

W miarę krótki i praktyczny kurs L^AT_EX-a w π^e minut

Ryszard Paweł Kostecki

rpkost [at] fuw.edu.pl
www.rysiq.prv.pl

18 maja 2006

Streszczenie

Tak, to prawda: masz przed sobą kurs wprowadzający w arkana systemu L^AT_EX. Kurs ten zrodził się w wyniku rocznej intensywnej pracy z L^AT_EX-em, w trakcie której musiałem zmierzyć się z publikowaniem najprzeróżniejszych rzeczy – od skryptu z mechaniki kwantowej, poprzez tomik poezji, kurs programowania, okładki na płyty kompaktowe, artykuł z fizyki jądrowej, aż na slajdach do prezentacji skończywszy. Aby profesjonalnie dopieścić te wszystkie publikacje, musiałem rozwiązać wiele problemów pod ogólnym hasłem “a jak to mogę zrobić?” oraz “a dlaczego to nie działa?”, przekopując się przez stertę manuali oraz dokumentacji i poszukując kluczowych informacji wśród morza szczegółów. Kurs ten opiera się w dużej mierze na problemach, które wystąpiły w praktyce i których pokonanie nie było trywialne. W tym właśnie – sędzę – kryje się wartość i potencjalna użyteczność tego tekstu – omówione są tu bowiem konkretne i sprawdzone rozwiązania, których odnalezienie w gąszczu dokumentacji wymaga czasem dużego nakładu sił i czasu.

Kurs ten początkowo miał mieć postać FAQ (tj. zbioru pytań i odpowiedzi). Uznałem jednakże, iż po dodaniu informacji o samym *języku* L^AT_EX, tekst ten może się stać dobrym wstępnym w miarę krótkim samouczkiem L^AT_EX-a nakierowanym na szybkie i praktyczne *użytkowanie* środowiska związanego z tym językiem. I tak też się stało.

Spis treści

0	Podstawowe informacje	2
0.1	Co to jest L ^A T _E X?	2
0.2	Jak należy wymawiać słowa T _E X i L ^A T _E X?	3
0.3	Czym L ^A T _E X różni się od T _E X-a?	3
1	W miarę krótkie omówienie języka L^AT_EX	3
1.1	Szkielet dokumentu	3
1.1.1	Deklaracja klasy dokumentu	3
1.1.2	Włączanie pakietów	4
1.1.3	Definicje globalne	4
1.1.4	Obszar dokumentu	4
1.1.5	Tytuł, autor i data	5
1.1.6	Rozdziały, sekcje, paragrafy	5
1.1.7	Spis treści	5
1.1.8	Bibliografia	6
1.1.9	Nagłówki i stopki	7
1.1.10	Przypisy	8
1.2	Formatowanie tekstu	8
1.2.1	Pogrubienie, pochylenie, podkreślenie, zmiana rozmiaru, koloru i stylu czcionki	8
1.2.2	Justowanie	9

1.2.3	Polecenia odpowiadające za znaki specjalne	9
1.2.4	Znaki diakrytyczne (ogólnie)	10
1.2.5	Polskie litery	10
1.2.6	Cyrylica	11
1.2.7	Język polski i cyrylica w tym samym tekście	12
1.2.8	Czcionki wektorowe i bitmapowe	12
1.3	Otoczenia, tabele i listy	12
1.4	Skład wyrażeń matematycznych	14
1.5	Rysunki	17
1.5.1	Włączanie do dokumentu rysunków i grafik z plików zewnętrznych	17
1.5.2	Rysowanie wewnątrz \LaTeX -a	17
1.5.3	Korzystanie z rysunków tworzonych przy pomocy Gnuplota	19
1.6	Odwołania	19
2	Praca z \LaTeX-em	19
2.1	Jak edytować pliki \LaTeX -a?	19
2.2	Jak obejrzeć pracę napisaną w \LaTeX -u?	20
2.3	Jak pracować ze środowiskiem \TeX -a/ \LaTeX -a pod Windows?	20
2.3.1	Praktyczne rozwiązanie polecane przez autora	20
2.4	Jak wykorzystać cały obszar kartki A4 lub A5?	21
2.5	Co robić z błędami w kodzie źródłowym?	21
3	Tworzenie dokumentów elektronicznych w różnych formatach plików	21
3.1	Tworzenie plików .ps (PostScript)	21
3.1.1	Tworzenie publikacji formatu A5 (i innych nietypowych)	22
3.2	Tworzenie plików .pdf (Adobe Acrobat)	22
3.2.1	ps2pdf	22
3.2.2	Bezpośrednia kompilacja pliku \LaTeX do dokumentu w formacie .pdf	22
4	Eksport/Import pomiędzy \LaTeX-em, Wordem, PowerPointem i innymi programami	23
4.1	Importowanie plików z Worda, PowerPointa i rozmaitych innych programów do \LaTeX -a	23
4.2	Eksportowanie wzorów napisanych w \LaTeX -u do Worda, PowerPointa i innych programów	24
4.3	Podobieństwo języków \LaTeX i HTML	24
4.3.1	Konwersja z \LaTeX do HTML	24
5	Specyficzne problemy praktyczne	24
5.1	Problemy z Bib \TeX -em	24
5.1.1	Bib \TeX a sprawa polska	24
5.1.2	Konflikt Bib \TeX -a z poleceniem <code>\textcompwordmark{}</code>	25
5.2	Problemy z rozmieszczeniem tekstu	25
5.2.1	Niektóre fragmenty tekstu wystają poza margines, lub, co gorsza, poza kartkę! Co robić?	25
5.2.2	Rysunki są niepokorne i wciąż mi je \LaTeX przerzuca względem tekstu! Co robić?	25
5.2.3	Czy jest jakaś możliwość szybkiego projektowania wyglądu dokumentów \LaTeX -owych, na przykład tak jak w Wordzie?	25
5.2.4	A jak został sformatowany ten dokument?	25
5.2.5	A jakiś prosty przykład?	26

0 Podstawowe informacje

0.1 Co to jest \LaTeX ?

\LaTeX jest systemem przeznaczonym do składania tekstu, tj. do tworzenia potencjalnie dowolnych publikacji opartych na tekście jako podstawowym nośniku informacji. W przeciwieństwie do innych systemów – takich jak Word, czy (z bardziej profesjonalnych) QuarkXPress – w \LaTeX -u nie projektuje się ostatecznego wyglądu publikacji metodą edycji połączonej z jednoczesnym podglądem ostatecznego efektu pracy (tzn. WYSIWYG – ang. *what you see is what you get*), lecz poprzez logiczny skład dokumentu. Pliki źródłowe \LaTeX -a stają się dokumentami dopiero po skompilowaniu (przetworzeniu) i odczytaniu w odpowiednim do tego programie. Ogólnie rzecz biorąc, można powiedzieć, iż \LaTeX przypomina w swej składni język HTML, zaś w sposobie przetwarzania C++ (lub raczej Javę).

0.2 Jak należy wymawiać słowa \TeX i \LaTeX ?

Tradycją w pracach na temat \TeX -a i \LaTeX -a stało się wyjaśnienie, iż nazwy te czytać należy wymawiając literę ‘X’ nie jako ‘iks’, lecz jako ‘ha’ (\TeX pochodzi od słowa ‘technical’, a nie od nazwy materiału tekstylnego).

0.3 Czym \LaTeX różni się od \TeX -a?

Można powiedzieć, że \LaTeX jest językiem wyższego rzędu niż \TeX , jakkolwiek w swym wnętrzu \LaTeX oparty jest właśnie na \TeX -u (w istocie \LaTeX to po prostu system makr \TeX -a). Dzięki temu ten sam efekt zajmujący wiele linii w \TeX -u można uzyskać przez jedną linijkę w \LaTeX -u. Ceną jaką się za to ponosi jest niemożność wykonania w \LaTeX -u wielu operacji na tekście, które w czystym \TeX -u są możliwe (chyba najbardziej irytującą jest niemożność użycia czcionki o dowolnym rozmiarze). A w \TeX -u z tekstem można zrobić (podobnie) wszystko.

1 W miarę krótkie omówienie języka \LaTeX

Plik \LaTeX -owy to po prostu plik tekstowy zawierający, oprócz tekstu, polecenia formatujące ten tekst. Język \LaTeX jest podobny do HTML – bowiem w obydwu tak samo (przy pomocy poleceń umieszczonych razem z tekstem w źródłowym pliku tekstowym) określa się to, w jaki sposób tekst ma być sformatowany. W \LaTeX -u można, tak samo jak w HTML, podzielić polecenia na obejmujące obustronnie fragment tekstu oraz na występujące w konkretnym miejscu. Polecenia obejmujące w \LaTeX -u mają składnię o postaci:

```
\begin{polecenie}
  jakiś tekst
\end{polecenie}
```

W takiej sytuacji polecenia zwie się *otoczeniami*. Polecenia mogą mieć również postać: `\innepolecenie{ jakiś tekst }`, lub po prostu `\prostepolecenie`.

1.1 Szkielet dokumentu

1.1.1 Deklaracja klasy dokumentu

Dokument \LaTeX zaczyna się linią deklaracji klasy dokumentu: `\documentclass[opcje]{klasa_dokumentu}`, gdzie *klasa_dokumentu* musi być jedną z następujących: `article` (artykuł), `report` (raport), `book` (książka), `letter` (list), `slides` (folie projekcyjne)¹, zaś *opcje* mogą składać się z kilku elementów (rozdzielonych przecinkami) spośród różnych wierszy poniższej tabelki:

typ opcji	możliwe opcje
rozmiar podstawowej czcionki	10pt, 11pt, 12pt
rozmiar papieru	letterpaper, legalpaper, executivepaper, a4paper, a5paper, b5paper
typ wyjściowej publikacji	draft, final
liczba stron na kartce	oneside, twoside
liczba kolumn na stronie	onecolumn, twocolumn
osobna strona tytułowa	titlepage, notitlepage
inne opcje	landscape, leqno, fleqno, openbib, openright, openany, clock

Opcja `openright` wymusza rozpoczynanie kolejnych rozdziałów wyłącznie od stron nieparzystych. Opcja `openany` zezwala na rozpoczynanie nowego rozdziału od dowolnej strony. Opcja `openbib` powoduje, że przy tworzeniu spisu literatury (co będzie omówione później) za pomocą polecenia `thebibliography` drugi i następnne wiersze każdej pozycji rozpoczynają się nie na wysokości początku pierwszej wiersza opisu, lecz

¹Klasa `slides` służy do tworzenia prezentacji, ale lepiej tworzyć je w \LaTeX -u przy pomocy wyspecjalizowanych pakietów. Popularnym pakietem służącym do tego celu jest pakiet `texpower`.

z wcięciem równym wartości parametru długości `\bibindent`. Opcja `clock` umożliwia szacowanie czasu potrzebnego do wygłoszenia prezentacji (patrz: [1, str. 108]).

Każda klasa dokumentu ma swoje domyślne ustawienia opcji. Można w ogóle nie podawać żadnych opcji. Nie wszystkie opcje można ze sobą łączyć, jak również nie wszystkie opcje są dostępne dla wszystkich klas dokumentów:

klasa	opcje domyślne
article	10pt, letterpaper, final, oneside, onecolumn, notitlepage
report	10pt, letterpaper, final, oneside, onecolumn, openany, titlepage
book	10pt, letterpaper, final, twoside, onecolumn, openright, titlepage
letter	letterpaper, final, oneside
slides	letterpaper, final, titlepage
klasa	opcje niedostępne
article	openright, openany, clock
report	clock
book	clock
letter	openright, openany, openbib, clock, onecolumn, twocolumn, (no)titlepage
slides	twocolumn, openright, openany, openbib

Warto przy okazji zwrócić uwagę na to, iż argumenty konieczne w \LaTeX -u podaje się w nawiasach klamrowych, zaś argumenty opcjonalne – w nawiasach kwadratowych.

