

# Rozpoczynamy pracę z L<sup>A</sup>T<sub>E</sub>X-em

*materiały pomocnicze do zajęć*

*Wprowadzenie do L<sup>A</sup>T<sub>E</sub>X-a*

Zofia Walczak

Wydział Matematyki UŁ

październik 2006

## 1. Struktura dokumentu w L<sup>A</sup>T<sub>E</sub>X-u

Aby dokument tekstowy został przeczytany i przetworzony przez L<sup>A</sup>T<sub>E</sub>X-a musi posiadać pewną ściśle określoną strukturę postaci:

prolog

```
\documentclass[lista opcji]{nazwa klasy}
```

preambuła

```
\begin{document}
```

Treść dokumentu

```
\end{document}
```

Prolog jest zwykle pusty, mogą się tam znaleźć tylko nieliczne polecenia które zawierają zazwyczaj instrukcje sterujące wykonywane jeszcze przed uruchomieniem T<sub>E</sub>X-a.

Klasa dokumentu to jest pakiet zawierający definicje i deklaracje m. inn. szerokości i wysokości strony, sposobu numerowania rozdziałów itp. Standardowo w L<sup>A</sup>T<sub>E</sub>X-u dostępne są klasy `article`, `book`, `report` i `letter`. Użytkownik może także napisać swoją własną klasę dokumentu. Większość czasopism naukowych przygotowało swoje klasy dokumentów i udostępnia je autorom artykułów na swoich stronach internetowych. Dla użytkowników polskich powstały klasy `mwart`, `mwbk` i `mwrep`, które uwzględniają polskie zwyczaje w piśmiennictwie naukowym.

Argument `Lista opcji`, również nieobowiązkowy, może zawierać informacje o wielkości czcionki, jaką ma być składany cały dokument, rozmiarze papieru, sposobie rozmieszczenia tekstu na stronach (np. dokument jednostronny, dwustronny, ze stroną tytułową lub bez) itd. Jeżeli lista ta będzie pusta przyjęte zostaną wartości i ustawienia domyślne.

Pomiędzy poleceniami `\begin{document}` i `\end{document}` umieszczamy tekst naszego dokumentu. Dokument L<sup>A</sup>T<sub>E</sub>X-owy może być przygotowany w dowolnym edytorze tekstowym, ale należy pamiętać, by plik był zapisany w czystym kodzie ASCII. Nazwa jaką mu nadamy powinna być możliwie krótka, nie może zawierać polskich liter, przerw i powinna

mieć przedłużenie `.tex`. Dlatego też wygodnie jest używać edytorów specjalnie przystosowanych do pracy z  $\text{T}_{\text{E}}\text{X}$ -em.

Należy też pamiętać, że pewne znaki i polecenia są zarezerwowane do specjalnych celów. Są to:

`#`, `$`, `%`, `&`, `\`, `^`, `_`, `{`, `}`.

Znakiem backslash (`\`) rozpoczyna się każda komenda  $\text{T}_{\text{E}}\text{X}$ -owa, znak procentu (`%`) oznacza początek komentarza, znak dolara (`$`) otwiera i zamyka środowisko matematyczne, znaki `^` i `_` oznaczają odpowiednio indeks górny i dolny zaś nawiasy `{`, `}` określają zakres działania danego polecenia.

**Zadanie 1.** Utworzyć dokument składający się z komend `\documentclass`, `\begin{document}`, `\end{document}` i dowolnego tekstu.

W preambule dokumentu, pomiędzy `\documentclass` i `\begin{document}`, umieszcza się wszystkie deklaracje, które mają obowiązywać w całym dokumencie i mają wpływ na końcowy wygląd dokumentu.

W pakietach, które wczytuje się poleceniem `\usepackage{nazwa}` umieszczonym w preambule zawartych jest wiele definicji ułatwiających pisanie w  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -u. W preambule umieszcza się także wszystkie definicje użytkownika, które mają obowiązywać w całym dokumencie. Mogą tam się więc znaleźć takie deklaracje jak `\newcommand`, `\renewcommand`, `\newenvironment` itd. Również tam umieszczamy tytuł dokumentu oraz nazwisko i adres autora.

**Zadanie 2.** Utworzyć dokument składający się z komend `\documentclass` z opisanymi wyżej opcjami (np. wielkość liter - 11pt, rozmiar papieru - a4paper, druk dwustronny - twoside) i datą, `\begin{document}`, `\end{document}` i dowolnego tekstu.

