programu `dvips`, aby otrzymać plik PS (*PostScript*), czy też kompilujemy dokument poprzez `pdflatex` aby otrzymać plik PDF (*Portable Document Format*). Poniżej opiszemy metody postępowania w obu przypadkach.

Jeśli chcemy otrzymać plik PS, to do naszego dokumentu będziemy wstawiali pliki graficzne w formacie EPS (*Encapsulated PostScript*). Mając grafikę w innym formacie należy skonwertować ją do postaci EPS. Można to zrobić na wiele sposobów. Autor poleca użycie programu `convert`. Jest to program spoza systemu \LaTeX , wchodzący w skład programu `ImageMagick`, a dostępny m.in. na komputerze `chaos.math.uni.wroc.pl`. Podstawowa składnia jego użycia to `convert plik-wejściowy plik-wyjściowy`, przy czym istotne jest podanie rozszerzeń (przyrostków) obu plików.

Do importu plików graficznych służy pakiet `graphicx` z opcją `dvips` (jeśli używamy programu `dvips`). Użycie tego pakietu deklarujemy w preambule rozkazem `\usepackage[dvips]{graphicx}`. Poszczególne pliki wstawiamy umieszczając w odpowiednich miejscach części właściwej dokumentu rozkaz `\includegraphics[klucz=wartość,...]{plik}`, gdzie `plik` jest nazwą (wraz z rozszerzeniem) wstawianego pliku, `klucz` może oznaczać jedną z nazw: `width` (rysunek jest skalowany do zadanej szerokości), `height` (rysunek jest skalowany do zadanej wysokości), `angle` (rysunek jest obracany o zadany kąt) lub `scale` (skalowanie równomierne), zaś `wartość` określa wielkość skalowania lub obrotu. W przypadku braku jednego z kluczy `width` lub `height` jego wartość jest automatycznie dobierana tak, by zachować proporcje rysunku.

Na przykład

```
\includegraphics[width=2cm,height=2cm,angle=90]{uwr.eps}
```

daje po wycentrowaniu następujący rezultat:



W przypadku kiedy chcemy publikować w formacie PDF, przy użyciu kompilatora `pdflatex`, uprzednio przedstawiona metoda wymaga małych modyfikacji. Po pierwsze należy przekonwertować pliki graficzne z formatu EPS na format PDF (np. przy pomocy `epstopdf` *plik-wejściowy.eps*). Po drugie trzeba zadeklarować użycie pakietu `graphicx` z opcją `pdftex`, umieszczając w preambule rozkaz `\usepackage[pdftex]{graphicx}`. Po trzecie wstawiając poszczególne pliki graficzne należy wskazywać na pliki w formacie PDF, np.

```
\includegraphics[width=2cm,height=2cm,angle=90]{uwr.pdf}
```

7 Wyrażenia matematyczne

Do tej pory używaliśmy standardowego dla \LaTeX a trybu akapitowego, w którym m.in. przejście do nowego wiersza (nowej strony) odbywało się automatycznie (por. podrozdział 2.1). Wyjątkami były polecenie `\verb` i otoczenie `verbatim`, które stosują tryb wierszowy, bez automatycznej zmiany linii. Wyrażenia matematyczne będziemy składać w trybie matematycznym. Wyróżniamy dwa rodzaje trybu matematycznego: wewnątrzakapitowy (wyrażenia są składane w ramach tekstu) i eksponowany (wyrażenia znajdują się w nowej linii i będą domyślnie wycentrowane).

Do trybu matematycznego wewnątrzakapitowego przechodzimy używając otoczenia `math`. Na przykład kod `\begin{math} x^2 \end{math}` kompiluje się jako x^2 . Zamiast `\begin{math}` można pisać skrótowo `\(` lub `$`. Podobnie `\end{math}` można skrócić do `\)` lub `$`. Przykładowo `\(x^3\)` oraz `x^4` kompiluje się jako x^3 oraz x^4 .

Do trybu matematycznego eksponowanego przechodzimy za pomocą otoczenia `displaymath`. Na przykład następujący kod `\begin{displaymath} x^{y^2} \end{displaymath}` daje po przetworzeniu

$$x^{y^2}$$

Tak jak w poprzednim przypadku możemy używać skrótów: `\[` lub `$$` oraz `\]` lub `$$`, odpowiednio. Przykładowo `\[x^2 \]` oraz `$$ x^3`. `$$` przetwarza się jako:

$$x^2$$

oraz

$$x^3.$$

Zwróćmy uwagę, iż większość wyrażeń matematycznych (np. ułamki piętrowe, sumy itd.) będzie składana odmiennie w zależności od użytego rodzaju trybu matematycznego.

7.1 Indeksy, wykładniki i pochodne

Do tworzenia indeksów górnych oraz wykładników służy znak `^` (patrz przykłady powyżej). Indeksy dolne otrzymujemy za pomocą znaku `_`: kod `x_2` daje x_2 . Jeśli w indeksach występuje więcej niż jeden symbol, to używamy nawiasów klamrowych do grupowania. Na przykład napis $A_{n+1}^{i,j,k}$ powstał z polecenia `$A^{\{i,j,k\}}_{n+1}$`.

Do pisania pochodnych niskich rzędów używamy apostrofu. Przykładowo $f'(x) = g''(x) = h'''(x)$ otrzymaliśmy z kodu `$f'(x)=g''(x)=h'''(x)$`.

7.2 Odstępy

W trybie matematycznym \LaTeX nie tworzy odstępów w miejscach umieszczenia spacji (np. kod $\$ab=a b=a b\$$ daje po kompilacji $ab = ab = ab$). Odstępy są dobierane automatycznie, w zależności od kontekstu (np. po obu stronach operatora binarnego). Do deklarowania dodatkowych odstępów w trybie matematycznym służą rozkazy: \backslash , (mały odstęp), $\backslash:$ (średni odstęp), $\backslash;$ (duży odstęp), $\backslash!$ (odstęp „ujemny”), $\backslash_$ (zwykła spacja), \backslashquad (odstęp równy kwadratowi) oraz \backslashqqquad (odstęp równy dwóm kwadratowi). Dokładne miary w.w. odstępów można znaleźć np. w rozdziale 3.4 pozycji [1] (kwadrat to w przybliżeniu szerokość „M” w bieżącej czcionce).