Przykładowymi poprawnymi deklaracjami klasy dokumentu są więc:

```
\documentclass[10pt,a5paper,final,twoside,fleqn,onecolumn]{book},
\documentclass{article}.
```

1.1.2 Włączanie pakietów

Po deklaracji klasy dokumentu włącza się potrzebne pakiety dodatkowe. Pakiety to zbiory makr \TeX -a i \LaTeX -a, które ułatwiają pracę i zwiększają możliwości dostępne użytkownikowi z poziomu kodu źródłowego. Pakiety, tak jak pliki nagłówkowe w C/C++, konieczne trzeba włączyć na początku dokumentu. Praca z pakietami jest bardzo podobna do pracy z bibliotekami w C/C++ również dlatego, że w obu przypadkach istnieją pliki tekstowe o określonym rozszerzeniu, będące “wrotami” do bibliotek (w C/C++ są to pliki z rozszerzeniem `.h`, w \LaTeX -u są to pliki `.sty`).

Przykładowo:

```
\usepackage[cp1250]{inputenc} włącza pakiet obsługi plików źródłowych z polskimi znakami napisanymi pod Windows (strona kodowa CP 1250),
\usepackage{times} włącza pakiet wektorowych zawierający czcionki Helvetica, Times i Courier.
```

1.1.3 Definicje globalne

Po włączeniu pakietów można pokusić się o wprowadzenie rozmaitych definicji globalnych dotyczących całego dokumentu. Na przykład:

```
\setlength{\textheight}{24.4cm} ustawi wysokość obszaru przeznaczonego na kartce na dokument na 24.4cm,
\setlength{\parindent}{0mm} ustawi rozmiar wcięcia na początku każdego akapitu na 0mm,
\setlength{\parskip}{4mm} ustawi rozmiar odstępu pomiędzy kolejnymi akapitami na 4mm,
\newcommand{\bs}{\backslash} zdefiniuje nową komendę \bs, której użycie spowoduje wyświetlenie w tekście znaku backslasha, czyli ‘\’.
```

1.1.4 Obszar dokumentu

Po nagłówku znajduje się już właściwy obszar dokumentu. Zawarty on jest pomiędzy dwoma linijkami, wewnątrz których znajduje się cała treść dokumentu:

```
\begin{document}

\end{document}
```

1.1.5 Tytuł, autor i data

Kiedy tworzymy pracę bądź artykuł, zwykle chcemy na początku pracy umieścić jej tytuł, autora i datę. Służą do tego polecenia: `\title{}`, `\author{}`, oraz `\date{}` (w nawiasach klamrowych należy oczywiście podać odpowiednie napisy). Jeśli nie zostanie użyte polecenie `\date{}`, \LaTeX sam wygeneruje aktualną datę. Po tych poleceniach należy użyć polecenia `\maketitle` w celu utworzenia tak skonstruowanego tytułu.

Jeśli jednak w opcjach klasy dokumentu zadeklarowane został tryb `titlepage` (czyli utworzenia osobnej strony tytułowej), to (oprócz polecenia `\maketitle`) zaleca się polecenia `\title{}`, `\author{}` oraz ewentualnie `\date{}` ująć w otoczenie `titlepage`, np:

```
\begin{titlepage}
\title{Niepodległość Trójkątów}
\author{Stanisław Ignacy Witkiewicz}
\date{2 II 1921}
\maketitle
\end{titlepage}
```

Można oczywiście nie korzystać w ogóle z poleceń `\title{}`, `\author{}`, `\date{}` i stworzyć stronę tytułową (lub napis tożsamy tytułowi) dowolnie według własnych upodobań.

1.1.6 Rozdziały, sekcje, paragrafy

Niniejsza praca jest przykładem dokumentu utworzonego w \LaTeX -u, używającego podziału na sekcje i paragrafy. Do tworzenia coraz bardziej zanieżdżonych paragrafów korzysta się kolejno z poleceń:

```
\section{}
\subsection{}
\subsubsection{}
\paragraph{}
\subparagraph{}
```

Kolejny poziom zagnieżdżenia wyrażony zostaje kolejną kropką z numerem porządkowym, rosnącym od liczby 1. W celu utrzymania sekcji nienumerowanych należy skorzystać z polecenia `\section*{}`. W nawiasach klamrowych należy oczywiście podać tytuły sekcji.

Dla klasy dokumentu `\book` istnieje jeszcze jeden poziom zagnieżdżenia – `chapter` – nadrzędny wobec wszystkich innych. Jego użycie spowoduje utworzenie kolejnej strony parzystej (co można oczywiście zmienić opcją `openany` w poleceniu `\documentclass[...]{...}`), zaczynającej się do tytułu *Rozdział N*, gdzie *N* jest numerem rozdziału (o ile do dokumentu zostanie włączony pakiet `babel` z opcją `polish` – w przeciwnym razie napis na początku *N*-tego rozdziału brzmiał będzie *Chapter N*; patrz paragraf 1.2.5).

Domyślnie numerowane są tylko rozdziały, sekcje i podsekcje, zaś paragrafy i podparagrafy – nie. Można to zmienić poleceniem `\setcounter{secnumdepth}{x}`, gdzie zamiast *x* należy wstawić wartość od 0 (jeśli numerowane mają być tylko rozdziały) do 5 (jeśli numerowane mają być również kolejne `subparagraph`). Numery porządkowe sekcji czy rozdziału można zmieniać do woli. Np. polecenie `\setcounter{section}{-1}` spowoduje że *następny* rozdział będzie miał numer porządkowy 0.

1.1.7 Spis treści

Spis treści generowany jest przez użycie polecenia `\tableofcontents` w miejscu, w którym spis treści ma wystąpić. Aby spis treści był utworzony w dokumencie poprawnie, trzeba dokument przetworzyć \LaTeX -em conajmniej dwukrotnie.

1.1.8 Bibliografia

Bibliografię, czyli spis literatury powiązanej z tekstem, można utworzyć w \LaTeX -u na dwa różne sposoby przedstawione niżej. Niezależnie od tego w jaki sposób utworzona zostanie bibliografia, odwoływać się do poszczególnych jej pozycji można w ten sam sposób – poprzez użycie w tekście polecenia `\cite{identyfikator_książki}`.

Prosty sposób wewnątrz pliku

1. Określenie stylu bibliografii poleceniem `\bibliographystyle{styl}`. Styl może być jednym z następujących: `plain`, `unsrt`, `abbrv`, `alpha` (szczegółowe omówienie każdego z tych stylów można znaleźć w pracach [1] oraz [8]). Polecenie to najlepiej umieścić bezpośrednio po poleceniu `\begin{document}`.
2. W przeznaczonym na bibliografię miejscu wypisanie bibliografii korzystając z otoczenia `\thebibliography`. Przykładowy kod bibliografii skonstruowanej tą metodą może wyglądać następująco:

```
\begin{thebibliography}{3}
\bibitem{diller} Antoni Diller, \textit{\LaTeX\ wiersz po wierszu},
wydawnictwo Helion, Gliwice 2001
\bibitem{grfguide} D.P. Carlisle, \textit{Packages in the 'graphics'
bundle}
\bibitem{lshort} Tobias Oetiker, \textit{The Not So Short Introduction
To \LaTeX2e}
\end{thebibliography}
```

3. Ewentualne zamieszczenie odnośników do bibliografii w odpowiednich miejscach w dokumencie przez użycie polecenia `\cite[opcjonalne_informacje]{identyfikator_książki}`. *Opcjonalne_informacje* mogą zawierać np. numer strony, z której pochodzi cytat. *Identyfikator_książki* w powyższym przykładzie to `diller`, `grfguide` oraz `lshort`.
4. Przetworzenie pliku źródłowego \LaTeX -em conajmniej dwukrotnie.

Bib \TeX

1. Określenie stylu bibliografii poleceniem `\bibliographystyle{styl}`. Styl może być jednym z następujących: `plain`, `unsrt`, `abbrv`, `alpha` (szczegółowe omówienie każdego z tych stylów można znaleźć w pracach [1] oraz [8]). Polecenie to najlepiej umieścić bezpośrednio po poleceniu `\begin{document}`.
2. Umieszczenie polecenia `\bibliography` w miejscu, w którym ma znajdować się bibliografia.
3. Utworzenie pliku bibliografii o rozszerzeniu `.bib`, a o nazwie takiej samej jak nazwa pliku ze źródłem \LaTeX -owego dokumentu (szczegóły: patrz dalej).
4. Ewentualne zamieszczenie odnośników do bibliografii w odpowiednich miejscach w dokumencie przez użycie polecenia `\cite[opcjonalne_informacje]{identyfikator_książki}`. *Opcjonalne_informacje* mogą zawierać np. numer strony, z której pochodzi cytat.
5. Przetworzenie pliku źródłowego najpierw \LaTeX -em, następnie Bib \TeX -em, po czym jeszcze dwa razy \LaTeX -em, np. poprzez wydanie poleceń:

```
latex rozważania_o_witkacym
bibtex rozważania_o_witkacym
latex rozważania_o_witkacym
latex rozważania_o_witkacym
```

Plik bibliografii `.bib` jest oczywiście plikiem tekstowym i składa się z rekordów o przykładowej postaci:

```
@book{haken:atomy:kwanty,
address = "Warszawa",
author = "Haken, Herman and Wolf, Hans Christoph",
edition = "drugie, zmienione",
isbn = "83-01-13626-X",
publisher = "Wydawnictwo Naukowe PWN",
note = "Tłumaczenie: Irena Deperasińska, Jerzy Prochorow",
title = "Atomy i kwanty. Wprowadzenie do współczesnej spektroskopii atomowej",
```

```
year = 2002
}
```

Napis `haken:atomy:kwanty` jest właśnie *identyfikatorem książki*. Identyfikator może mieć w nazwie dwukropkę, lecz nie musi. Nie wszystkie pola wewnątrz rekordu są konieczne, jakkolwiek istnieje zwyczajowa zasada zapisywania w pliku bibliografii maksymalnej ilości dostępnych informacji o książce. Istnieje kilka typów rekordów. Odpowiadają one różnym typom publikacji: `@article`, `@book`, `@booklet`, `@conference`, `@inbook`, `@incollection`, `@inproceedings`, `@manual`, `@masterthesis`, `@phdthesis`, `@proceedings`, `@misc`, `@techreport`, `@unpublished`. Każdy z typów ma swój obsługiwany zestaw pól. Szczegółowe informacje znajdują się w pracach [1, str. 73-82] i [8].

Korzystając z $\text{Bib}\TeX$ -a napotkać można na kilka problemów. Omówienie ich znajduje się w paragrafie 5.1.

1.1.9 Nagłówki i stopki

Do obsługi nagłówka i stopki powszechnie korzysta się z pakietu `fancyheaders`, który należy dołączyć do dokumentu poleceniem:

```
\usepackage{fancyhdr}.
```

Proste użycie `fancyheaders` przedstawione jest poniżej. Linijki te należy umieścić bezpośrednio po poleceniu `\begin{document}`:

```
\pagestyle{fancy}
\rhead{Prawa strona nagłówka. Strona nr \thepage}
\chead{Środkowy nagłówek}
\lhead{Lewy nagłówek}
\lfoot{Lewa stopka}
\cfoot{Środkowa stopka}
\rfoot{Prawa stopka}
```

Jeśli korzystamy z klasy dokumentu `book` z opcją dwustronną `twoside`, przyda się nam rozróżnienie między stronami lewymi a prawymi. Wtedy zamiast powyższych linijek warto użyć podobnych do następujących:²

```
\pagestyle{fancy}
\fancyhead{}
\fancyhead[LE,RO]{\textsf{W miarę krótki i praktyczny kurs \LaTeX-a w $\pi^e$ minut}}
\fancyhead[RE,LO]{\ }
\fancyhead[CE,CO]{\ }
\fancyfoot[LE,RO]{\textbf{\textsf{\thepage}}}
\fancyfoot[RE,LO]{\ }
\fancyfoot[CE,CO]{\ }
```

L oznacza tu lewą stronę, R – prawą stronę, C – środek strony, zaś O i E określają których stron ma polecenie dotyczyć. Strony O to strony prawe (dla dokumentu `twoside`) lub wszystkie (dla dokumentu `oneside`), zaś strony E to strony lewe (dla dokumentu `twoside`) lub żadne (dla dokumentu `oneside`). Jeśli chcemy, aby na każdej stronie stopka i nagłówek były oddzielone linią, należy do powyższych linijek dopisać:

```
\renewcommand{\headrulewidth}{0.4pt}
\renewcommand{\footrulewidth}{0.4pt}
```

W celu uzyskania braku nagłówka i stopki na którejś ze stron, wystarczy na tej stronie napisać komendę `\thispagestyle{empty}`. Gdy zaś w ogóle nie chcemy żadnych nagłówek, wystarczy nie włączać pakietu `fancyhdr`, a po `\begin{document}` napisać polecenie

```
\pagestyle{empty}.
```

²Polecenia `\textsf` i `\textbf` oznaczają odpowiednio użycie czcionki bezszeryfowej i pogrubionej (będzie o tym na następnej stronie), polecenie `\thepage` daje w efekcie numer bieżącej strony.