Uwaga: Poszczególne opcje oddzielamy od siebie przecinkiem.

## 2. Tryby pracy $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -a

Podczas pracy  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  znajduje się w jednym z trzech trybów:

1. akapitowym,
2. matematycznym,
3. LR (left-to-right) czyli wierszowym.

$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  znajduje się w trybie akapitowym podczas składania tekstu w wiersze, akapity i strony. Program czyta cały akapit tekstu a następnie dzieli go na wiersze, potem łamie strony itd. Dzięki takiej organizacji łamania udaje się uniknąć niechcianych odstępów np. między wyrazami. Jeżeli w trakcie czytania treści dokumentu  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  natrafi na tekst matematyczny (zawarty pomiędzy znakami dolara) wtedy automatycznie przełącza się na tryb matematyczny. W tryb LR  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  przełącza się np. po napotkaniu komendy `\mbox{...}`. W trybie LR tekst jest umieszczany w jednej linii bez względu na jej długość.

### 3. Odstępy poziome i pionowe

Rozpoczynając pracę z  $\text{\LaTeX}$ -em należy pamiętać, że

- jeden odstęp znaczy tyle samo co wiele, tylko pierwszy jest brany pod uwagę,
- odstępy na początku linii są ignorowane,
- odstęp kończący komendę jest usuwany,
- przejście do nowej linii (CR) jest traktowane jak pojedynczy odstęp,
- nowy akapit rozpoczynamy pozostawiając pustą linię lub umieszczając na początku komendę `\par`.

Dowolnej długości odstęp poziomy możemy uzyskać przy pomocy komendy `\hspace{odległość}` lub `\hspace*{odległość}`. Odległość musi być podana razem z jednostką np. `\hspace{2cm}`. Standardowa forma (bez gwiazdki) nie powoduje umieszczenia dodatkowego odstępu jeśli wypadnie między dwoma liniami (w takim przypadku stosowana jest zasada ignorowania dodatkowych odstępu na początku linii). Forma z gwiazdką powoduje bezwzględne wstawienie odstępu tam gdzie sobie życzymy.

**Zadanie 3.** Umieść w dokumencie tekst identycznie położony na stronie jak poniżej:

To jest odstęp wielkości 2 cm a to tylko 1 cm.  
A to jest odstęp 3 cm na początku linii.

Poziomy odstęp możemy też uzyskać przy pomocy komend `\quad` i `\qquad` oraz umieszczając znak `\` i pustą spację bezpośrednio po nim. Pierwsza z nich dodaje do pojedynczego odstępu równy szerokości litery `m` w bieżącym foncie a druga dwa razy więcej a trzecia standardowy odstęp międzywyrazowy w bieżącym foncie.

Dodatkowe, bardzo małe odstępy można też uzyskać umieszczając bezpośrednio po znaku sterującym dwukropki, przecinek, średnik lub wykrzyknik `\,`, `\;`, `\:`, `\!`. Wszystkie one działają w środowisku matematycznym i są bardzo pomocne przy pisaniu matematycznych wzorów (`\`, można używać również w trybie tekstowym). Poniższa tabelka podaje wielkości odstępu, jakie można przy ich pomocy uzyskać.

<code>\,</code>	mały odstęp	= 3/18 odstępu <code>\quad</code> ,
<code>\:</code>	średni odstęp	= 4/18 odstępu <code>\quad</code> ,
<code>\;</code>	duży odstęp	= 5/18 odstępu <code>\quad</code> ,
<code>\!</code>	odstęp ujemny	= -3/18 odstępu <code>\quad</code> .

#### Przykład 1.

<code>a b</code>	- normalny odstęp między literami
<code>a b</code>	- normalny odstęp między literami i dodatkowy <code>\</code> plus spacja
<code>a \ b</code>	- odstęp większy <code>a\quad b</code>
<code>a \ b</code>	- dodatkowy odstęp (mniejszy niż poprzedni) <code>a \ b</code>
<code>a \quad b</code>	- podwójny dodatkowy odstęp <code>a\qquad b</code>

Komenda `\hfill` będąca skróceniem komendy `\hspace{fill}` powoduje wypełnienie spacjami przestrzeni między wyrazami i "rozepchnięcie" ich do lewego i prawego marginesu, jak w następującym przykładzie.