7.3 Ułamki

Ułamki generujemy przy pomocy polecenia $\backslashfrac{\text{\code{licznik}}\{\text{\code{mianownik}}\}$. Oczywiście w liczniku lub mianowniku również możemy umieścić ułamki. Przykładowy napis $\backslashfrac{\backslashfrac{1}{x+y}-1}{a+b+c}$ daje wewnątrz akapitu $\frac{\frac{1}{x+y}-1}{a+b+c}$, natomiast w trybie eksponowanym

$$\frac{\frac{1}{x+y} - 1}{a + b + c}.$$

7.4 Pierwiastki

Pierwiastki tworzymy posługując się poleceniem \backslashsqrt o następującej składni $\backslashsqrt[\text{\code{stopień}}]{\text{\code{wyrażenie-podpierwiastkowe}}}$. Jeśli stopień pierwiastka wynosi 2 (tzn. mamy pierwiastek kwadratowy), to pomijamy argument opcjonalny. Przykładowo $\$ \backslashsqrt{a}$, $\backslashsqrt[4]{7}$ daje \sqrt{a} , $\sqrt[4]{7}$.

7.5 Sumy, iloczyny i całki

Aby wygenerować symbol sumy posługujemy się poleceniem \backslashsum postaci $\backslashsum_{\text{\code{indeks-dolny}}\wedge\{\text{\code{indeks-górny}}\}}$. Na przykład napis $\backslashsum_{i=1}\wedge n$ daje w tekście $\sum_{i=1}^n$, a w wierszu eksponowanym

$$\sum_{i=1}^n.$$

Podobnie, aby wygenerować symbol iloczynu stosujemy rozkaz \backslashprod (indeks wpisujemy tak jak przy \backslashsum).

Do tworzenia całek służy rozkaz \backslashint , analogiczny składniowo do \backslashsum (oczywiście w miejsce indeksów wpisujemy odpowiednio granice całkowania).

Na przykład fragment `\int_a^b f(x) \, dx` kompiluje się w zależności od trybu jako $\int_a^b f(x) dx$ lub

$$\int_a^b f(x) dx.$$

Do utworzenia symbolu całki po krzywej zamkniętej używamy polecenia `\oint`.

Dla wyraźniejszego zaznaczenia granic całkowania w powyższych symbolach można zastosować rozkaz `\limits`. Na przykład kompilacja kodu `$$\oint_I \quad \oint\limits_I$$` daje

$$\oint_I \quad \oint\limits_I.$$

7.6 Funkcje

Jeśli w trybie matematycznym napiszemy nazwę którejkolwiek z popularnych funkcji wprost, tzn. np. `sin`, to \LaTeX potraktuje ten napis jak iloczyn paru zmiennych i złoży go kursywą, np. *sin*. Aby złożyć nazwy funkcji pismem prostym (tak jak przyjęło się je pisać), należy poprzedzić je znakiem `\`. Polecenia generujące nazwy podstawowych funkcji wymieniono w tabeli 1.

<code>\arccos</code>	<code>\arcsin</code>	<code>\arctan</code>	<code>\arg</code>	<code>\cos</code>	<code>\cosh</code>	<code>\cot</code>	<code>\coth</code>
<code>\csc</code>	<code>\deg</code>	<code>\det</code>	<code>\dim</code>	<code>\exp</code>	<code>\gcd</code>	<code>\hom</code>	<code>\inf</code>
<code>\ker</code>	<code>\lg</code>	<code>\lim</code>	<code>\liminf</code>	<code>\limsup</code>	<code>\ln</code>	<code>\log</code>	<code>\max</code>
<code>\min</code>	<code>\Pr</code>	<code>\sec</code>	<code>\sin</code>	<code>\sinh</code>	<code>\sup</code>	<code>\tan</code>	<code>\tanh</code>

Tabela 1: Podstawowe funkcje

7.7 Skalowalne nawiasy

Nawiasy w trybie matematycznym mogą być powiększane do wysokości obejmowanego wyrażenia. Aby nawiasy zostały odpowiednio przeskalowane należy poprzedzić je poleceniami `\left` (przed nawiasem otwierającym) oraz `\right` (przed nawiasem zamykającym). Należy przy tym pamiętać, że nawiasy zawsze występują w parach. Zatem w sytuacji, gdy na wydruku powinien pojawić się wyłącznie jeden nawias, w miejscu drugiego należy umieścić tzw. nawias pusty (oznaczany symbolem `.`). Na przykład polecenia

```
$$ \left[ \frac{1}{x^2+1} \right]^3 \quad \left\{ \{x+y=0 \atop 2x-y=1\} \right.
```

generują po kompilacji

$$\left[\frac{1}{x^2 + 1} \right]^3 \quad \begin{cases} x + y = 0 \\ 2x - y = 1 \end{cases}$$

7.8 Macierze

Macierze są tworzone przez środowisko `array`. Składnia tego środowiska jest taka sama jak składnia środowiska `tabular` (patrz podrozdział 5.3). Nie należy przy tym zapominać o umieszczeniu całego otoczenia `array` w trybie matematycznym. Przykładowo, kod

```

 $\begin{array}{ccc}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{array}$ 

```

kompiluje się jako

$$\begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{array}$$

Otoczenia `array` można również użyć do definiowania funkcji przez przypadki. Na przykład polecenia

```

 $|x| = \begin{array}{r} \left\{ \begin{array}{l} x \text{ dla } x \geq 0 \\ -x \text{ dla } x < 0 \end{array} \right. \end{array}$ 

```

dają po przetworzeniu

$$|x| = \begin{cases} x & \text{dla } x \geq 0 \\ -x & \text{dla } x < 0 \end{cases}$$

Więcej o zastosowaniach otoczenia `array` można przeczytać np. w podrozdziale 10.1 książki [8].