Na każdej stronie dokumentu można zmienić którykolwiek z elementów stopki/marginesu, wydając na tej stronie polecenie na przykład takie: `\fancyfoot[CE,RE,LE]{!! Uwaga! Dzikie jeże !!}`. Puste polecenie `\fancyhead{}` czyści nagłówek i stopkę.

1.1.10 Przypisy

Przypisy³ uzyskuje się poleceniem `\footnote{...}`.

Polecenie `\thepage` zwraca bieżący numer strony. Wskazania automatycznego licznika numeru stron można przesuwać na dowolnej stronie. Rezultat przesunięcia przejawiają się począwszy od strony *następnej*. Przykładowo zmiana numeru strony wydana na stronie 4 poleceniem `\setcounter{page}{38}` spowoduje, że w otrzymanym dokumencie numeracja stron będzie następująca: 1, 2, 3, 4, 39, 40, 41,...

1.2 Formatowanie tekstu

1.2.1 Pogrubienie, pochylenie, podkreślenie, zmiana rozmiaru, koloru i stylu czcionki

efekt	polecenie	przykład	kod przykładu
pogrubienie	<code>\textbf{}</code>	zgubne frytki	<code>\textbf{zgubne frytki}</code>
normalna grubość	<code>\textmd{}</code>	zgubne frytki	<code>\textmd{zgubne frytki}</code>
kursywa	<code>\textit{}</code>	<i>zgubne frytki</i>	<code>\textit{zgubne frytki}</code>
pochylenie	<code>\textsl{}</code>	<i>zgubne frytki</i>	<code>\textsl{zgubne frytki}</code>
podkreślenie	<code>\underline{}</code>	<u>zgubne frytki</u>	<code>\underline{zgubne frytki}</code>
pismo proste	<code>\textup{}</code>	zgubne frytki	<code>\textup{zgubne frytki}</code>
kapitaliki	<code>\textsc{}</code>	ZGUBNE FRYTKI	<code>\textsc{zgubne frytki}</code>
czcionka o stałej szerokości	<code>\texttt{}</code>	zgubne frytki	<code>\texttt{zgubne frytki}</code>
czcionka bezszeryfowa	<code>\textsf{}</code>	zgubne frytki	<code>\textsf{zgubne frytki}</code>
czcionka typu antykw	<code>\textrm{}</code>	zgubne frytki	<code>\textrm{zgubne frytki}</code>

Polecenia te można oczywiście łączyć, zagnieżdżając jedno w drugim (np. `\textbf{\textsf{Baltazar Gąbka}}`). Rozmiar czcionki może być jednym z następujących:

rozmiar	przykład	kod przykładu
<code>tiny</code>	<small>Miłorzęby</small>	<code>\tiny{Miłorzęby}</code>
<code>scriptsize</code>	<small>Miłorzęby</small>	<code>\scriptsize{Miłorzęby}</code>
<code>footnotesize</code>	<small>Miłorzęby</small>	<code>\footnotesize{Miłorzęby}</code>
<code>small</code>	<small>Miłorzęby</small>	<code>\small{Miłorzęby}</code>
<code>normalsize</code>	Miłorzęby	<code>\normalsize{Miłorzęby}</code>
<code>large</code>	<big>Miłorzęby</big>	<code>\large{Miłorzęby}</code>
<code>Large</code>	Miłorzęby	<code>\Large{Miłorzęby}</code>
<code>LARGE</code>	Miłorzęby	<code>\LARGE{Miłorzęby}</code>
<code>huge</code>	Miłorzęby	<code>\huge{Miłorzęby}</code>
<code>Huge</code>	Miłorzęby	<code>\Huge{Miłorzęby}</code>

Są to jedyne dostępne w L^AT_EX-u rozmiary czcionek, niestety (jest to jedna z jego największych wad). Domyślną czcionką dokumentu jest czcionka o rozmiarze normalnym (`normalsize`). W zależności od opcji użytej w deklaracji klasy dokumentu (`10pt`, `11pt` lub `12pt`), wszystkie czcionki mogą być nieco mniejsze lub nieco większe. Także sama klasa dokumentu ma wpływ na rodzaj domyślnych i dostępnych w dokumencie czcionek (charakterystycznym i osobliwym przykładem tego zjawiska jest klasa dokumentu `slides`).

Kolorowe czcionki uzyskać można po dołączeniu pakietu `graphicx` poleceniem

```
\usepackage{graphicx}.
```

Wówczas tekst o określonym kolorze uzyskuje się poleceniem

³Nie należy mylić przypisu (takiego jak ten) ze stopką – jak widać zresztą są to dwa różne obiekty. Przypis jest zawsze tworzony *wewnątrz* obszaru dokumentu, natomiast stopka znajduje się poniżej tego obszaru.


```
\textcolor{nazwa_koloru}{tekst_do_pokolorowania},
```

zaś tekst o określonym kolorze tła otrzymuje się poleceniem

```
\colorbox{nazwa_koloru}{tekst_do_pokolorowania_tla},
```

gdzie w obydwu przypadkach *nazwa_koloru* jest nazwą w języku angielskim (choć również – w trochę inny sposób – można podawać nazwy kolorów w systemie RGB), np. `blue` czy też `green`. Kolory można podawać również w kodzie RGB, przy pomocy polecenia `\color[RGB]`, pisząc na przykład `\color[RGB]{255,214,76}` {kolorowe kredki}. Więcej szczegółów dotyczących kolorowania tekstu znajduje się w manualu pakietu `graphics/graphicsx` (patrz [13]). Ciekawe informacje na temat pakietu `colortbl` przeznaczonego specjalnie do kolorowania tabelek znajdują się w pliku [14].

1.2.2 Justowanie

\LaTeX automatycznie justuje tekst obustronnie. Justowanie do prawej uzyskuje się, umieszczając wybrany fragment tekstu w otoczeniu `flushright`:

```
\begin{flushright}
...
\end{flushright}
```

Analogicznie za justowanie do lewej odpowiada otoczenie `flushleft`, zaś za centrowanie – otoczenie `center`.

1.2.3 Polecenia odpowiadające za znaki specjalne

Podstawowym poleceniem generującym znak specjalny jest oczywiście polecenie przejścia do następnej linii – odpowiednik polecenia `
` znanego z HTML-a. Polecenie to brzmi:

```
\\
```

Zamiast tego polecenia można się posłużyć poleceniem `\newline` (czyniącym to samo), lub poleceniem `*`, zabraniającym przejścia w nowej linii do nowej strony. Wymuszenie przejścia do nowej strony uzyskuje się poleceniem `\newpage`, lub `\clearpage` (to ostatnie powoduje też wyczyszczenie bufora pamięci rysunków, poprzez wymuszenie narysowania ich na dotychczasowej stronie⁴). Aby uzyskać kilka czystych stron z rzędu nie wystarczy połączyć poleceń `\newpage\newpage`, bowiem \LaTeX zignoruje drugie z nich. Trzeba zatem na nowej stronie utworzyć niewidzialny znak – spację, lub jeszcze lepiej dwa niewidzialne znaki – spację i nową liniijkę:

```
\newpage\\\_\newpage
```

Przy okazji okazało się, że spację można utworzyć komendą `\` . Oczywiście, spację można też uzyskiwać normalnie, przy czym trzeba uważać, bowiem \LaTeX , tak samo jak HTML, traktuje dwie spacje położone obok siebie jako jedną. Warto tutaj zaznaczyć, że inaczej jest w przypadku znaków przejścia do nowej linii: jeden `<ENTER>` w kodzie źródłowym zostanie przez \LaTeX zignorowany, lecz dwa `<ENTER>`-y zostaną potraktowane jako znak nowego akapitu, zatem tak, jak użytkownik by oczekiwał.

\TeX oraz \LaTeX używają niektórych znaków obecnych na klawiaturze do specjalnych celów.

Znak specjalny `%` służy do umieszczania komentarzy w tekście źródłowym. Jego działanie jest analogiczne do działania znaku `' ; '` w assemblerze, czy też polecenia `'//'` w C++ (cały wiersz, począwszy od tego znaku, jest ignorowany przy przetwarzaniu).

Znak specjalny `~` użyty w tekście jest traktowany jako niepodzielna spacja (aby uzyskać niepodzielny dłuższy fragment tekstu, należy objąć go poleceniem `\mbox{...}`).

Znak specjalny `_` użyty w trybie matematycznym (szerzej o tym trybie w paragrafie 1.4) generuje indeks dolny, zaś znak specjalny `^` generuje indeks górny.

⁴Jest to polecenie analogiczne do C++'owego manipulatora strumienia `endl`, który to spełnia dwie funkcje: `'\n'` oraz `flush`.

Znak specjalny " służy do tworzenia znaków diakrytycznych (patrz paragraf 1.2.5). W celu otrzymania cudzysłowów podwójnych należy wpisać następujące po sobie dwa pojedyncze cudzysłowy i/lub przecinki (w zależności od potrzeb) albo po prostu skorzystać z pojedynczych cudzysłowów.

kod	efekt
<code>``żlebotron''</code>	“żlebotron”
<code>,,żlebotron,,</code>	”żlebotron”
<code>`żlebotron'</code>	'żlebotron'
<code>'żlebotron'</code>	'żlebotron'

Aby otrzymać w dokumencie któryś ze znaków używanych przez \LaTeX jako znak specjalny, należy posłużyć się poniższą tabelką:

znak do otrzymania	polecenie
~	<code>\ensuremath{\sim}</code>
#	<code>\#</code>
\$	<code>\\$</code>
%	<code>\%</code>
^	<code>{\textasciicircum}</code>
&	<code>\&</code>
_	<code>_</code>
	<code>{\textbar}</code>
\	<code>{\textbackslash}</code>
"	<code>{\textquotedbl}</code>
{	<code>\{</code>
}	<code>\}</code>

1.2.4 Znaki diakrytyczne (ogólnie)

Paragraf ten opracowany jest na podstawie tabeli na stronie 23 książki [1].

nazwa znaku diakrytycznego	przykład	kod przykładu
<i>aigu (acute)</i>	á	<code>\' {a}</code>
<i>breve</i>	ĸ	<code>\u {b}</code>
kółko	ĉ	<code>\r {c}</code>
daszek (<i>circumflex</i>)	Ď	<code>\^ {d}</code>
dierезa (umlaut)	ë	<code>\" {e}</code>
dierезa węgieska	ř	<code>\H {f}</code>
kropka	ġ	<code>\. {g}</code>
<i>grave</i>	ĥ	<code>\' {h}</code>
<i>háček</i>	ř	<code>\v {q}</code>
makron	ī	<code>\= {r}</code>
tylda	ķ	<code>\~ {k}</code>
podkreślenie	l̄	<code>\b {l}</code>
<i>cédille</i>	ŋ	<code>\c {m}</code>
kropka-pod	ŋ	<code>\d {n}</code>
znak ligatury	op̄	<code>\t {op}</code>

Polecenia `\i` oraz `\j` tworzą litery *i* oraz *j* pozbawione kropek. Polecenie to jest przydatne na przykład wówczas gdy chcemy napisać nazwisko Naĩmark (w tej sytuacji korzysta się z połączenia `{\u{\i}}`).