### Przykład 2.

`\noindent Strona lewa \hfill strona prawa`

Strona lewa strona prawa

Polecenie `\noindent` likwiduje wcięcie akapitowe, które L<sup>A</sup>T<sub>E</sub>X umieszcza na początku każdego akapitu poza rozpoczynający dokument (w klasie `article`).

**Zadanie 4.** Umieść w swoim dokumencie następującą linię:

Strona lewa środek strona prawa

Pustą przestrzeń między wyrazami można wypełnić także kropkami lub ciągłą linią przy pomocy polecenia `\dotfill`, `\hrulefill`, jak w następnym przykładzie.

### Przykład 3.

abecadło ..... z pieca spadło  
o ziemię \_\_\_\_\_ się hukło  
rozsypało się ..... po \_\_\_\_\_ kątach  
strasznie \_\_\_\_\_ się ..... potłukło

W tej samej linii możemy umieścić dowolną kombinację poleceń `\hfill`, `\dotfill`, `\hrulefill`. Jeżeli któraś z nich wystąpi więcej niż jeden raz w tym samym miejscu, to odpowiednie wypełnienie będzie wykonane tylko raz.

### Przykład 4.

Ala `\hfill\hrulefill\hfill` ma kota

Ala \_\_\_\_\_ ma kota

W tym przykładzie odległość między wyrazami Ala i ma została podzielona na trzy części i środkowa została wypełniona kreską.

**Zadanie 5.** Umieść w swoim dokumencie następujący tekst:

Nad rzeczką .....  
Opodal krzaczka \_\_\_\_\_  
Mieszkała Kaczka Dziwaczka  
Lecz zamiast trzymać się \_\_\_\_\_ rzeczki .....  
robiła ..... pieszce wycieczki.

## 4. Style strony

Styl strony to inaczej jej wygląd. Możemy go zadeklarować umieszczając w preambule komendę

`\pagestyle{nazwa stylu}`. Standardowo dostępne są następujące:

- `plain` – główka (head) strony jest pusta, stopka (foot) zawiera wycentrowany numer strony. Ten styl przyjmowany jest jako domyślny jeżeli żaden inny nie jest określony w preambule.
- `empty` – główka i stopka są puste, nie drukują się numery stron.

– headings – główka zawiera numer strony i nazwę i tytuł rozdziału (paragrafu, zależnie od klasy dokumentu). Na stronie z tytułem główka jest pusta. Stopka jest zawsze pusta.

– myheadings – tak samo jak w stylu headings ale trzeba samemu podać treść jaka ma być umieszczona w główce przy pomocy poleceń `\markright{tekst}`, `\markleft{tekst}` lub `\markboth{tekst lewy}{tekst prawy}`. Stopka, tak jak w stylu headings jest pusta.

Jeżeli chcemy, by na jakiejś stronie nie było numeru, wtedy na tej stronie gdziekolwiek w tekście umieszczamy komendę `\thispagestyle{empty}`. Następna strona będzie miała numer kolejny (poprzednia będzie policzona lecz numer nie będzie drukowany).

## 5. Numerowanie stron

Numery stron domyślnie są pisane cyframi arabskimi, możemy jednak zmienić sposób numerowania. Dostępne są następujące style numerowania:

**arabic** – liczby arabskie (domyślnie)

**roman** – liczby rzymskie małe

**Roman** – liczby rzymskie duże

**alph** – małe litery alfabetu łacińskiego

**Alph** – duże litery alfabetu łacińskiego.

Sposób numerowania zmienia się za pomocą komendy `\pagenumbering{styl numerowania}`.

Za numerowanie stron odpowiada licznik o nazwie `page`. Aby go zmienić należy na stronie umieścić komendę `\setcounter{page}{numer}`. Strona na której pojawiła się ta komenda zostaje znumerowana liczbą "numer" a następne strony otrzymują numery kolejne.