7.9 Zwykły tekst we wzorach

Czasami zachodzi potrzeba umieszczenia fragmentu zwykłego tekstu we wzorze matematycznym. Można tego dokonać stosując rozkaz `\textrm`. Jego argumentem obowiązkowym jest oczywiście dany fragment tekstu. Na przykład fragment

$$x^2 \geq 0 \quad \text{dla każdego } x \in \mathbb{R}$$

powstał z napisu

```

 $x^2 \geq 0 \text{ dla każdego } x \in \mathbb{R}$ 

```

7.10 Podkreślenia i inne ozdobniki

Do tworzenia poziomych kresek nad lub pod wyrażeniami służą polecenia `\overline` i `\underline`. Poziome klamry nad lub pod wyrażeniami generowane są przez rozkazy `\overbrace` oraz `\underbrace`. Do dwóch ostatnich poleceń można dołączyć indeks górny bądź dolny, który zdefiniuje opis umieszczony nad lub pod klamrą. Na przykład kod

```

 $\underline{m+n}$  \quad  $\overline{x+y}$  \quad
 $\overbrace{x_1, x_2, \dots, x_n}^{\text{zmienne}}$  \quad
 $\underbrace{a+a+\dots+a}_{n \text{ składników}}$ 

```

daje po przetworzeniu

$$\underline{m+n} \quad \overline{x+y} \quad \overbrace{x_1, x_2, \dots, x_n}^{\text{zmienne}} \quad \underbrace{a+a+\dots+a}_{n \text{ składników}}$$

7.11 Matematyczne kroje pisma

W notacji matematycznej używamy wiele symbolów literowych, złożonych niestandardowymi krojami pisma. Sztandarowym przykładem są oznaczenia zbiorów liczbowych: \mathbb{N} , \mathbb{Z} , \mathbb{Q} , \mathbb{R} , \mathbb{C} . Niektóre dostępne w trybie matematycznym kroje pisma przedstawione są w tabelicy 2.

Przykład	Rozkaz	Wymagany pakiet
ABCdef	<code>\mathrm{ABCdef}</code>	
ABCdef	<code>\mathit{ABCdef}</code>	
\mathnormal{ABCdef}	<code>\mathnormal{ABCdef}</code>	
\mathcal{ABC}	<code>\mathcal{ABC}</code>	
\mathscr{ABC}	<code>\mathscr{ABC}</code>	eucal z opcją <code>mathscr</code>
\mathfrak{ABCdef}	<code>\mathfrak{ABCdef}</code>	eufrak (podzbiór <code>amsfonts</code>)
\mathbb{ABC}	<code>\mathbb{ABC}</code>	<code>amsfonts</code>

Tabela 2: Matematyczne kroje pisma

7.12 Akcenty

Do tworzenia różnorodnych akcentów matematycznych stosujemy polecenia, zawarte w tabelicy 3.

\hat{a}	<code>\hat{a}</code>	\dot{a}	<code>\dot{a}</code>	\vec{a}	<code>\vec{a}</code>
\check{a}	<code>\check{a}</code>	\ddot{a}	<code>\ddot{a}</code>	\grave{a}	<code>\grave{a}</code>
\bar{a}	<code>\bar{a}</code>	\tilde{a}	<code>\tilde{a}</code>	\widehat{AB}	<code>\widehat{AB}</code>
\breve{a}	<code>\breve{a}</code>	\acute{a}	<code>\acute{a}</code>	\widetilde{AB}	<code>\widetilde{AB}</code>

Tabela 3: Akcenty matematyczne

7.13 Litery greckie

Rozkazy generujące małe litery alfabetu greckiego są przedstawione w tablicy 4. Natomiast tablica 5 przedstawia odpowiednie polecenia dla dużych liter tego alfabetu.

α	<code>\alpha</code>	β	<code>\beta</code>	γ	<code>\gamma</code>	δ	<code>\delta</code>
ϵ	<code>\epsilon</code>	ε	<code>\varepsilon</code>	ζ	<code>\zeta</code>	η	<code>\eta</code>
θ	<code>\theta</code>	ϑ	<code>\vartheta</code>	ι	<code>\iota</code>	κ	<code>\kappa</code>
λ	<code>\lambda</code>	μ	<code>\mu</code>	ν	<code>\nu</code>	ξ	<code>\xi</code>
o	<code>o</code>	π	<code>\pi</code>	ϖ	<code>\varpi</code>	ρ	<code>\rho</code>
ϱ	<code>\varrho</code>	σ	<code>\sigma</code>	ς	<code>\varsigma</code>	τ	<code>\tau</code>
υ	<code>\upsilon</code>	ϕ	<code>\phi</code>	φ	<code>\varphi</code>	χ	<code>\chi</code>
ψ	<code>\psi</code>	ω	<code>\omega</code>				

Tabela 4: Małe litery greckie

Γ	<code>\Gamma</code>	Δ	<code>\Delta</code>	Θ	<code>\Theta</code>	Λ	<code>\Lambda</code>	Ξ	<code>\Xi</code>
Π	<code>\Pi</code>	Σ	<code>\Sigma</code>	Υ	<code>\Upsilon</code>	Φ	<code>\Phi</code>	Ψ	<code>\Psi</code>
Ω	<code>\Omega</code>								

Tabela 5: Duże litery greckie

7.14 Niektóre symbole matematyczne

Zapisując symbolicznie „funkcja f ze zbioru X w zbiór Y ” nie będziemy stowowali dwukropka, lecz polecenie `\colon`. Napis $f: X \rightarrow Y$ powstał z poleceń `\$f \colon X \to Y\$`.