1.2.5 Polskie litery

Problem polskich liter w tekście jest w istocie dwoma problemami: problemem poprawnej interpretacji polskich znaków diakrytycznych podanych w rozszerzonym kodzie ASCII z klawiatury (np. poprzez wciśnięcie `<ALT>+A`), oraz problemem poprawnego wyświetlenia ich na ekranie monitora lub/i na wydruku.

Aby polskie znaki ukazały się w sposób satysfakcjonujący w dokumencie wyjściowym, do kodu źródłowego trzeba włączyć pakiet `fontenc` w następujący sposób:

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

Parametr [T1] określa jaki zestaw czcionek ma być użyty przy tworzeniu dokumentu. Spotyka się również używanie innych zestawów, np. OT4, lecz autor niniejszej pracy korzysta wyłącznie z T1 i ma się dobrze.

Aby móc korzystać z polskich znaków wpisywanych z klawiatury przy użyciu klawisza <ALT>, trzeba dołączyć pakiet obsługujący odpowiednią stronę kodową. Dla Windows będzie to strona kodowa cp1250:

```
\usepackage[cp1250]{inputenc},
```

zaś dla Unixa ISO-8859-2, zwana też Latin2:

```
\usepackage[latin2]{inputenc}.
```

Po tych zabiegach należy jeszcze włączyć dwa pakiety, w celu ostatecznej i całkowitej polonizacji:

```
\usepackage[english,polish]{babel}
\usepackage{polski}
```

Pakiet babel w liście opcji oczekuje języków, z których zamierzamy skorzystać w trakcie pisania tekstu, przy czym położony najdalej na prawo język uznaje za język podstawowy pracy.

Jeśli chcemy mieć możliwość wygodnej pracy nad jednym kodem źródłowym zarówno pod Unixem, jak i pod Windows, to warto zrezygnować z korzystania z polskich znaków otrzymywanych poprzez wciśnięcie klawisza <ALT>+znak, bowiem, z powodu różnych stron kodowych na różnych systemach, w jednym z nich będziemy widzieć tzw. “krzaki”.

Uwaga: spotyka się czasem korzystanie tylko z pakietu babel lub tylko z pakietu polski. Jednakże dopiero włączenie ich obydwu umożliwi poprawne korzystanie z wielu języków w jednym dokumencie (np. do wygenerowania dwujęzycznych streszczeń), co zapewnia pakiet babel, jak i poprawne łamanie wyrazów zgodnie z polską gramatyką, co zapewnia pakiet polski.

Wygodną metodą w takiej sytuacji jest (oczywiście poza ciągłymi konwersjami pliku) zapisywanie polskich znaków w jeden z następujących sposobów:

sposób #1	"a	"c	"e	"l	"o	"n	"s	\.z	"z
efekt #1	ą	ć	ę	ł	ó	ń	ś	ż	ź
sposób #2	\k{a}	\'c	\k{e}	{\l}	\'o	\'n	\'s	\.z	\'z
efekt #2	ą	ć	ę	ł	ó	ń	ś	ż	ź

I analogicznie dla dużych liter. Jak widać, obydwa sposoby dają te same rezultaty, przy czym – co ważne – można je stosować jednocześnie.

1.2.6 Cyrylica

W życiu każdego poety przychodzi taka chwila, kiedy musi zacząć pisać cyrylicą. Wówczas sięga po \LaTeX -a i... okazuje się, że pisanie w nim cyrylicą jest dziecinnie proste.⁵

Jeżeli w systemie operacyjnym jest już zainstalowana i uruchomiona odpowiednia strona kodowa, to wystarczy włączyć w pliku źródłowym pakiet babel z odpowiednią opcją – odpowiadającą występującemu językowi:

```
\usepackage[english,russian]{babel}
```

Znaki cyrylicy można w takiej sytuacji uzyskiwać w sposób naturalny – w Windows przełączając klawiaturę, na Linuxie wydając polecenie setxkbmap ru.

⁵Tak było, dopóki pracowałem pod Linuxem, tudzież Windowsami 95 (OSR2). W Windows XP pojawił się problem, gdyż system ten zbyt inteligentnie koduje wprowadzane z klawiatury znaki w cyrylicy (tzn. zapisuje je – nawet Notepad – w Unikodzie, zamiast w kodzie 256-znakowym pod stroną CP-1251). Prosty, skuteczny i szybki sposób ominięcia tych problematycznych ‘udogodnień’ jeszcze nie jest mi znany. Można jednak zawsze skorzystać z OpenOffice (który przy zapisie cyrylicy proponuje różne strony kodowe do wyboru) lub z Worda, zapisując tekst jako plik HTML i wybierając kodowanie ‘Cyrylica’, a następnie tak utworzony plik otworzyć w Notatniku by wyciąć z niego cyrylicę do schowka (spod Explorera zrobić się tego z zachowaniem kodowania CP-1251 nie da, bo jest zbyt ‘inteligentny’).

1.2.7 Język polski i cyrylica w tym samym tekście

Aby móc jednocześnie korzystać z języka polskiego (a ściślej mówiąc: z polskich znaków diakrytycznych) oraz cyrylicy (czyli najczęściej języka rosyjskiego) w tym samym dokumencie należy do dokumentu włączyć następujące pakiety:

```
\usepackage[russian,english,polish]{babel}
\usepackage{polski}
\usepackage[T1]{fontenc}
```

Wówczas polskie litery można uzyskiwać korzystając z dwóch sposobów omówionych nieco wyżej, zaś rosyjskie litery uzyskuje się poprzez przełączenie strony kodowej na język rosyjski.

1.2.8 Czcionki wektorowe i bitmapowe

Domyślnie \LaTeX korzysta z czcionek bitmapowych. Są to czcionki, które przy przeskalowaniu wyglądają kanciasto. Ujawnia się to okrutnie zwłaszcza przy konwersji `.tex` \rightarrow `.dvi` \rightarrow `.ps` \rightarrow `.pdf`. W takim przypadku czcionki w końcowym dokumencie `.pdf` są strasznie poszarpane.

Aby uniknąć takiej sytuacji, warto skorzystać z jednego z pakietów czcionek wektorowych, np. `times`:

```
\usepackage{times}.
```

Niniejszy dokument jest napisany właśnie z wykorzystaniem tego pakietu. Jest wiele pakietów czcionek wektorowych. Podstawowe znajdują się w katalogu `.../texmf/tex/latex/psnfss` i są to między innymi: `antiqua`, `avant`, `bookman`, `chancery`, `charter`, `courier`, `grosesq`, `helvet`, `lubright`, `lucida`, `newcent`, `nimbus`, `palatino`, `utopia` i oczywiście `times`. Warto zaznaczyć, iż nie wszystkie typy czcionek mają wsparcie polskich znaków diakrytycznych oraz wszystkich trzech postawowych stylów pisma: `\textrm{}`, `\textsf{}` i `\texttt{}`.

Chociaż, z drugiej strony, standardem w publikacjach naukowych preprintów (przede wszystkim dotyczy to podstawowego archiwum e-printów **arXiv** pod adresem <http://www.arxiv.org>) jest używanie domyślnych czcionek \LaTeX -a, a nie czcionek wektorowych.⁶ Wynika to z istnienia problemów z drukowaniem czcionek wektorowych na niektórych drukarkach (także na drukarkach postscriptowych). Problemy te występują zwłaszcza przy drukowaniu wzorów matematycznych.

Zatem decyzja co do wyboru typu używanych w dokumencie czcionek może być, w zależności od sytuacji, różna. (Przy czym, korzystając z pakietu `times`, przyjemnie jest od razu tworzyć pliki `.pdf` przy pomocy `pdf \LaTeX -a`, o czym będzie w jednym z najbliższych rozdziałów.)

1.3 Otoczenia, tabele i listy

Nienumerowaną listę (odpowiednik HTML-owego znacznika ``) tworzy się wykorzystując otoczenie `itemize`, o następującej składni:

```
\begin{itemize}
\item Pierwszy punkt
\item Drugi punkt
\item Trzeci punkt
\end{itemize}
```

Co daje w efekcie:

- Pierwszy punkt
- Drugi punkt

⁶Od pewnego czasu spotyka się jednak prace korzystające z pakietu `bookman` (np. prace Lee Smolina i Fotini Markopoulou począwszy od jesieni 2003).

- Trzeci punkt

Listę numerowaną (odpowiednik $\langle\text{OL}\rangle$) otrzymuje się korzystając z otoczenia `enumerate`:

```
\begin{enumerate}
\item Pierwszy punkt
\item Drugi punkt
\item Trzeci punkt
\end{enumerate}
```

co daje efekt:

1. Pierwszy punkt
2. Drugi punkt
3. Trzeci punkt

Aby wyświetlić partię tekstu dokładnie tak jak wygląda ona w źródłowym pliku tekstowym, korzysta się z polecenia `verbatim`, odpowiednika polecenie $\langle\text{PRE}\rangle$:

```
\begin{verbatim}

string trim(string t, string delim)
{
string s=t;
for(int i=1,ii;i<delim.length()+1;i++)
{
t=s;
s=" ";
for(ii=1;ii<t.length()+1;ii++)
if(t[ii]!=delim[i])
s+=t[ii];
}
return s;
}

\end{verbatim}
```

daje właśnie taki efekt jak powyższy.

Tabele tworzy się zaś przy pomocy otoczenia `tabular`. Przykładowo:

```
\begin{tabular}{lr|c}
towar & & waga netto & & cena\\\hline
szyneczka wieprzowa & & 1kg & & 22zł\\
pasztet drobiowy & & 175g & & 1zł\\
patyczki do uszu & & 20kg & & 130zł
\end{tabular}
```

utworzy tabelkę o pierwszej kolumnie wyjustowanej na lewo, drugiej wyjustowanej na prawo, trzeciej wycenowanej, z linią pomiędzy drugą a trzecią kolumną oraz pierwszym a drugim wierszem. Istotne jest, że ostatni wiersz nie kończy się znakiem przejścia do następnej linii. `\hline` to oczywiście polecenie produkujące *horizontal line*. A oto nasza tabelka:

towar	waga netto	cena
szyneczka wieprzowa	1kg	22zł
pasztet drobiowy	175g	1zł
patyczki do uszu	20kg	130zł

1.4 Skład wyrażeń matematycznych

Paragraf ten znajduje się na końcu tego szybkiego omówienia języka, jednakże \TeX (a zatem i \LaTeX) powstał właśnie w celu umożliwienia jego twórcy, Donaldowi Knuthowi, poprawnego składu wyrażeń matematycznych. Wyrażenia matematyczne formułuje się wewnątrz otoczeń matematycznych. Trzy najpopularniejsze z nich to:

- równania:

```
\begin{equation}
...
\end{equation}
```

- równania bez podanego numeru porządkowego równania:

```
\[...\]
```

- oraz tryb matematyczny wierszowy (który, w odróżnieniu od dwóch pierwszych, nie powoduje przejścia do nowego akapitu i może być stosowany w jednej linii z tekstem):

```
$...$
```

A oto przykłady:

```
\begin{equation}
p_x = \frac{\delta L}{\delta v} = mv,
\end{equation}
```

utworzy nam

$$p_x = \frac{\delta L}{\delta v} = mv. \quad (1)$$

Zaś

```
\[\hat{H} = -\frac{\hbar^2}{8m\pi^2} \frac{d^2}{dx^2} + V(x)\]
```

utworzy

$$\hat{H} = -\frac{\hbar^2}{8m\pi^2} \frac{d^2}{dx^2} + V(x)$$

A cóż z trybem matematycznym wierszowym? Można powiedzieć, że przydaje się on często, zwykle co $\alpha * 7 + \beta^2 - \frac{\gamma}{2}$ sekund, co jest równe mniej więcej $\sin \delta + \sum_{i=1}^{\infty} 0.2^i + \int_a^b x dx$ minut, nieprawdaż?

Fragment ten został uzyskany z kodu:

A cóż z trybem matematycznym wierszowym? Można powiedzieć, że przydaje się on często, zwykle co $\alpha * 7 + \beta^2 - \frac{\gamma}{2}$ sekund, co jest równe mniej więcej $\sin \delta + \sum_{i=1}^{\infty} 0.2^i + \int_a^b x dx$ minut, nieprawdaż?