**Zadanie 6.** Wypróbuj w swoim dokumencie różne style strony.

**Zadanie 7.** Zmień w swoim dokumencie sposób numerowania stron.

**Zadanie 8.** Zmień w swoim dokumencie numerację stron.

## 6. Format strony

Aby właściwie umieścić tekst na stronie musimy prawidłowo ustawić marginesy i odległości między poszczególnymi elementami strony. W  $\text{\LaTeX}$ -u odpowiedzialne za te ustawienia są następujące komendy:

1. `\oddsidemargin` – ustawia lewy margines na wszystkich stronach, a w przypadku klasy `book` lub kiedy dokument jest składany dwustronnie lewy margines stron nieparzystych. Nową wartość można ustawić komendą `\setlength{\oddsidemargin}{2cm}`.
2. `\evensidemargin` – ustawia lewy margines na stronach parzystych w przypadku klasy `book` lub kiedy dokument jest składany dwustronnie.
3. `\topmargin` – ustawia górny margines do główki.
4. `\headheight` – ustawia wysokość główki.

5. `\headsep` – odległość między główką a tekstem dokumentu.
6. `\topskip` – odległość pierwszej linii tekstu od górnego marginesu.
7. `\textheight` i `\textwidth` – wysokość i szerokość tekstu.
8. `\footskip` - odległość końca tekstu od stopki.
9. `\paperwidth`, `\paperheight` – szerokość i wysokość papieru.

Domyślne wartości powyższych parametrów można zmienić umieszczając w preambule polecenie `\setlength{nazwa}{wielkość}`

**Zadanie 9.** Zmień szerokość i wysokość tekstu w swoim dokumencie.

### 6.1. Tekst w dwu kolumnach

Domyślnie tekst na stronie umieszczony jest w jednej kolumnie. Jeżeli w deklaracji klasy dokumentu umieścimy opcję `twocolumn`, wówczas cały dokument zostanie złożony w dwóch kolumnach. Jeśli chcemy, by tylko część tekstu była złożona w dwóch kolumnach, wówczas umieszczamy na początku tego tekstu polecenie `\twocolumn{tekst}`. Uruchomienie tej komendy powoduje zadziałanie komendy `\pagebreak` i `\newpage`, tj. zakończenie bieżącej strony (kolumny) i rozpoczęcie nowej. Jeżeli dalszy ciąg tekstu ma być również składany w jednej kolumnie, to należy umieścić na końcu tekstu dwukolumnowego (za nawiasem zamykającym) polecenie `\onecolumn`.

**Zadanie 10.** Wypróbuj działanie opcji `twocolumn` i komendy `\twocolumn` w swoim dokumencie.

Jeżeli chcemy aby na jednej stronie można było umieścić różną ilość kolumn to należy skorzystać z pakietu `multicol` napisanego przez Franka Mittelbacha. Definiuje on środowisko `multicols` które pozwala przełączać się z jednej na dwie (i więcej) kolumn na jednej stronie. Składnia jego wywołania jest następująca:

```
\begin{multicols}{ilość kolumn}[przedmowa] [skok]
```

Normalnie rozpoczynamy pisanie w wielu kolumnach przez podanie tylko ich ilości. Może się jednak zdarzyć, że chcemy ten fragment tekstu poprzedzić jakąś dodatkową informacją i nie chcemy, aby znalazła się ona np. na poprzedniej stronie. Wówczas tę informację umieszczamy w miejscu parametr „przedmowa”. Drugi parametr „skok” pozwala nam uniknąć przypadku, gdy nie ma dostatecznie dużo miejsca na stronie i tekst przedmowy byłby oddzielony od części tekstu w kolumnach. Użycie tego środowiska jest możliwe po umieszczeniu w preambule dokumentu polecenia `\usepackage{multicol}`.

#### Przykład 1.

A to praktyczne wykorzystanie możliwości pakietu `multicol`:

Normalnie rozpoczynamy pisanie w wielu kolumnach przez podanie tylko ich ilości. Może się jednak zdarzyć, że chcemy ten fragment tekstu poprzedzić jakąś dodatkową informacją i nie chcemy, aby znalazła się ona np. na poprzedniej stronie. Wówczas tę informację umieszczamy w miejscu parametr „przedmowa”. Drugi parametr „skok” pozwala nam uniknąć przypadku, gdy nie ma dostatecznie dużo miejsca na stronie i tekst przedmowy byłby oddzielony od części tekstu w kolumnach.

napis

Normalnie rozpoczynamy pisanie w wielu kolumnach przez podanie tylko ich ilości. Może się jednak zdarzyć, że chcemy ten fragment tekstu poprzedzić jakąś dodatkową informacją i nie chcemy, aby znalazła się ona np. na poprzedniej stronie. Wówczas tę informa-

cję umieszczamy w miejscu parametr „przedmowa”. Drugi parametr „skok” pozwala nam uniknąć przypadku, gdy nie ma dostatecznie dużo miejsca na stronie i tekst przedmowy byłby oddzielony od części tekstu w kolumnach.