Do składu dwumianów lub konstrukcji podobnych wykorzystujemy pole-

cenie `\choose` lub `\atop`. Na przykład napis

$$\binom{n}{k} \quad \begin{array}{c} x \\ y \end{array}$$

powstał z poleceń

`$$ {n \choose k} \quad \{x \atop y\} $$`

W tabeli 6 przedstawiono parę innych przydatnych symboli matematycznych (symbol \square jest dostępny po dołączeniu pakietu `amsmath`). Kompletnie zestawienie symboli, dostępnych standardowo w trybie matematycznym, można znaleźć np. w podrozdziale 3.9 publikacji [1] lub w dodatku A książki [8].

\neg	<code>\neg</code>	\neq	<code>\neq</code>	\leq	<code>\leq</code>	\geq	<code>\geq</code>
\dots	<code>\dots</code>	\cdots	<code>\cdots</code>	\vdots	<code>\vdots</code>	\ddots	<code>\ddots</code>
∞	<code>\infty</code>	∂	<code>\partial</code>	∇	<code>\nabla</code>	\emptyset	<code>\emptyset</code>
\aleph	<code>\aleph</code>	\forall	<code>\forall</code>	\exists	<code>\exists</code>	\square	<code>\square</code>
\in	<code>\in</code>	\subset	<code>\subset</code>	\iff	<code>\iff</code>	\rightarrow	<code>\rightarrow</code>
\implies	<code>\implies</code>	\Longrightarrow	<code>\Longrightarrow</code>	\Longleftarrow	<code>\Longleftarrow</code>		

Tabela 6: Różne symbole matematyczne

7.15 Wzory numerowane

W publikacjach matematycznych numeruje się zazwyczaj wzory eksponowane. W systemie \LaTeX do tworzenia automatycznie numerowanych jednowierszowych wzorów wyeksponowanych służy otoczenie `equation`. Użycie tego otoczenia powoduje automatyczne wejście do trybu matematycznego. Dodatkową zaletą tego środowiska jest fakt, iż umożliwia ono stosowanie mechanizmu automatycznych odwołań (polecenia `\label` oraz `\ref` – por. podrozdział 4.4). Przykładowo do równania

$$(1) \quad a^2 + b^2 = c^2$$

utworzonego poleceniami

```
\begin{equation} \label{pitagoras} a^2+b^2=c^2 \end{equation}
```

możemy odwołać się pisząc w tekście

Korzystając z równania (\ref{pitagoras}) można \ldots

co zostanie przetworzone na: Korzystając z równania (1) można

L^AT_EX domyślnie umieszcza numery wzorów przy prawym marginesie. Aby numery te znajdowały się po lewej stronie stosujemy opcję leqno klasy dokumentu, np.

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

7.16 Wzory wielowierszowe

Do składu wzorów wielowierszowych stosujemy otoczenie eqnarray, które jest zbliżone do otoczenia array z kolumnami typu {rcl} (porównaj podrozdział 7.8). Każdy z wierszy umieszczonych w tym otoczeniu zostanie opatrzone własnym numerem. Jeśli chcemy, by któryś z wierszy pozostał bez numeracji, należy umieścić w nim rozkaz \nonumber. Wzory wielowierszowe całkowicie pozbawione numeracji tworzymy przy pomocy otoczenia eqnarray*. Użycie powyższych otoczeń powoduje automatyczne wejście do trybu matematycznego. Na przykład napis

$$\begin{aligned}
 (2) \quad \pi_f &= \sum_{n \in \mathbb{N}} f(p_n) p_n \\
 &= f(1) \cdot 1 + f(\sharp) \cdot \sharp + f(rot) \cdot rot + f(CH) \cdot CH + f(I) \cdot I + \\
 &+ \sum f(K_j) \cdot K_j + \sum f(M_i) \cdot M_i + \sum f(N_l) \cdot N_l
 \end{aligned}$$

powstał po kompilacji następującego kodu:

```
\begin{eqnarray}
\pi_f &= & \sum_{n \in \mathbb{N}} f(p_n) \ , \ p_n \ \backslash \\
\backslash \nonumber &= & f(1) \ \cdot 1 + f(\sharp) \cdot \sharp + \\
& f(rot) \cdot rot + f(CH) \cdot CH + f(I) \cdot I + \backslash \\
\backslash \nonumber &+ & \sum f(K_j) \cdot K_j + \sum f(M_i) \cdot M_i + \\
& \sum f(N_l) \cdot N_l \\
\backslash \end{eqnarray}
```

7.17 Struktury tekstowe

W publikacjach matematycznych przyjęło się wyróżniać w składzie i numerować twierdzenia, lematy, definicje i tym podobne elementy. Takie struktury tekstowe definiujemy w preambule rozkazem \newtheorem o następującej składni: \newtheorem{nazwa}[nazwa']{tekst}[punkt]. Argument obowiązkowy nazwa to nazwa nowo utworzonego otoczenia, zaś argument

tekst określa napis, który wystąpi na wydruku (np. „Twierdzenie”, „Lemat” itp.). Przy pomocy argumentów opcjonalnych określamy sposób numerowania struktury tekstowej (w razie ich braku będzie ona numerowana kolejnymi liczbami naturalnymi w ramach całego dokumentu). Podanie argumentu *nazwa* spowoduje, że oba środowiska *nazwa* oraz *nazwa*’ będą posiadały wspólną numerację. Natomiast argument *punkt* określa numerowanie danej struktury wewnątrz jednostki podziału dokumentu i może przyjąć jedną z wartości: `section`, `subsection` itd. (patrz rozdział 4).

Po umieszczeniu rozkazu `\newtheorem{nazwa}...` w preambule, w części właściwej dokumentu używamy otoczenia *nazwa* w standardowy sposób. Jako argument opcjonalny po `\begin{nazwa}` możemy wpisać nazwisko twórcy, komentarz itp.