Duże operatory matematyczne generuje się następującymi poleceniami:

\cap	<code>\bigcap</code>	\cup	<code>\bigcup</code>	\odot	<code>\bigodot</code>	\oplus	<code>\bigoplus</code>
\otimes	<code>\bigotimes</code>	\sqcup	<code>\bigsqcup</code>	\uplus	<code>\biguplus</code>	\sum	<code>\sum</code>
\vee	<code>\bigvee</code>	\wedge	<code>\bigwedge</code>	\coprod	<code>\coprod</code>	\prod	<code>\prod</code>
\int	<code>\int</code>	\oint	<code>\oint</code>				

Górny i dolny indeks w takich operatorach zapełnia się korzystając ze poleceń `_` oraz `^`, przy czym granice zostaną inaczej zapisane w trybie matematycznym wierszowym (aby to zmienić, pomiędzy poleceniem operatora a znakami `_`/lub `^` należy wpisać (bez żadnej spacji!) polecenie `\limits`). Na przykład kod

```
\sum_{i=0}^{i=\infty} (i-7)^3
```

zapisany w trybie wierszowym, czyli wewnątrz otoczenia `$...$` wygeneruje $\sum_{i=0}^{i=\infty} (i-7)^3$ (z poleceniem

```
\limits
```

otrzymamy zaś: $\sum_{i=0}^{i=\infty} (i-7)^3$), natomiast zapisany wewnątrz otoczenia matematycznego `equation`

lub wewnątrz otoczenia `\[. . . \]` da rezultat:

$$\sum_{i=0}^{i=\infty} (i - 7)^3.$$

Istnieje bardzo dużo poleceń służących do składu wyrażeń matematycznych (bowiem istnieje wiele znaków i rozmaitych typów wyrażeń). Bardzo dużo informacji na ten temat znajduje się w książce [1], oraz w dokumencie [2]. Poniższe tabelki (stworzone na bazie tabelki w książce [1]) przedstawiają najpopularniejsze używane symbole matematyczne:⁷

mała litera	kod	duża litera	kod	duża pochyła	kod
α	<code>\alpha</code>	A	<code>\mathrm{A}</code>	A	<code>A</code>
β	<code>\beta</code>	B	<code>\mathrm{B}</code>	B	<code>B</code>
γ	<code>\gamma</code>	Γ	<code>\Gamma</code>	Γ	<code>\mathit{\Gamma}</code>
δ	<code>\delta</code>	Δ	<code>\Delta</code>	Δ	<code>\mathit{\Delta}</code>
ϵ	<code>\epsilon</code>	E	<code>\mathrm{E}</code>	E	<code>E</code>
ζ	<code>\zeta</code>	Z	<code>\mathrm{Z}</code>	Z	<code>Z</code>
η	<code>\eta</code>	H	<code>\mathrm{H}</code>	H	<code>H</code>
θ	<code>\theta</code>	Θ	<code>\Theta</code>	Θ	<code>\mathit{\Theta}</code>
ι	<code>\iota</code>	I	<code>\mathrm{I}</code>	I	<code>I</code>
κ	<code>\kappa</code>	K	<code>\mathrm{K}</code>	K	<code>K</code>
λ	<code>\lambda</code>	Λ	<code>\Lambda</code>	Λ	<code>\mathit{\Lambda}</code>
μ	<code>\mu</code>	M	<code>\mathrm{M}</code>	M	<code>M</code>
ν	<code>\nu</code>	N	<code>\mathrm{N}</code>	N	<code>N</code>
ξ	<code>\xi</code>	Ξ	<code>\Xi</code>	Ξ	<code>\mathit{\Xi}</code>
o	<code>o</code>	O	<code>\mathrm{O}</code>	O	<code>O</code>
π	<code>\pi</code>	Π	<code>\Pi</code>	Π	<code>\mathit{\Pi}</code>
ρ	<code>\rho</code>	P	<code>\mathrm{P}</code>	P	<code>P</code>
σ	<code>\sigma</code>	Σ	<code>\Sigma</code>	Σ	<code>\mathit{\Sigma}</code>
τ	<code>\tau</code>	T	<code>\mathrm{T}</code>	T	<code>T</code>
υ	<code>\upsilon</code>	Υ	<code>\Upsilon</code>	Υ	<code>\mathit{\Upsilon}</code>
ϕ	<code>\phi</code>	Φ	<code>\Phi</code>	Φ	<code>\mathit{\Phi}</code>
χ	<code>\chi</code>	X	<code>\mathrm{X}</code>	X	<code>X</code>
ψ	<code>\psi</code>	Ψ	<code>\Psi</code>	Ψ	<code>\mathit{\Psi}</code>
ω	<code>\omega</code>	Ω	<code>\Omega</code>	Ω	<code>\mathit{\Omega}</code>

Inne często stosowane znaki matematyczne (ostatnie dwa wymagają włączenia pakietu `amssymb`):

symbol	kod
∂	<code>\partial</code>
ε	<code>\varepsilon</code>
ϑ	<code>\vartheta</code>
ϖ	<code>\varpi</code>
ϱ	<code>\varrho</code>
φ	<code>\varphi</code>
ς	<code>\varsigma</code>
\varkappa	<code>\varkappa</code>
\digamma	<code>\digamma</code>

Oto tabela poleceń generujących zwyczajne symbole matematyczne:

⁷Funkcjonowanie komendy `\mathit` może zależeć od tego, czy do dokumentu włączony jest pakiet `amsmath` – jeśli tak, to takie polecenie jak `\mathit{\Theta}` może nie dać pochyłej litery i trzeba będzie skorzystać z takiego polecenia jak `\varTheta`. Polecenia `\varCośtam` przynależą do pakietu `amsmath`.

\forall	<code>\forall</code>	\exists	<code>\exists</code>	\neg	<code>\neg</code>	\top	<code>\top</code>
\perp	<code>\bot</code>	\emptyset	<code>\emptyset</code>	∞	<code>\infty</code>	\aleph	<code>\aleph</code>
\hbar	<code>\hbar</code>	/	/			\Vdash	<code>\Vert</code>
\backslash	<code>\backslash</code>	\P	<code>\P</code>	\S	<code>\S</code>	!	!
#	<code>\#</code>	%	<code>\%</code>	-	<code>_</code>	\$	<code>\\$</code>
&	<code>\&</code>	\imath	<code>\imath</code>	\jmath	<code>\jmath</code>	ℓ	<code>\ell</code>
\wp	<code>\wp</code>	\Re	<code>\Re</code>	\Im	<code>\Im</code>	'	<code>\prime</code>
∇	<code>\nabla</code>	\surd	<code>\surd</code>	\flat	<code>\flat</code>	#	<code>\sharp</code>
\natural	<code>\natural</code>	†	<code>\dag</code>	‡	<code>\ddag</code>	?	?
\sphericalangle	<code>\angle</code>	\spadesuit	<code>\spadesuit</code>	\heartsuit	<code>\heartsuit</code>	\diamondsuit	<code>\diamondsuit</code>
\clubsuit	<code>\clubsuit</code>	\triangle	<code>\triangle</code>	∂	<code>\partial</code>	\int	<code>\smallint</code>
@	@	.	.				

Polecenia dostępne po włączeniu pakietu `latexsym`:

\diamond `\Diamond` \square `\Box`

Polecenia dostępne po włączeniu pakietu `amssymb`:

#	<code>\nexists</code>	\emptyset	<code>\varnothing</code>
\beth	<code>\beth</code>	\beth	<code>\gimel</code>
\daleth	<code>\daleth</code>	\hbar	<code>\hslash</code>
/	<code>\diagup</code>	\diagdown	<code>\diagdown</code>
\Bbbk	<code>\Bbbk</code>	\backprime	<code>\backprime</code>
\eth	<code>\eth</code>	\bigstar	<code>\bigstar</code>
\circledS	<code>\circledS</code>	\Finv	<code>\Finv</code>
\sphericalangle	<code>\measuredangle</code>	\sphericalangle	<code>\sphericalangle</code>
\lozenge	<code>\lozenge</code>	\blacklozenge	<code>\blacklozenge</code>
\vartriangle	<code>\vartriangle</code>	\blacktriangle	<code>\blacktriangle</code>
\triangledown	<code>\triangledown</code>	\blacktriangledown	<code>\blacktriangledown</code>
\square	<code>\square</code>	\blacksquare	<code>\blacksquare</code>
\complement	<code>\complement</code>	\mho	<code>\mho</code>
\Game	<code>\Game</code>		

A oto jeszcze jedna, tym razem już ostatnia tabelka z przydatnymi symbolami (po więcej zapraszam do prac [1] i/lub [2]):

\pm	<code>\pm</code>	\mp	<code>\mp</code>	\times	<code>\times</code>
\div	<code>\div</code>	$*$	<code>\divideontimes</code>	\star	<code>\star</code>
\cdot	<code>\cdot</code>	\bullet	<code>\bullet</code>	\circ	<code>\circ</code>
\oplus	<code>\oplus</code>	\otimes	<code>\otimes</code>	\odot	<code>\odot</code>
\wr	<code>\wr</code>	\wedge	<code>\wedge</code>	\vee	<code>\vee</code>
\ll	<code>\ll</code>	\lless	<code>\lless</code>	\leq	<code>\leq</code>
\leq	<code>\leq</code>	\lessssim	<code>\lessssim</code>	\prec	<code>\prec</code>
\in	<code>\in</code>	\subset	<code>\subset</code>	\subseteq	<code>\subseteq</code>
\dashv	<code>\dashv</code>	\vdash	<code>\vdash</code>	\gg	<code>\gg</code>
\ggg	<code>\ggg</code>	\geq	<code>\geq</code>	\geq	<code>\geq</code>
\gtrsim	<code>\gtrsim</code>	\succ	<code>\succ</code>		<code>\mid</code>
\doteq	<code>\doteq</code>	\circeq	<code>\circeq</code>	\equiv	<code>\equiv</code>
\sim	<code>\sim</code>	\backsimeq	<code>\backsimeq</code>	\simeq	<code>\simeq</code>
\cong	<code>\cong</code>	\approx	<code>\approx</code>	\propto	<code>\propto</code>
\nearrow	<code>\nearrow</code>	\searrow	<code>\searrow</code>	\swarrow	<code>\swarrow</code>
\nwarrow	<code>\nwarrow</code>	\leftarrow	<code>\leftarrow</code>	\longleftarrow	<code>\longleftarrow</code>
\Leftarrow	<code>\Leftarrow</code>	\Lleftarrow	<code>\Lleftarrow</code>	\rightarrow	<code>\rightarrow</code>
\longrightarrow	<code>\longrightarrow</code>	\Rightarrow	<code>\Rightarrow</code>	\Longrightarrow	<code>\Longrightarrow</code>
\leftrightarrow	<code>\leftrightarrow</code>	\Leftrightarrow	<code>\Leftrightarrow</code>	\mapsto	<code>\mapsto</code>

Negację (przekreślenie) tych symboli można otrzymać np. zamiast `\in` pisząc `\notin`. Ułamki otrzymuje się w trybie matematycznym korzystając z polecenia `\frac{licznik}{mianownik}`, pierwiastki korzystając z polecenia `\sqrt{wyrażenie_pod_pierwiastkiem}`, zaś takie nawiasy, które same się dopasowują do rozmiarów wyrażenia matematycznego uzyskuje się korzystając z poleceń `\right(i\left)` lub `\right[i\left]` lub też `\right\}` i `\left\{`. Napisy typu `\arccos`, `\sin`, `\exp`, `\log`, `\ker`, `\lim` (i inne) uzyskuje się poprzez polecenia `\arccos`, `\sin`, `\exp`, i tak dalej. Uwaga! Wewnątrz trybu matematycznego używa się często polecenia

nia `\mbox{}` w celu otrzymania zwykłego tekstu. Aby wewnątrz trybu matematycznego otrzymać poprawnie napisane polskie litery, trzeba stosować jeden z opisanych wcześniej (patrz par. 1.2.5) sposobów używających wyłącznie znaków siedmiobitowego ASCII.