Normalnie rozpoczynamy pisanie w wielu kolumnach przez podanie tylko ich ilości. Może się jednak zdarzyć, że chcemy ten fragment tekstu poprzedzić jakąś dodatkową informacją i nie chcemy, aby znalazła się ona np. na poprzedniej stronie.

Normalnie rozpoczynamy pisanie w wielu kolumnach przez podanie tylko ich ilości. Może się jednak zdarzyć, że chcemy ten fragment tekstu poprzedzić jakąś dodatkową informacją i nie chcemy, aby znalazła się ona np. na poprzedniej stronie. Wówczas tę informację umieszczamy w miejscu parametr „przedmowa”. Drugi parametr „skok” każe  $\TeX$ -owi sprawdzić, czy jest dostatecznie dużo miej-

sca na stronie i tekst przedmowy nie będzie oddzielony od tekstu w kolumnach. Jeżeli miejsca jest za mało, tekst w kolumnach rozpoczyna się na nowej stronie.

## 6.2. Łamanie linii tekstu

$\LaTeX$  łamie linie tekstu automatycznie, ale nie zawsze chcemy, by robił to za nas. Możemy go zmusić do złamania linii w konkretnym miejscu (poleceniem `\linebreak[liczba]` lub nie pozwolić mu jej złamać (poleceniem `\nolinebreak[liczba]`). Argument `liczba` jest opcjonalny i informuje jak ważne jest złamanie linii właśnie w tym miejscu. Może przyjmować wartości od 0 do 4 i im wyższa wartość tym polecenie jest silniejsze. Domyślnie jest przyjęta wartość 4.

Komenda `\` umieszczona w dowolnym miejscu tekstu powoduje złamanie linii w tym miejscu. Dalszy ciąg tekstu umieszczony jest w kolejnej linii. Jeżeli dodamy opcjonalny argument `odległość` (`\[odległość]`) wówczas odległość złamanej linii od następnej jest zwiększona o wielkość `odległość`.

Podobny efekt jak `\` otrzymamy gdy zastosujemy komendę `\newline`, z tym, że nie ma tu możliwości dodania dodatkowego odstępu między wierszami.

Polecenie `\linebreak` powoduje, że tekst w łamanej linii jest wyrównywany do lewego i prawego marginesu poprzez dodanie dodatkowych odstępów międzywyrazowych podczas gdy polecenia `\` i `\newline` powodują, że tekst jest uzupełniany do końca linii pustymi odstępami i odstępy między wyrazami pozostają niezmienione.

## 6.3. Łamanie stron

Podobnie jak z linią możemy postąpić ze stroną tekstu i zażyczyć sobie złamanie jej w konkretnym, wygodnym dla nas miejscu. Służą do tego dwa polecenia: `\pagebreak[liczba]` i `\newpage`. Argument `liczba` jest opcjonalny i informuje jak ważne jest złamanie strony właśnie w tym miejscu. Może przyjmować wartości od 0 do 4 i im wyższa wartość tym polecenie jest silniejsze. Domyślnie jest przyjęta wartość 4. Polecenia te różnią się podobnie jak te dotyczące łamania linii. Po zastosowaniu `\pagebreak` tekst na stronie poprzedzającej

to polecenie zostanie rozmieszczony na całej szerokości i wysokości strony poprzez dodanie pustych linii tak by zaczynał się na górze strony a kończył na dole. Polecenie `\newpage` powoduje zakończenie strony w miejscu jego pojawienia, wypełnienie jej pustymi odstępami i rozpoczęcie nowej.

Możemy również zabronić  $\text{\LaTeX}$ -owi złamania strony w danym miejscu. Polecenie `\nopagebreak` umieszczone między paragrafami zabrania złamania tam strony, a umieszczone wewnątrz paragrafu zabrania złamania strony na końcu bieżącej linii.