Na przykład, jeżeli w preambule umieścimy polecenia

```
\newtheorem{tw}{Twierdzenie}[section]
\newtheorem{lem}[tw]{Lemat}
```

to po wpisaniu w części właściwej dokumentu kodu

```
\begin{tw}[Pitagoras] W trójkącie prostokątnym \ldots \end{tw}
\begin{lem} Jeżeli założymy \ldots \end{lem}
```

otrzymamy

Twierdzenie 7.1 (Pitagoras) *W trójkącie prostokątnym ...*

Lemat 7.2 *Jeżeli założymy ...*

7.18 Parę słów na zakończenie

Przedstawione w niniejszym rozdziale elementy formatowania wzorów matematycznych to zaledwie mały fragment możliwości \LaTeX -a. Zainteresowanych tematem odsyłamy np. do rozdziału 3 publikacji [1] bądź też do rozdziałów 9–10 książki [8]. Ponadto należy stwierdzić, iż w standardowym \LaTeX -u występuje kilka ograniczeń, ujawniających się przy składzie bardziej skomplikowanych wyrażeń matematycznych. Niedostatki te możemy przewyciężyć używając pakietów z dystrybucji $\mathcal{A}\mathcal{M}\mathcal{S}$ - \LaTeX . Krótkie, aczkolwiek wystarczające dla początkujących, wprowadzenie do tej dystrybucji można znaleźć np. w rozdziale 11 pozycji [8].

8 Obiekty ruchome

Często zdarza się, że chcemy odwołać się do tabeli lub rysunku, zamieszczonych w naszym dokumencie. Możliwość taką daje nam mechanizm tzw. obiektów ruchomych. Ponadto umożliwia on „ładne” tytułowanie tabel i rysunków oraz automatyczne umieszczanie takich tytułów w spisie tych obiektów.

Ruchome tablice i rysunki definiujemy przy pomocy otoczeń `table` oraz `figure`, odpowiednio. Otoczenia te mają jeden argument opcjonalny *miejsce*, który będzie wskazywał \LaTeX -owi miejsce umieszczenia obiektu ruchomego: `\begin{table}[miejsce]` lub `\begin{figure}[miejsce]`. Argument *miejsce* jest sekwencją od jednego do pięciu znaków: `h` (here=tutaj), `t` (top=na górze strony), `b` (bottom=na dole strony), `p` (page of floats=osobna strona na końcu dokumentu) lub `!`. Domyślnie przyjmowana jest wartość `tbp`. Niestety, \LaTeX nie zawsze umieszcza obiekt zgodnie z intencją użytkownika (np. wtedy, gdy na danej stronie brakuje na niego miejsca). W takim przypadku może być pomocne umieszczenie znaku `!` na początku argumentu *miejsce*. Znak ten mówi \LaTeX -owi, by ten zignorował większość parametrów regulujących umieszczanie obiektów ruchomych (np. maksymalną ilość obiektów na stronie). Więcej o mechanizmie umiejscawiania obiektów ruchomych można przeczytać np. w podrozdziale 2.11 publikacji [1] lub podrozdziale 8.4.1 książki [10].

W ramach otoczenia `table` tabele tworzymy np. przy pomocy otoczenia `tabular` (patrz podrozdział 5.3). W otoczeniu `figure` rysunki wstawiamy np. poleceniem `\includegraphics` (patrz rozdział 6).

Tytuły obiektów ruchomych wstawiamy rozkazem `\caption`. \LaTeX automatycznie nazwie i ponumeruje te obiekty (oddzielnie tabele i oddzielnie rysunki). Do utworzenia spisu tabel służy rozkaz `\listoftables`, natomiast rozkaz `\listoffigures` tworzy spis wszystkich rysunków (porównaj polecenie `\tableofcontents` w podrozdziale 4.2). Odsyłacze do obiektów ruchomych tworzymy przy pomocy polecenia `\label` (umieszczonego w ramach argumentu polecenia `\caption` lub tuż za nim) oraz polecenia `\ref` (patrz podrozdział 4.4).

Jako przykład podamy następujący kod:

```
Poniżej przedstawiamy rysunek \ref{rysunek} i tablicę
\ref{tabelka}.
\begin{table}[h]
\begin{center}
\begin{tabular}{|*{2}{1}|c||p{2cm}|} \hline
\multicolumn{4}{|c|}{Indeks naszego studenta} \\ \hline
Przedmiot & Prowadzący & Wymiar & Ocena \\ \hline
```



```

\multicolumn{4}{|c|}{Ciąg dalszy znamy \ldots} \\ \hline
\end{tabular}
\caption{Taka sobie tabelka \label{tabelka}}
\end{center}
\end{table}
\begin{figure}[h]
\begin{center}
\includegraphics[width=2cm,height=2cm]{uwr.eps}
\caption{Taki sobie rysunek \label{rysunek}}
\end{center}
\end{figure}

```

Kod ten po kompilacji daje następujący wynik:
 Poniżej przedstawiamy rysunek 1 i tablicę 7.

Indeks naszego studenta			
Przedmiot	Prowadzący	Wymiar	Ocena
Ciąg dalszy znamy ...			

Tabela 7: Taka sobie tabelka



Rysunek 1: Taki sobie rysunek

9 Praca z dużym dokumentem

Pracując nad dużym dokumentem, np. pracą magisterską, wygodnie jest podzielić go na mniejsze części, np. rozdziały, utworzyć jeden plik główny i dołączyć do niego zawartość wszystkich części. Takie dołączenie umożliwia rozkaz `\include{plik}`, gdzie argument *plik* jest nazwą pliku (bez rozszerzenia `.tex`) zawierającego treść jednej części. Istotne jest, aby w takim pliku nie umieszczać preambuły oraz poleceń `\begin{document}` i `\end{document}`. Pojawia się one jedynie w pliku głównym. Zawartość każdego pliku dołączonego poleceniem `\include` będzie składana od nowej strony. Aby zawartość zbioru była składana bez rozpoczynania nowej strony, należy użyć polecenia `\input` zamiast `\include`. Aby złożyć cały dokument kompilujemy plik główny.