W \LaTeX -u można też pisać różnymi czcionkami *MATEMATYCZNYMI* ujmując poszczególne litery (oczywiście wewnątrz trybu matematycznego) w odpowiednie polecenia (style `\mathfrac` oraz `\mathbb` wymagają załadowania pakietu `amsmath` lub `amssymb`):

polecenie	przykład
<code>\mathfrac{}</code>	$\mathfrac{OLA}{MA} kota$
<code>\mathcal{}</code>	$\mathcal{OLA} MA \ \int\cup\cap$
<code>\mathbb{}</code>	$\mathbb{OLA} MA \nabla \times \approx \emptyset$
<code>\mathrm{}</code>	$\mathrm{OLA} MA kota$
<code>\mathit{}</code>	$\mathit{OLA} MA kota$

1.5 Rysunki

1.5.1 Włączanie do dokumentu rysunków i grafik z plików zewnętrznych

\LaTeX umożliwia włączanie do dokumentów rysunków zapisanych w formacie Encapsulated PostScript (`.eps`)⁸. W tym celu trzeba do dokumentu włączyć pakiet `graphicx`:

```
\usepackage{graphicx}
```

oraz, w odpowiednim ku temu miejscu, wydać polecenie umieszczenia grafiki, np.

```
\includegraphics[width=5cm]{kotek.eps}
```

Opcjonalnym argumentem jest docelowy rozmiar obrazka wewnątrz dokumentu, do którego to rozmiaru rysunek zostanie przeskalowany. Można podać inny argument opcjonalny, opisujący bezwzględny rozmiar pliku w pikselach:

```
\includegraphics[bb = 0 0 243 170]{dok2Fig1.eps}
```

ewentualnie skalując go nieco:

```
\includegraphics[bb = 0 0 243 170, scale=0.75]{dok2Fig1.eps}
```

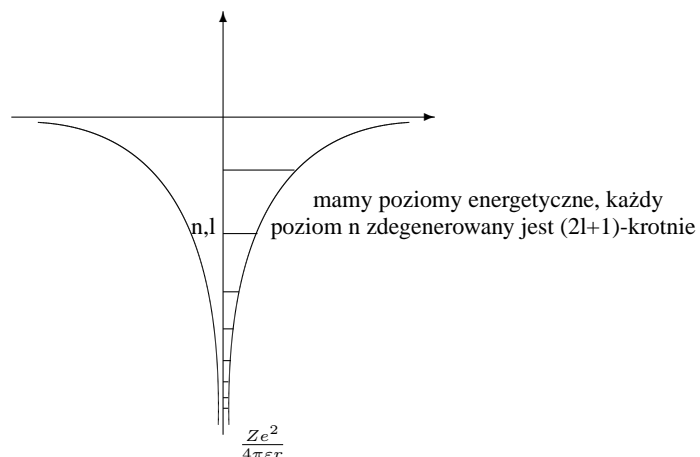
lub obracając (służy do tego pakiet `wrapfig`). Z włączaniem rysunków wiążą się w sposób istotny kwestie związane z tworzeniem dokumentów w formacie Adobe Acrobat (`.pdf`), które to kwestie zostały omówione w paragrafie 3.2.2.

Warto również wiedzieć o pewnym problemie z umieszczaniem rysunków w tekście: otóż \LaTeX często nie umieszcza rysunku tam, gdzie użytkownik by chciał. Nie mam pojęcia co z tym zrobić.

1.5.2 Rysowanie wewnątrz \LaTeX -a

\LaTeX dysponuje swoim własnym językiem przeznaczonym do wykonywania prostych rysunków. Oto przykład:

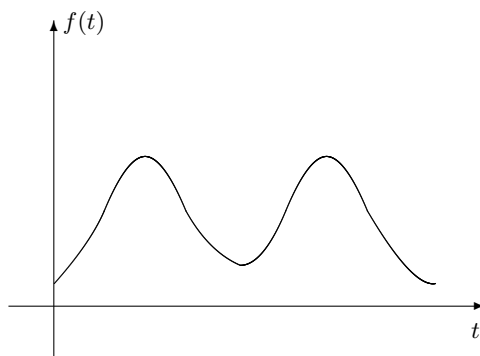
⁸Do konwersji zwyczajnego rysunku (np. `.gif`, `.bmp` czy `.jpg`) do formatu `.eps` można pod Windows wykorzystać program Paint Shop Pro. Na Linuxie istnieje wiele możliwości konwersji, ale najbardziej uniwersalnym jest program GIMP.



Rysunek ten został utworzony przy pomocy kodu:

```
\begin{center}
\setlength{\unitlength}{0.7mm}
\begin{picture}(100,100)
\put(0,75){\vector(1,0){80}}
\put(40,15){\vector(0,1){80}}
\put(43,12){\mbox{\$\frac{Ze^2}{4\pi\varepsilon r}\$}}
\q bezier(41,17)(40,72)(75,74)
\q bezier(39,17)(40,72)(5,74)
\put(40,20){\line(1,0){1}}
\put(40,22){\line(1,0){1}}
\put(40,25){\line(1,0){1}}
\put(40,29){\line(1,0){1.5}}
\put(40,35){\line(1,0){2}}
\put(40,42){\line(1,0){3}}
\put(40,53){\line(1,0){6.5}}
\put(34,53){\mbox{n,l}}
\put(57,58){\mbox{mamy poziomy energetyczne, ka\zdy}}
\put(49,53){\mbox{poziom n zdegenerowany jest (2l+1)-krotnie}}
\put(40,65){\line(1,0){13.5}}
\end{picture}
\end{center}
```

Zaś ten rysunek:



otrzymany został przy pomocy kodu:

```

\begin{center}
\setlength{\unitlength}{0.6mm}
\begin{picture}(110,80)
\put(0,12){\vector(1,0){105}}
\put(10,0){\vector(0,1){75}}
\put(102,5){\mbox{\$t\$}}
\put(12,73){\mbox{\$f(t)\$}}
\qbezier(10,17)(18,26)(21,33)
\qbezier(21,33)(26,45)(30,45)
\qbezier(30,45)(34,45)(39,33)
\qbezier(39,33)(44,24)(51,21)
\qbezier(51,21)(56,21)(61,33)
\qbezier(61,33)(66,45)(70,45)
\qbezier(70,45)(74,45)(79,33)
\qbezier(79,33)(89,16)(94,17)
\end{picture}
\end{center}

```

Liczby bezwymiarowe oznaczają liczbę jednostek ustalonych w zmiennej `unitlength`. Polecenie `\begin{picture}(x,y)` ustala rozmiar przestrzeni dostępnej dla rysunku. Współrzędne punktu zaczepienia wektora, linii lub pola tekstowego `mbox` ustala się w argumencie polecenia `\put`. Linie prostą i wektor tworzy się odpowiednio poleceniami `\line(a,b){r}` oraz `\vector(a,b){r}`, gdzie a oraz b oznaczają względne współczynniki nachylenia linii lub wektora, zaś r określa jego długość. Wartości współczynników a oraz b mogą być jedynie z przedziału od -4 do $+4$, co istotnie ogranicza liczbę możliwych kierunków w jakich może być skierowany wektor lub linia. Polecenie `qbezier` generuje oczywiście krzywą Beziera, zaś polecenie `circle` – okrąg. Bardzo polecam własne eksperymenty z rysowaniem.

1.5.3 Korzystanie z rysunków tworzonych przy pomocy GnuPlota

Rysunki i wykresy można tworzyć także w programie GnuPlot (posiada on takie możliwości, jak np. dopasowywanie zależności funkcyjnych do danych, wykreślanie słupków błędów, legendy, trójwymiarowych wykresów itd), a następnie eksportować te rysunki do \LaTeX -a. W tym celu należy uruchomić GnuPlota poleceniem `gnuplot -mono` i wpisać polecenia:

```

set terminal pstex
set output "plot.tex"

```

Tworzone rysunki będą zapisywane do pliku `plot.tex`. Zawartość tego pliku można włączyć w odpowiednim miejscu naszego dokumentu przy pomocy polecenia `\input{plot.tex}`.

1.6 Odwołania

Często zdarza się potrzeba odwołania w jakimś miejscu dokumentu do określonego rysunku, równania, czy paragrafu. Do tego celu służy polecenie `\ref{identyfikator}`. Identyfikator określa właśnie ten obiekt w tekście, do którego chcemy się odwołać. Nadanie identyfikatora obiektowi (np. równaniu) następuje przez umieszczenie wewnątrz niego polecenia `\label{identyfikator}`. Identyfikator powinien być dowolną nazwą składającą się z liter i ewentualnie liczb, przy czym pierwszy znak identyfikatora powinien być literą. Ponieważ rysunki, równania czy choćby paragrafy zazwyczaj bywają numerowane, rezultatem użycia odwołania jest odpowiedni numer posiadany przez obiekt do którego się odwoływaliśmy.

2 Praca z \LaTeX -em

2.1 Jak edytować pliki \LaTeX -a?

Korzystając z dowolnego edytora tekstów, bowiem pliki `.tex` to pliki siedmiobitowego kodu ASCII. Istnieje wiele środowisk wspomagających pracę z kodem źródłowym \TeX -a i \LaTeX -a (pod Windows takie jak np.

\TeX Shell, $4\TeX$, czy też WinShell), lecz autor niniejszego tekstu przyjmuje stanowisko (zdając sobie sprawę z nieortodoksyjności tegoż stanowiska), że najsympatyczniejszym edytorem do \TeX -a, tak jak i do HTML-a, jest `gedit` pod X-Windows oraz `picco` pod konsolę Unixa, które właśnie poprzez swe niewyspecyfikowanie ułatwiają pracę. Natomiast w kwestii Windowsów wypowie się gościnnie Krzysiek Miernik vel Pan Mierniczy:

Ponieważ jestem użytkownikiem Windowsów, to mogę dorzucić małą uwagę do tekstu. Otóż większość problemów z uruchamianiem pod Windowsami rozwiązuje doskonały darmowy edytor TeXnicCenter. Jest to bardzo miłe środowisko pracy, które koloruje polecenia i insze duperele na odpowiednie barwy co by były wśród tekstu widoczne tudzież jest na tyle otwarte i rozbudowane, że można robić w nim co się chce. Ma też różne fajne guziczki i można sobie wybierać w co i jak się kompiluje bez głębokiego myślenia. Wystarczy ściągnąć MiXTeXA, a potem TeXnicCenter i życie jest proste. Głęboko polecam.

2.2 Jak obejrzeć pracę napisaną w \LaTeX -u?

Trzeba ją wpierek skompilować (przetworzyć) do formatu, który może być odczytany przez przeglądarkę dokumentów napisanych w \TeX -u. I tak jak pliki `.cpp` kompilują się czasem (czasem, bowiem w zależności od systemu) do plików `.exe`, tak samo pliki `.tex` kompilują się do plików `.dvi` (pliki \TeX -a i \LaTeX -a zwyczajowo mają to samo rozszerzenie). DVI jest skrótem od *Device Independent Format*, co znaczy, że przynajmniej w teorii pliki te powinny być bez problemu odczytywane na każdym systemie. W praktyce są podobnie spotykane problemy pomiędzy systemami Linux i DOS, chociaż nie spotkałem się (mimo intensywnej praktyki) z konfliktami na linii pomiędzy systemami Linux i Windows. Zwykle ewentualne konflikty i problemy w korzystaniu z \LaTeX -a wynikają ze specyfiki używanej jego dystrybucji. Dla każdego systemu i dla każdej dystrybucji istnieje właściwa przeglądarka plików `.dvi`. Przeglądarką właściwą dla Windows jest `windvi` [11].