**Zadanie 11.** Wypróbuj działanie komend łamiących linie i strony w swoim dokumencie.

## 6.4. Formatowanie paragrafu

Nowy paragraf w dokumencie rozpoczyna pusta linia w tekście lub polecenie `\par`. Odległość między paragrafami jest ustalana poleceniem `\parskip` i możemy ją zmienić komendą `\setlength{\parskip}{odległość}`. W ten sposób zmienimy odległość we wszystkich następujących po tym poleceniu paragrafach w dokumencie. Jeżeli chcemy zmienić odstęp między konkretnymi dwoma paragrafami wtedy możemy użyć polecenia `\space{odległość}` (lub `\space*{odległość}`). Parametr `odległość` musi być liczbą (dodatnią lub ujemną) z mianem (np. 1cm, 3ex, itp). Ujemny parametr wstawia ujemny odstęp (zmniejsza odstęp istniejący). Komenda z gwiazdką wstawia dodatkowy pionowy odstęp od góry strony.

Komenda `\vfill` (skrótowa forma `\space{\fill}`) powoduje wypełnienie strony pustą przestrzenią, a gdy pojawia się u góry strony musi mieć postać z gwiazdką `\space*{\fill}`.

Dodatkowe pionowe odstępy można także uzyskać stosując komendy `\bigskip`, `\medskip` i `\smallskip`. Wielkość odstępu zależy od wielkości zadeklarowanego fontu.

Wielkość wcięcia paragrafu regulowana jest poleceniem `\parindent`. Można ją zmienić w standardowy sposób komendą `\setlength{\parindent}{odległość}`. Aby zlikwidować wcięcie akapitowe możemy użyć komendy `\noindent`, aby wstawić wcięcie możemy użyć komendy `\indent`.

## 6.5. Dzielenie wyrazów

$\text{\LaTeX}$  ma wbudowany algorytm dzielenia wyrazów dla języka angielskiego. Jeżeli używamy polskiego, to razem z pakietem `polski` wczytywany jest też algorytm dzielenia wyrazów w języku polskim. W pojedynczych przypadkach może zaistnieć konieczność ręcznego podzielenia jakiegoś wyrazu. Użyć wtedy można komendy `\-` w miejscu, w którym chcemy dany wyraz podzielić. Możemy też "podpowiedzieć"  $\text{\LaTeX}$ -owi gdzie może długi wyraz podzielić pisząc np. `kon\-\stan\-\ty\-\no\-\po\-\li\-\tańczykiewiczówna`.

Wyrazy, które występują często w dokumencie i są błędnie przenoszone mogą być umieszczone w preambule jako lista wyjątków przy pomocy komendy `\hyphenation{lista}` (często może to dotyczyć nazwisk lub jakichś terminów naukowych). `Lista` to zbiór naszych długich słów oddzielonych od siebie spacjami w których miejsca dopuszczalnego podziału zaznaczone są kreseczką (dywizem).

### Przykład 2.

```
\hyphenation{kar-ko-ło-mny kon-stan-ty-no-po-li-tań-czyk
pod-noś-nik ...}
```



Jeżeli z jakiegoś powodu chcemy aby w części tekstu L<sup>A</sup>T<sub>E</sub>X dzielił jak najmniej wyrazów, możemy umieścić nasz tekst w środowisku `sloppypar` o następującej składni:

```
\begin{sloppypar}
tekst paragrafu
\end{sloppypar}
```

W takim przypadku dodawane są dodatkowe odstępy między wyrazami i linia tekstu jest traktowana jako zbiór poszczególnych wyrazów a nie jako całość a przenoszenie wyrazów zostanie wtedy zredukowane do minimum. Nie pojawiają się też informacje o przekroczeniu dozwolonej szerokości tekstu nawet, jeśli taka sytuacja ma miejsce.

Użycie środowiska `\sloppypar` jest wskazane przy wąskich kolumnach tekstu. Podobnie działa komenda `\sloppy`.

## 7. Wprowadzanie wyrażeń i wzorów matematycznych

L<sup>A</sup>T<sub>E</sub>X przełącza się na tryb matematyczny gdy napotka znak dolara (\$) lub dwóch dolarów (\$\$) a powraca do trybu tekstowego po napotkaniu następnego dolara lub dwóch.

Inaczej składane są wzory wewnątrz akapitu a inaczej wzory eksponowane.

**Uwaga!** Aby wszystkie potrzebne znaki matematyczne były dostępne należy w preambule wczytać pakiet `amssymb`, natomiast by móc skorzystać z definicji używanych przez AMS należy dodatkowo wczytać pakiet `amsmath`.