Jako przykład podajemy fragment pliku głównego niniejszego skryptu.

```
\documentclass[12pt,a4paper,leqno,twoside]{article}
\usepackage[polish]{babel}
\usepackage[T1]{fontenc}
\usepackage{polski}
\usepackage[latin2]{inputenc}
\usepackage[dvips]{graphicx}
\usepackage{amsfonts}
\usepackage{amsmath}
\usepackage[mathscr]{eucal}

\begin{document}
\include{tytul}
\include{1}
\include{2}
\include{3}
\include{4}
\include{5}
\include{6}
\include{biblio}
\end{document}
```

10 Modyfikowanie L^AT_EX-a

Do tej pory korzystaliśmy z gotowych poleceń oraz środowisk systemu L^AT_EX. Teraz zajmiemy się tworzeniem własnych rozkazów i otoczeń.

Własne rozkazy definiujemy rozkazem `\newcommand` o następującej składni: `\newcommand{\nazwa}[liczba-argumentów]{tekst}`. W celu utworzenia własnych otoczeń posługujemy się instrukcją `\newenvironment`:

```
\newenvironment{nazwa}[liczba-argumentów]{początek}{koniec}.
```

Oba powyższe polecenia należy umieszczać w preambule dokumentu. Argument *nazwa* to nazwa nowo tworzonego polecenia (otoczenia). W pliku źródłowym obiekty te będziemy wywoływać w sposób standardowy: `\nazwa` lub `\begin{nazwa}` oraz `\end{nazwa}`. Argument *liczba-argumentów* wyznacza opcjonalnie ilość argumentów obowiązkowych nowo tworzonego obiektu. Może on przyjmować wartość naturalną z zakresu od 1 do 9. Argument *tekst* oznacza ciąg poleceń, które będą wykonywane po napotkaniu napisu `\nazwa`. Argumenty *początek* oraz *koniec* to polecenia, które będą wykonywane po napotkaniu `\begin{nazwa}` i `\end{nazwa}`, odpowiednio. W ramach napisów *tekst*, *początek* oraz *koniec* pierwszy argument nowo utworzonej struktury będzie występować jako #1, drugi – jako #2 itd. Przy wykonywaniu nowego polecenia (otoczenia) w miejsce każdego z napisów #1, #2, itd. zostanie podstawiony odpowiedni argument obowiązkowy, użyty przy wywołaniu tego polecenia (otoczenia).

Jako przykład zdefiniujemy rozkaz `\rowvec` oraz otoczenie `tititem`. Aby to uczynić w preambule umieścimy następujący kod (znaków % użyliśmy dla przejrzystości kodu):

```
\newcommand{\rowvec}[2]{$\left[ \{#1\}_{1}, \ldots, \{#1\}_{\{#2\}} \right.$}
\right]$}
\newenvironment{tititem}[1]{\par \begin{center} \textbf{#1} %
\end{center} \begin{itemize}}{\end{itemize}}
```

Tak zdefiniowane obiekty możemy wywołać w części właściwej dokumentu pisząc na przykład:

```
\begin{tititem}{Tytułik.} \item X oznacza \rowvec{x}{n}
\item Inny element \end{tititem}
```

Po kompilacji otrzymamy na wydruku:

Tytułik.

- X oznacza $[x_1, \dots, x_n]$
- Inny element

11 Często Zadawane Pytania

Pytanie:

Mam dużą tabelę, która nie mieści się na kartce. Jak ją przeskalować bądź obrócić (np. o 90^0) tak, by się mieściła?

Odpowiedź:

Do skalowania różnych obiektów służą rozkazy `\scalebox` lub `\resizebox`. Obróty można wykonać używając rozkazu `\rotatebox` lub pakietów `lscap` lub `rotating`. Zainteresowanym polecam lekturę rozdziału 9 publikacji [2].

Pytanie:

Co trzeba zrobić, aby równania były numerowane w ramach rozdziałów (sekcji)?

Odpowiedź:

Są na to (co najmniej) trzy sposoby:

1. Użycie $\mathcal{A}\mathcal{M}\mathcal{S}$ - \LaTeX -a – w preambule umieszczamy

```
\usepackage{amsmath}
\numberwithin{equation}{section}
```

Ten sposób ma dodatkowe zalety: otoczenie `subequations` pozwala numerować „podwzory” (np. (4.9a), (4.9b), (4.9c) itd.), a ponadto wzory wielowierszowe są domyślnie numerowane na środku swej wysokości.

2. Przedefiniowanie rozkazu `\theequation` – w preambule umieszczamy

```
\renewcommand{\theequation}{\thesection.\arabic{equation}}
```

Ten sposób wymaga pewnej kontroli nad licznikiem `equation`, co uzyskamy pisząc dodatkowo w preambule

```
\newcommand{\mysection}[1]{\section{ #1 %
\setcounter{equation}{0} }}
```

i używając konsekwentnie w części właściwej dokumentu `\mysection` zamiast `\section`.

3. Użycie klasy dokumentu `report` lub `book` zamiast `article`.

W klasach `report` oraz `book` równania numerowane są wewnątrz rozdziałów (`chapter`) np. X.1, X.2, ..., gdzie X to numer rozdziału.

Wymaga to używania rozkazu `\chapter` (rozdział) zamiast `\section` (sekcja), `\section` zamiast `\subsection` itd. W klasie `article` nie ma rozkazu `\chapter`. (Uwaga: Użycie w.w. klas zmienia wygląd pracy. Warto poczytać o właściwościach poszczególnych klas np. w rozdziale 8 książki [8]).

Pytanie:

W pliku PDF, utworzonym przy pomocy kompilatora `pdflatex`, znaki są brzydkie, postrzępione, co widać szczególnie po powiększeniu dokumentu. Jak to naprawić?