2.3 Jak pracować ze środowiskiem \TeX -a/ \LaTeX -a pod Windows?

Ponieważ środowisko \TeX -a wywodzi się ze środowiska Unixa, jest ono przystosowane do obsługi przede wszystkim z poziomu linii komend. Aby mieć dostęp do wszystkich istotnych poleceń w linii komend, trzeba do zmiennej środowiskowej `PATH` dopisać koniecznie katalog binariów \TeX -a: `... \TEX\bin\win32`, jak również (aby móc korzystać z `GhostView` oraz `ps2pdf`) katalog binariów i bibliotek `GhostView`: `... \gs\bin, ... \gs\lib`. Pozostałe użyteczne programy (takie jak np. `rtf2latex2e`) należy podpiąć pod linię komend umieszczając pliki `.bat` do nich w którymś z katalogów zawartych w zmiennej `PATH`. Autor niniejszego opracowania poleca `... \WINDOWS\COMMAND` (a w nowszych Windowsach: `... \WINDOWS\SYSTEM32`) jako uniwersalny katalog zbiorczy dla plików wsadowych `.bat`. Zmianę zmiennej środowiskowej `PATH` należy przeprowadzić w pliku `AUTOEXEC.BAT`, zaś przy tworzeniu plików wsadowych trzeba pamiętać o przekazaniu argumentów wywołania batcha do samego programu. Przykładowy batch dla `rtf2latex` może więc mieć nazwę `C:\WINDOWS\COMMAND\R2L.BAT` i mieć zawartość: `@D:\text\r2l\rtf2latex2e.exe %1 %2 %3 %4 %5`.

2.3.1 Praktyczne rozwiązanie polecane przez autora

1. Stworzyć plik, (np. `pretex.bat`) zawierający ścieżki dostępu do katalogów z binariami i bibliotekami \TeX -a i `GhostScriptu`, o zawartości podobnej do tej:


```
@SET PATH=%PATH%;D:\PROGRAMY\TEX\BIN\WIN32;"D:\PROGRAM FILES\ALLADIN\GS6.01\BIN";"D:\PROGRAM FILES\ALLADIN\GS6.01\LIB"
```
2. Umieścić go w katalogu `C:\WINDOWS\COMMAND` (dla Windows 95, 98) lub `C:\WINDOWS\SYSTEM32` (dla Windows 2000, XP). Czynność tę należy wykonać tylko raz, pod warunkiem, że później ani \TeX , ani `GhostScript/GhostView` nie będą przenoszone do innych katalogów. Jakby co, to trzeba będzie odpowiednio zmienić skrypt.
3. Aby uruchomić interpretator komend, trzeba wybrać *Start/Uruchom...* (lub wcisnąć przyciski 'Windows' + 'R') i wpisać odpowiednie polecenie: `command` (dla Windows 95, 98) lub `cmd` (dla Windows 2000, XP). Można też wpisać je w `Windows/Total Commanderze`.
4. Po każdym uruchomieniu okna interpretatora komend wpisać `pretex`. Od tego momentu w tym konkretnym oknie można sobie spokojnie \LaTeX -ować, $\pdf\LaTeX$ -ować, `dvipsować`, `ps2pdfować`, i tak dalej.
5. Interpretator komend zamyka się poleceniem `exit`.

2.4 Jak wykorzystać cały obszar kartki A4 lub A5?

Domyślnie \LaTeX pozostawia po wszystkich stronach kartki dość duże marginesy. Można to jednak zmienić dodając w nagłówku dokumentu (czyli przed poleceniem `\begin{document}`) następujące polecenia:

```
\setlength{\textheight}{24cm}
\setlength{\textwidth}{15.92cm}
\setlength{\footskip}{10mm}
\setlength{\oddsidemargin}{0mm}
\setlength{\evensidemargin}{0mm}
\setlength{\topmargin}{0mm}
\setlength{\headsep}{5mm}
```

Powyższe komendy ustalają (dla całego dokumentu): wysokość obszaru właściwego dokumentu (nie licząc stopki i nagłówka) na 24cm, szerokość obszaru dokumentu na 15.92cm, odległość stopki od spodu tekstu na 10mm (stopki, lecz nie przypisów uzyskiwanych poprzez `\footnote` – te są zawsze umieszczane wewnątrz obszaru właściwego dokumentu), odległość nagłówka od tekstu na 5mm, zaś marginesy boczne oraz górny na 0mm. Ustawienie ich na zero nie likwiduje marginesów całkowicie – w \LaTeX -u istnieje nieprzekraczalna granica sztywnego marginesu równego 1 cal.

Dla kartki A5 całkiem niezłe ustawienia podstawowe to:

```
\setlength{\textheight}{16.92cm}
\setlength{\textwidth}{9.6cm}
```

Dla dokumentu o formacie A5 warto jeszcze dodatkowo umieścić na początku dokumentu, bezpośrednio po poleceniu `\begin{document}`, polecenie `\special{papersize=14.85cm,21.0cm}`.

2.5 Co robić z błędami w kodzie źródłowym?

Najlepiej – poprawić je :-). Gdy jednak chcesz skompilować plik źródłowy mimo błędów i ostrzeżeń, a nie masz ochoty na ciągłe wciskanie `<ENTER>`, możesz uruchomić kompilację \LaTeX -a w następujący sposób:

```
latex -c-style-errors plik_źródłowy.tex
```

W ten sposób \LaTeX będzie sam starał się naprawić zaistniałe błędy. Ujemną stroną tej metody jest występowanie czasami zjawiska ‘martwej pętli’ - poprawiania błędów w nieskończoność. Wówczas pozostaje jedynie przerwanie pracy \LaTeX -a poprzez wciśnięcie kombinacji klawiszy `<CTRL>+C` lub `<CTRL>+Y`.

3 Tworzenie dokumentów elektronicznych w różnych formatach plików

Format `.dvi` nie jest poręczny – nie zawiera wewnątrz siebie plików graficznych (przez co czasem ładowanie ich z ogromnych plików `.eps` zajmuje krocie), występują problemy z drukowaniem plików `.dvi` na drukarkach igłowych, nie są również wystarczająco rozpowszechnione jego przeglądarki. Zatem przydatna jest umiejętność zamiany pliku w tym formacie do najbardziej popularnych formatów elektronicznej publikacji – PostScript (`.ps`) oraz Adobe Acrobat (`.pdf`). W sekcji tej omówię również możliwość bezpośredniej kompilacji \LaTeX -a do formatu `.pdf`.

3.1 Tworzenie plików `.ps` (PostScript)

Pliki `.ps` najłatwiej jest stworzyć poprzez konwersję DVI→PS. Do takiej konwersji służy oczywiście program `dvips`. Polecenie

```
dvips plik
```

Tworzy plik `plik.ps` na podstawie pliku `plik.dvi`. Plik `.ps` można już odczytać odpowiednimi przeglądarkami dokumentów tego formatu, z których najpopularniejszą jest bez wątpienia GhostView.

3.1.1 Tworzenie publikacji formatu A5 (i innych nietypowych)

`dvips` domyślnie ustawia rozmiar dokumentu na A4, ignorując informacje zawarte w nagłówku \LaTeX -owego dokumentu (np. `\documentclass[a5paper, titlepage]{article}`). Aby otrzymać w wyniku konwersji `dvips` dokument o rozmiarze strony A5 trzeba użyć następującego polecenia:

```
dvips -T 14.85cm,21.0cm plik,
```

lub po prostu utworzyć plik wsadowy `dvipsa5.bat` o zawartości `dvips -T 14.85cm,21.0cm %1` w katalogu znajdującym się na systemowej ścieżce dostępu (np. w `... \WINDOWS \SHELL32`). Dla innych nietypowych rozmiarów papieru wystarczy po prostu podać odpowiednie wartości długości i szerokości strony.

3.2 Tworzenie plików `.pdf` (Adobe Acrobat)

Pliki `.pdf` można otrzymywać w dwojaki sposób: poprzez konwersję pliku `.ps` na plik `.pdf`, oraz przez bezpośrednią kompilację \LaTeX -owego źródła do dokumentu `.pdf`. Drugi z opisanych poniżej sposobów jest zdecydowanie bardziej uniwersalny.

3.2.1 `ps2pdf`

Polecenie

```
ps2pdf plik.ps plik.pdf
```

konwertuje `plik.ps` na `plik.pdf`. I już. Sposób ten ma jednak istotne wady:

1. Wymaga posiadania zainstalowanego w systemie środowiska GhostScript/GhostView.
2. Konwersja `ps2pdf` bywa bardzo wolna (dla dużych dokumentów).
3. Pliki `.pdf` otrzymane konkurencyjną metodą `pdflatex` potrafią być nawet dwa i pół raza mniejsze.

3.2.2 Bezpośrednia kompilacja pliku \LaTeX do dokumentu w formacie `.pdf`

Do tego celu służy program `pdflatex`. Jednakże, aby źródła \LaTeX -owe kompilowały się w ten sposób bez problemu, trzeba wprowadzić do nich kilka uzupełnień, związanych przede wszystkim z grafiką:

1. Wiersz

```
\usepackage{graphicx}
zastąpić trzeba sekwencją:
\newif\ifpdf
\ifx\pdfoutput\undefined
  \pdffalse
\else
  \pdfoutput=1
  \pdftrue
\fi
\ifpdf
  \usepackage[pdftex]{graphicx}
  \pdfcompresslevel=9
\else
  \usepackage{graphicx}
\fi
```

2. Po `\begin{document}` należy dodać linijkę

```
\DeclareGraphicsExtensions{.pdf,.png,.eps}
```

3. Zaś każde wystąpienie polecenia włączenia grafiki do dokumentu, np.

```
\includegraphics[width=5cm]{kotek.eps}
```

zastąpić trzeba sekwencją włączania do dokumentów odpowiednich plików, w zależności od używanego kompilatora:

```
\ifpdf
  \includegraphics[width=5cm]{kotek}
\else
  \includegraphics[width=5cm]{kotek.eps}
\fi
```

Tak zmieniony plik źródłowy można już skompilować (przetworzyć) zarówno pdf \LaTeX -em, wydając polecenie `pdflatex plik`, lub `pdflatex -c-style-errors plik`, jak i zwykłym \LaTeX -em, wydając polecenie `latex plik`, lub `latex -c-style-errors plik`.

Z powyższym przepisem wiąże się istotna uwaga: \LaTeX akceptuje przy kompilacji rysunki w formacie .eps (Encapsulated PostScript), natomiast pdf \LaTeX plików tych nie trawi, akceptując za to pliki w formatach .pdf, .png, oraz .jpg. Dlatego też, chcąc produkować na podstawie tego samego dokumentu pliki .ps (poprzez `latex` i `dvips`) oraz .pdf (poprzez `pdflatex`), trzeba w katalogu ze źródłem przechowywać grafikę w dwóch wersjach: w formacie .eps oraz w jednym spośród formatów: .pdf, .png, .jpg. Notabene, format .png najlepiej spośród wszystkich istniejących formatów *bezstratnie* kompresuje grafikę. Jego kompresja jest lepsza od kompresji .gif, przy czym, w odróżnieniu od .gif, .png radzi sobie bezproblemowo z 24-bitową paletą kolorów. Rysunki w formacie .eps konwertować można do formatu .pdf przy pomocy programu `epstopdf`, należącego do dystrybucji \TeX -a, jednak wymagającego obecności programu `GhostView` w systemie. Pod Windows obustronną konwersję między .eps, .jpg i .png, jak i innymi formatami, można przeprowadzić z pomocą np. ulubionego przez autora tego tekstu programu `PaintShop Pro`.

4 Eksport/Import pomiędzy \LaTeX -em, Wordem, PowerPointem i innymi programami

4.1 Importowanie plików z Worda, PowerPointa i rozmaitych innych programów do \LaTeX -a

Skonwertowanie dokumentów napisanych w formacie programów typu Word, czy PowerPoint do formatu \LaTeX nie jest trudne. Wystarczy wyeksportować dany dokument z poziomu jednego z tych programów do pliku w formacie *Rich Text Format* (pliki z rozszerzeniem .rtf), a następnie posłużyć się programem `Rtf2 \LaTeX 2 ϵ` pisząc po prostu:

```
rtf2latex2e plik.rtf
```

W wyniku pracy tego programu zostanie wygenerowany plik `plik.tex` w formacie \LaTeX -a. Tak utworzony plik wymaga zwykle pewnych ręcznych poprawek w kodzie źródłowym, lecz jest to niezwykle szybka i użyteczna metoda, zwłaszcza, że `rtf2latex2e`:

1. Konwertuje polskie litery do postaci, która jest uniwersalna i może być użyta jednocześnie z cyrylicą.
2. Konwertuje zawarte w dokumencie pliki graficzne jednocześnie do formatu .png oraz .eps, dzięki czemu możemy od razu kompilować źródło \LaTeX -owe i do formatu .ps i do formatu .pdf.