### Zadanie 12.

W swoim pliku umieść dowolny wzór matematyczny wewnątrz akapitu i ten sam wzór jako eksponowany. Zwróć uwagę na różnice w wyglądzie tych wzorów.

Aby otrzymać wzór eksponowany możemy też umieścić go w środowisku `displaymath` lub pomiędzy znakami `\[` i `\]`. We wszystkich przypadkach nasz wzór nie zostanie zanumerowany.

Jeśli chcemy, by wzór miał numer, do którego moglibyśmy się odwołać należy umieścić go w środowisku `equation`. Sposób użycia tego środowiska pokazuje następujący przykład.

### Przykład 1.

Pisząc

```
\begin{equation}
\forall x \in \mathbb{R} \quad x^2 + x + 1 \geq 0
\end{equation}
```

otrzymujemy

$$\forall x \in \mathbb{R} \quad x^2 + x + 1 \geq 0 \tag{1}$$

### Indeksy górny i dolny, pierwiastki

Większość instrukcji składu wzorów matematycznych odnosi się do jednego następującego po niej znaku. Jeśli więc chcemy, by instrukcja zastosowana była do więcej niż jednego znaku należy wszystkie je umieścić w nawiasach klamrowych.

Indeksy górne i wykładniki otrzymujemy za pomocą znaku `^` a dolne za pomocą znaku `_`.

Pierwiastek kwadratowy z liczby otrzymujemy przy pomocy instrukcji `\sqrt {liczba pod pierwiastkiem}` a pierwiastek stopnia różnego od dwóch instrukcją

`\sqrt[n]{wyrażenie pod pierwiastkiem}`. Znak pierwiastka bez wyrażenia pod nim możemy uzyskać przy pomocy instrukcji `\surd`.

### Przykład 2.

$$\begin{aligned} &\sqrt{\pi R^2 + xyz} \\ &\sqrt[5]{\pi R^2 + xyz} \end{aligned} \tag{2}$$

A teraz tylko sam znak pierwiastka  $\surd$ .

### Podkreślenia i klamry

Matematycy lubią używać różnych sposobów wyróżnień. Najprostsze z nich to podkreślenie otrzymywane komendą `\underline{...}`, umieszczenie poziomej kreski nad wyrażeniem otrzymywane komendą `\overline{...}` oraz umieszczanie poziomych klamer nad i pod wyrażeniem komendą `\overbrace{...}` i `\underbrace{...}`. Zastosowanie tych instrukcji pokazuje przykład. **Przykład 3.**  $\underline{x+y} + \overline{x-y} = 5$ ,

$$\overbrace{n_1 + n_2 + \dots + n_k} \quad \underbrace{a \cdot a \cdot a \cdot \dots \cdot a}_n$$

```
$\underline{x+y} + \overline{x-y} =5,$
$\overbrace{n_1+n_2+ \dots n_k} \quad \quad \quad
\underbrace{a \cdot a \cdot a \cdot \dots \cdot a}_n $
```

Sposób pisania wszelkiego rodzaju znaków matematycznych można znaleźć w tabelach umieszczonych w każdym podręczniku do  $\text{\LaTeX}$ -a.

### Nawiasy

Do składu różnych ograniczników np. nawiasów okrągłych, kwadratowych i sześciennych (klamrowych) można używać symboli z klawiatury lub specjalnych poleceń. Nawiasy okrągłe i kwadratowe wpisujemy bezpośrednio z klawiatury natomiast nawias klamrowy musimy poprzedzić znakiem rozpoczynającym każdą komendę czyli `\`. Tak więc nawias klamrowy otrzymamy pisząc `\{` i `\}`. Sposób wprowadzania innych ograniczniki można znaleźć w tabelach dowolnego poręcznika.

$\text{\LaTeX}$  ma również zdefiniowanych kilka komend, które w ustalony sposób powiększają znaki nawiasów. Są to:

`\big`, `\Big`, `\bigg` i `\Bigg`.