Odpowiedź:

Nie wglębiamy się w szczegóły techniczne problem ten można rozwiązać stosując rodzinę fontów Latin Modern. Aby tego dokonać należy do preambuły dokumentu dopisać `\usepackage{lmodern}` i ... liczyć na to, że dysponujemy dystrybucją \TeX -a z zainstalowanymi fontami LM (większość aktualnych dystrybucji spełnia ten warunek). Jeśli nie, to musimy doinstalować te fonty w swojej dystrybucji.

Literatura

- [1] Tobias Oetiker, Hubert Partl, Irene Hyna i Elisabeth Schlegl, *Nie za krótkie wprowadzenie do systemu $\LaTeX 2_{\epsilon}$* ; polskie tłumaczenie autorstwa Janusza Gołdasza, Ryszarda Kubiaka i Tomasza Przechlewskiego jest dostępne pod adresem <ftp://ftp.gust.org.pl/pub/CTAN/info/lshort/polish/>.
- [2] Wojciech Myszka, *Włączanie grafik do tekstów w $\LaTeX 2_{\epsilon}$* , publikacja aktualizowana, w postaci dokumentu elektronicznego dostępna pod adresem <http://www.immt.pwr.wroc.pl/~myszka/grafika/grafika.pdf>.
- [3] Keith Reckdahl, *Using Imported Graphics in $\LaTeX 2_{\epsilon}$ and pdf \LaTeX* , publikacja w postaci dokumentu elektronicznego dostępna jest pod adresem <ftp://sunsite.icm.edu.pl/pub/CTAN/info/epslatex.pdf>.
- [4] Donald E. Knuth, *The $T_{E}X$ book*, Addison Wesley Publishing Company, 1984
- [5] Leslie Lamport, *\LaTeX : A Document Preparation System*, Addison Wesley Publishing Company, 1985 (2nd edition 1994)
- [6] Leslie Lamport, *\LaTeX : System przygotowywania dokumentów*, ARIEL, Kraków, 1992
- [7] Leslie Lamport, *\LaTeX : System opracowywania dokumentów. Podręcznik i przewodnik użytkownika*, WNT, Warszawa, 2004
- [8] Antoni Diller, *\LaTeX wiersz po wierszu*, Wyd. Helion, 2001
- [9] Michael Goosens, Sebastian Rahtz, Frank Mittelbach, *The \LaTeX Graphics Companion: Illustrating Documents With $T_{E}X$ and Postscript*, Addison-Wesley, Reading, Massachusetts, 1997
- [10] Jerzy Kucharczyk, *Wprowadzenie do systemu składu tekstów drukarskich \LaTeX* , Wyd. Uniwersytetu Wrocławskiego, 1994
- [11] Kazimierz M. Borkowski, *\LaTeX , Profesjonalny skład publikacji*, Wyd. Adam Marszałek, Toruń, 1992
- [12] Ewaryst Rafajłowicz, Wojciech Myszka, *\LaTeX , Podręcznik użytkownika*, Akademicka Oficyna Wydawnicza PLJ, Warszawa, 1992
- [13] Ewaryst Rafajłowicz, Wojciech Myszka, *\LaTeX – zaawansowane narzędzia*, Akademicka Oficyna Wydawnicza PLJ, Warszawa, 1996

Skorowidz

#, 9
\$, 9, 23
\$\$, 23
\$\backslashbackslash\$, 9
%, 9, 10
&, 9, 20
^, 9, 23
_, 9, 23
{, 9
}, 9
~, 9
\(, 23
\), 23
\,, 24
\:, 24
\;, 24
\[, 23
\\, 8, 20
_, 24
\!, 24
\], 23
\, 5, 9

\addcontentsline, 13
amsmath, 29
\and, 13
array, 26, 30
\atop, 29
\author, 13

babel, 6
\begin, 5
\bibitem, 14

\caption, 16, 32
center, 5, 17
\chapter, 37
\choose, 29
\cite, 14

\colon, 28

\date, 13
description, 19
displaymath, 23
document, 6
\documentclass, 6, 14

\emph, 11
\end, 5
enumerate, 18
eqnarray, 30
eqnarray*, 30
equation, 29

fancyhdr, 7
figure, 32
flushleft, 17
flushright, 17
fontenc, 6
\footnote, 16
\footnotesize, 11
\frac, 24

graphicx, 21, 22

\hfill, 9
\hline, 20
\hspace, 8
\hspace*, 9
\huge, 11
\Huge, 11

\include, 34
\includegraphics, 21, 32
\input, 34
inputenc, 6
\int, 24
\item, 17
itemize, 17

`\label`, 15, 29, 32
`\large`, 11
`\Large`, 11
`\LARGE`, 11
`\LaTeX`, 5
`\ldots`, 10
`\left`, 25
`\limits`, 25
`\listoffigures`, 32
`\listoftables`, 32
`lscape`, 36

`\maketitle`, 13
`math`, 23
`\multicolumn`, 20

`\newcommand`, 35
`\newenvironment`, 35
`\newline`, 8
`\newpage`, 8
`\newtheorem`, 30
`\noindent`, 8
`\nonumber`, 30
`\normalsize`, 11

`\oint`, 25
`otoczenie`, 5

- `array`, 26, 30
- `center`, 5, 17
- `description`, 19
- `displaymath`, 23
- `document`, 6
- `enumerate`, 18
- `eqnarray*`, 30
- `eqnarray`, 30
- `equation`, 29
- `figure`, 32
- `flushleft`, 17
- `flushright`, 17
- `itemize`, 17
- `math`, 23
- `picture`, 21
- `subequations`, 36
- `table`, 32
- `tabular`, 19, 26, 32
- `thebibliography`, 14
- `titlepage`, 14
- `verbatim`, 12, 23