`Rtf2 \LaTeX 2 ϵ` ma problem z poprawną konwersją cyrylicy, jednakże można ten problem rozwiązać: wystarczy przetworzoną przez niego cyrylicę zastąpić tekstem skopiowanym bezpośrednio z dokumentu źródłowego.

`Rtf2 \LaTeX 2 ϵ` jest ogromnie użyteczny w sytuacjach gdy piszemy dużo tekstu o małym wewnętrznym zróżnicowaniu logicznym, lecz o dużej ilości pochyleń i pogrubień czcionki. Wówczas warto skorzystać z programów typu `WordPad`, `KWord`, `AbiWord`, czy po prostu `Word` (bo szybciej jednak zaznacza się określony fragment i klika na pogrubienie (lub wciska `<Ctrl>+B`), niż pisze `\textbf{ }`) i dopiero końcowy efekt takiej pracy skonwertować do \LaTeX -a.

4.2 Eksportowanie wzorów napisanych w \LaTeX -u do Worda, PowerPointa i innych programów

Często spotykana jest sytuacja, kiedy ktoś chce użyć \LaTeX -a do wyprodukowania wzorów matematycznych, które chce umieścić później na jakiejś prezentacji. W tym celu autor niniejszej pracy postępuje (i uczy innych) następująco:

1. Wyprodukować pdf \LaTeX -em dokument PDF zawierający wzory.
2. Otworzyć go pod Acrobat Readerem (lub GhostView lub xpdf).
3. Uruchomić program graficzny do robienia zrzutów ekranu (np. Paint Shop Pro lub GIMP).
4. Uruchomić w programie graficznym opcję robienia zrzutów (Shift+C w PSP).
5. Zrzucić sobie cały ekran ze wzorem, lub tylko wyciąć fragment ze wzorem/wzorami, albo tyle fragmentów ile trzeba.
6. Z programu graficznego, korzystając z metody <CTRL>+C/<CTRL>+V, przenieść ładne wzorki do programu docelowego (np. PowerPointa).

4.3 Podobieństwo języków \LaTeX i HTML

W (bardzo) prostych zastosowaniach składnia \LaTeX -a i HTML-a jest bardzo podobna. Oto tabela porównawcza niektórych poleceń:

HTML	\LaTeX
<BODY>...</BODY>	\begin{document}... \end{document}
<TITLE>...</TITLE>	\title{...}
<PRE>...</PRE>	\begin{verbatim}... \end{verbatim}
...	\begin{itemize}... \end{itemize}
...	\begin{enumerate}... \end{enumerate}
...	\item ...
<I>...</I>	\textit{...}
...	\textbf{...}
<U>...</U>	\underline{...}
	\includegraphics[width=5cm]{kotek.eps}
 	//
 	~

4.3.1 Konwersja z \LaTeX do HTML

Istnieje szereg programów do takiej konwersji. Osobiście polecam program o nazwie TTH ("A \TeX to HTML Translator"). Jego kod źródłowy (napisany w C i kompilowalny zarówno pod DOS/Windows jak i Unix/Linux), jest do ściągnięcia z sieci.

5 Specyficzne problemy praktyczne

5.1 Problemy z Bib \TeX -em

5.1.1 Bib \TeX a sprawa polska

Bib \TeX nie do końca dobrze przetwarza polską bibliografię (właściwie powinno się mówić: spis literatury) – pozostają w niej angielskie słowa, np. *edition* zamiast *wydanie*. W takiej sytuacji trzeba ręcznie zmodyfikować plik wyjściowy bibliografii (o rozszerzeniu .bb1), już po przetworzeniu bibliografii Bib \TeX -em, ale przed ostateczną kompilacją przy pomocy \LaTeX -a lub pdf \LaTeX -a.

5.1.2 Konflikt BibT_EX-a z poleceniem `\textcompwordmark{}`

Korzystając z BibT_EX-a można natknąć się na ciekawy, choć denerwujący błąd: jeśli w dokumencie użyto polecenia `\textcompwordmark{}`, służącego do zabrania tworzenia ligatur (tj. połączenia niektórych liter w jeden ciąg, np. ‘fl’), a następnie dołącza się do niego bibliografię przy pomocy BibT_EX-a, to BibT_EX zgłosi błąd i uniemożliwi dołączenie bibliografii do dokumentu. Rozwiązaniem tego problemu jest, niestety, usunięcie polecenia `\textcompwordmark{}` z kodu źródłowego. (Choć być może istnieje jakieś inne rozwiązanie tego problemu.)

5.2 Problemy z rozmieszczeniem tekstu

5.2.1 Niektóre fragmenty tekstu wystają poza margines, lub, co gorsza, poza kartkę! Co robić?

Spokojnie. Trzeba wiedzieć, że L^AT_EX przenosi wyrazy zgodnie ze znanymi przez niego zasadami pisowni (za poprawne łamanie tekstu po polsku odpowiada pakiet `poliski`). Jednak nie musi on rozpoznać każdego istniejącego w języku wyrazu. Wówczas trzeba mu podpowiedzieć, jakie są możliwości podziału wyrazu przy przejściu do następnej linii. Dokonuje się to przy pomocy polecenia `\-`. Przykładowo wyraz *Wszechświat* daje się poprawnie złamać dopiero po zapisaniu go (w kodzie źródłowym) w postaci *Wszech\świat*.

Aby poprawnie złamać takie wyrazy jak *biało-czerwony*, trzeba włączyć do dokumentu pakiet `poliski` i skorzystać z polecenia `\dywiz{}`:

```
biało\dywiz{ }czerwony,
```

co daje w momencie złamania efekt:

```
biało-  
-czerwony.
```

5.2.2 Rysunki są niepokorne i wciąż mi je L^AT_EX przerzuca względem tekstu! Co robić?

Otóż nie wiem, drogi czytelniku. Sam się z tym męczę od dwóch lat, a moi znajomi też nie wiedzą. Może ktoś nam pomoże?

5.2.3 Czy jest jakaś możliwość szybkiego projektowania wyglądu dokumentów L^AT_EX-owych, na przykład tak jak w Wordzie?

Tak. Istnieje edytor do L^AT_EX-a, zwący się LyX i właśnie taki on jest.

5.2.4 A jak został sformatowany ten dokument?

Następującym poleceniami:

```
\documentclass[a4paper,notitlepage,twoside]{article}  
\newif\ifpdf  
\ifx\pdfoutput\undefined  
\pdffalse  
\else  
\pdfoutput=1  
\pdftrue  
\fi  
\ifpdf  
\usepackage[pdftex]{graphicx}  
\pdfcompresslevel=9  
\else  
\usepackage{graphicx}  
\fi
```

```

\usepackage[cp1250]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[english,polish]{babel}
\usepackage{polski}
\usepackage{amsmath}
\usepackage{amssymb}
\usepackage{fancyhdr}
\usepackage{times}
\newcommand{\szerokosc}{15.92cm}
\newcommand{\tab}{\ }
\newcommand{\bs}{\$\\backslash$}
\setlength{\textheight}{23.5cm}
\setlength{\textwidth}{\szerokosc}
\setlength{\footnotesep}{5mm}
\setlength{\footskip}{10mm}
\setlength{\oddsidemargin}{0mm}
\setlength{\evensidemargin}{0mm}
\setlength{\topmargin}{0mm}
\setlength{\headsep}{5mm}
\setlength{\parindent}{0mm}
\setlength{\parskip}{2.5mm}
%-----
\begin{document}
\pagestyle{fancy}
\fancyhead{}
\fancyhead[LE,RO]{\textsf{W miarę krótki i praktyczny
kurs \LaTeX{}-a w $\pi^e$ minut}}
\fancyhead[RE,LO]{\textbf{\textsf{\thepage}}}}
\fancyfoot[CE,CO]{\ }
\renewcommand{\headrulewidth}{0.4pt}
\renewcommand{\footrulewidth}{0.4pt}
\bibliographystyle{plain}
\nocite{*}
%-----
\title{
...i tak dalej...

```

5.2.5 A jakiś prosty przykład?

```

\documentclass{article}
\begin{document}
\[\frac{ih}{2\pi}\partial_t\psi=\nabla^2\psi+V\psi\]
\end{document}

```

Podziękowania

Dziękuję Konradowi Grochowskiemu i Magdzie Chudzik za wnikliwy β -test wcześniejszej wersji tego tekstu i szereg konstruktywnych uwag, aivogool. Dziękuję także Ewie Słomińskiej, Krzyszkowi Miernikowi, Włodkowi Natorfowi i Konradowi Piszczatowskiemu za uwagi i komentarze.

<>

Literatura

- [1] Antoni Diller, *\LaTeX wiersz po wierszu*, wydawnictwo Helion, Gliwice 2001
Jest to dobra i dość obszerna książka z której do dziś korzystam gdy mam jakiś problem (z \LaTeX -em oczywiście :-).
- [2] Tobias Oetiker, *The Not So Short Introduction To $\LaTeX 2\epsilon$* ,
../texmf/doc/guides/lshort-english/lshort
Jest to chyba najbardziej popularny (znany również w polskim tłumaczeniu) skrypt do nauki \LaTeX -a. Jest w nim więcej informacji niż w niniejszym tekście, lecz w zamian za to – nie jest *so short* i nie jest w π^e minut ;-).
- [3] Michael Doob, *A Gentle Introduction To $T_{\text{E}}X$* ,
../texmf/doc/guides/gentle/gentle
To delikatne wprowadzenie jest do $T_{\text{E}}X$ -a, a nie do \LaTeX -a, więc jeśli ktoś ma dużo wolnych wieczorów i chce mieć dobre podstawy do bycia guru $T_{\text{E}}X$ -em, to polecam...
- [4] Guido Gonzato, *\LaTeX for Word Processor Users*,
Bardzo specyficzny manual do \LaTeX -a, pokazujący jak w nim otrzymać efekty takie, jakie generują poszczególne opcje menu w Wordzie. Dostępny w internecie.
- [5] Wojciech Myszkowski, *Włączanie grafik do tekstów w $\LaTeX 2\epsilon$* ,
Tytuł mówi chyba wszystko. Polecam. Dostępny w internecie.
- [6] Hàn Thế Thành, Sebastian Rahtz, Hans Hagen *The pdf $T_{\text{E}}X$ manual*,
../texmf/doc/pdftex/pdftex-a
Podręcznik do pdf \LaTeX -a.
- [7] C. V. Radhakrishnan *The pdf $T_{\text{E}}X$ FAQ*,
../texmf/doc/pdftex/pdftex-faq
Pytania i odpowiedzi związane z pdf \LaTeX -em.
- [8] Oren Patashnik, *Bib $T_{\text{E}}X$ ing*,
../texmf/doc/bibtex/btxdoc
- [9] Ujwal S. Sathyam, *rtf2 $\LaTeX 2\epsilon$ Documentation*,
../rtf2latex2e/doc/rtf2latex2edoc
- [10] Tomas Rokicki, *Dvips: A DVI-to-PostScript Translator*,
../texmf/doc/dvips/dvips
- [11] Fabrice Popineau, *Windvi 0.67 User's Manual*,
../texmf/doc/windvi/windvi
- [12] Piet van Oostrum, *Page layout in \LaTeX* ,
../texmf/doc/latex/fancyhdr/fancyhdr
- [13] David P. Carlisle, *Packages in the 'graphics' bundle*,
../texmf/doc/latex/graphics/grfguide
- [14] David P. Carlisle, *The colortbl package*,
../texmf/doc/latex/carlisle/colortbl

Jako znacznie bogatsze od tego kursu źródło wiedzy o języku \LaTeX polecam książkę [1]. \LaTeX -a i $T_{\text{E}}X$ -a można się również nauczyć (lub poszerzyć swoje wiadomości) czytając dokumenty [2], [3], oraz manuale do konkretnych składników środowiska oraz pakietów, np. [13], [9], [10], [11], [6], [7], [12], [8], [14], ...