`\overbrace`, 27
`\overline`, 27

`\pageref`, 16
`\pagestyle`, 7
`pakiet`, 6

- `amsfonts`, 29
- `babel`, 6
- `fancyhdr`, 7
- `fontenc`, 6
- `graphicx`, 21, 22
- `inputenc`, 6
- `lscape`, 36
- `polski`, 6
- `PSTricks`, 21
- `rotating`, 36

`\par`, 8
`\paragraph`, 13
`\part`, 13
`picture`, 21
`polski`, 6
`\prod`, 24
`PSTricks`, 21

`\qqquad`, 24
`\quad`, 24

`\ref`, 16, 29, 32
`\resizebox`, 36
`\right`, 25
`\rotatebox`, 36
`rotating`, 36
`rozkaz`, 5

- `$`, 23
- `$$`, 23
- `\backslash$`, 9

<code>%</code> , 10	<code>\includegraphics</code> , 21, 32
<code>&</code> , 20	<code>\input</code> , 34
<code>^</code> , 23	<code>\int</code> , 24
<code>-</code> , 23	<code>\item</code> , 17
<code>~</code> , 9	<code>\label</code> , 15, 29, 32
<code>\(</code> , 23	<code>\large</code> , 11
<code>\)</code> , 23	<code>\Large</code> , 11
<code>\,</code> , 24	<code>\LARGE</code> , 11
<code>\:</code> , 24	<code>\LaTeX</code> , 5
<code>\;</code> , 24	<code>\ldots</code> , 10
<code>\[</code> , 23	<code>\left</code> , 25
<code>\\</code> , 8, 20	<code>\limits</code> , 25
<code>_</code> , 24	<code>\listoffigures</code> , 32
<code>\!</code> , 24	<code>\listoftables</code> , 32
<code>\]</code> , 23	<code>\maketitle</code> , 13
<code>\,</code> , 5, 9	<code>\multicolumn</code> , 20
<code>\addcontentsline</code> , 13	<code>\newcommand</code> , 35
<code>\and</code> , 13	<code>\newenvironment</code> , 35
<code>\atop</code> , 29	<code>\newline</code> , 8
<code>\author</code> , 13	<code>\newpage</code> , 8
<code>\begin</code> , 5	<code>\newtheorem</code> , 30
<code>\bibitem</code> , 14	<code>\noindent</code> , 8
<code>\caption</code> , 16, 32	<code>\nonumber</code> , 30
<code>\chapter</code> , 37	<code>\normalsize</code> , 11
<code>\choose</code> , 29	<code>\oint</code> , 25
<code>\cite</code> , 14	<code>\overbrace</code> , 27
<code>\colon</code> , 28	<code>\overline</code> , 27
<code>\date</code> , 13	<code>\pageref</code> , 16
<code>\documentclass</code> , 6, 14	<code>\pagestyle</code> , 7
<code>\emph</code> , 11	<code>\par</code> , 8
<code>\end</code> , 5	<code>\paragraph</code> , 13
<code>\footnote</code> , 16	<code>\part</code> , 13
<code>\footnotesize</code> , 11	<code>\prod</code> , 24
<code>\frac</code> , 24	<code>\qqquad</code> , 24
<code>\hfill</code> , 9	<code>\quad</code> , 24
<code>\hline</code> , 20	<code>\ref</code> , 16, 29, 32
<code>\hspace</code> , 8	<code>\resizebox</code> , 36
<code>\hspace*</code> , 9	<code>\right</code> , 25
<code>\huge</code> , 11	<code>\rotatebox</code> , 36
<code>\Huge</code> , 11	<code>\scalebox</code> , 36
<code>\include</code> , 34	<code>\scriptsize</code> , 11

<code>\section</code> , 13, 37	<code>\tableofcontents</code> , 14, 32
<code>\section*</code> , 13	<code>tabular</code> , 19, 26, 32
<code>\small</code> , 11	<code>\textbf</code> , 11
<code>\sqrt</code> , 24	<code>\textit</code> , 11
<code>\subparagraph</code> , 13	<code>\textrm</code> , 11, 26
<code>\subsection</code> , 13, 37	<code>\textsc</code> , 11
<code>\subsubsection</code> , 13	<code>\textsf</code> , 11
<code>\sum</code> , 24	<code>\textsl</code> , 11
<code>\tableofcontents</code> , 14, 32	<code>\texttt</code> , 11
<code>\textbf</code> , 11	<code>thebibliography</code> , 14
<code>\textit</code> , 11	<code>\theequation</code> , 36
<code>\textrm</code> , 11, 26	<code>\thispagestyle</code> , 7
<code>\textsc</code> , 11	<code>\tiny</code> , 11
<code>\textsf</code> , 11	<code>\title</code> , 13
<code>\textsl</code> , 11	<code>titlepage</code> , 14
<code>\texttt</code> , 11	<code>\today</code> , 13
<code>\theequation</code> , 36	
<code>\thispagestyle</code> , 7	<code>\underbrace</code> , 27
<code>\tiny</code> , 11	<code>\underline</code> , 27
<code>\title</code> , 13	<code>\usepackage</code> , 6
<code>\today</code> , 13	
<code>\underbrace</code> , 27	<code>\verb</code> , 11, 23
<code>\underline</code> , 27	<code>verbatim</code> , 12, 23
<code>\usepackage</code> , 6	<code>\vfill</code> , 9
<code>\verb</code> , 11, 23	<code>\vspace</code> , 8
<code>\vfill</code> , 9	<code>\vspace*</code> , 9
<code>\vspace</code> , 8	
<code>\vspace*</code> , 9	
<code>\scalebox</code> , 36	
<code>\scriptsize</code> , 11	
<code>\section</code> , 13, 37	
<code>\section*</code> , 13	
<code>\small</code> , 11	
<code>\sqrt</code> , 24	
<code>subequations</code> , 36	
<code>\subparagraph</code> , 13	
<code>\subsection</code> , 13, 37	
<code>\subsubsection</code> , 13	
<code>\sum</code> , 24	
<code>table</code> , 32	