### Przykład 4.

Poniżej umieściliśmy znaki nawiasów okrągłego i klamrowego, otwierających i zamykających, najpierw wielkości bieżącego fontu a następnie powiększone instrukcjami `\big`, `\Big`, `\bigg` i `\Bigg` odpowiednio.

$$\left( \left( \left( \left( \left( \left\{ \left\{ \left\{ \left\{ \left( \right) \right\} \right\} \right\} \right\} \right) \right) \right) \right) \right) \right)$$

Możemy również skorzystać z tego, że L<sup>A</sup>T<sub>E</sub>X potrafi sam dobrać odpowiednio duży nawias. Należy wtedy rodzaj nawiasu poprzedzić komendą `\left` jeśli chcemy lewy nawias i komendą `\right` jeśli chcemy nawias prawy. Polecenia `\left` i `\right` zawsze występują parami i gdy użyliśmy jednego z nich, drugi musi też pojawić się w tej samej linii wzoru. Jeśli jednak nie chcemy mieć znaku nawiasu zamykającego w tej samej linii wzoru co otwierający należy na końcu linii umieścić polecenie `\right.` a na początku następniej `\left.` i dopiero nawias zamykający w odpowiednim miejscu drugiej linii.

### Operatory

Operatory sumy, całki i produktu (iloczynu) wprowadza się w następujący sposób:

$$\begin{aligned}
 & \sum_{n=1}^{\infty} \sum_{k=1}^{100} \\
 & \int_0^x f(x) dx \quad \int_0^{\frac{n}{2}} g(y) dy \\
 & \prod_k X_k
 \end{aligned}$$

Jeżeli sumujemy po więcej niż jednym wskaźniku wtedy należy postąpić jak w pierwszym lub drugim przykładzie:

$$\begin{aligned}
 & \sum_{\substack{0 < i < n \\ 0 < j < m}} K_{ij}(x) \\
 & \sum_{\substack{0 < i < n \\ 0 < j < m}} K_{ij}(x) \quad \sum_{\substack{i \in A \\ 0 < j < m}} K_{ij}(x)
 \end{aligned}$$

### Inne

Symbole Newtona zapisujemy używając polecenia `\binom{}{}`:

$$\binom{n}{k} C_n^k$$

Do nadpisywania znaków nad innymi znakami polecenia `\stackrel{}{}`:

$$x \stackrel{!}{=} y, \quad y \stackrel{?}{>} 6, \quad x_n \stackrel{pw}{\longrightarrow} \infty$$

Do umieszczania znaków lub wyrażeń nad lub pod innymi wyrażeniami lub znakami możemy też użyć poleceń `\overset{}{}` i `\underset{}{}` odpowiednio. Użycie tych poleceń pokazuje następujący przykład.

### Przykład 5.

$$\stackrel{p.w.}{\equiv} a_n \xrightarrow[n \rightarrow \infty]{} b$$

$$\begin{aligned}
 & \overset{p.w.}{\equiv} B \quad \underset{n \rightarrow \infty}{\longrightarrow} b
 \end{aligned}$$

Wielokropek w trybie tekstowym to trzy kropki na linii tekstu. Uzyskujemy je przy pomocy polecenia `\dots`. W trybie matematycznym są dwa rodzaje wielokropka `\ldots` i `\cdots`. Zastosowanie pokazuje przykład.

### Przykład 6.

Teraz będzie przykład wielokropka w trybie tekstowym `...`, a teraz w trybie matematycznym, najpierw kropki na dole czyli `\ldots` a potem pośrodku czyli `\cdots`  $a_1, a_2, \dots, x_1 \cdot x_2 \cdots x_n$ .

### Macierze i wyznaczniki

Do składu macierzy, wyznaczników jak również wyrównywania tekstu w kolumnach można użyć środowiska `array` o następującej składni:

```
$$
\begin{array}{cc}
a_{11} & a_{12} \\
a_{21} & a_{22} \\
a_{31} & a_{32}
\end{array}
$$
```

$$\begin{array}{cc} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{array}$$

W powyższym przykładzie widać, że środowisko `array` należy umieszczać w trybie matematycznym zawsze, kiedy chcemy zapisać macierz lub wyznacznik.

W nawiasie po wywołaniu środowiska umieszczamy parametry określające położenie tekstu w komórce (c - wyrównany, l - wyrównany do lewej, r - wyrównany do prawej). Ilość parametrów musi być równa ilości kolumn macierzy. Każdy element macierzy jest oddzielony od następnego znakiem `&`, koniec wiersza zaznaczony jest znakiem `\\`. Jeżeli elementy macierzy mają być umieszczone pomiędzy nawiasami wtedy przed otwarciem środowiska umieszczamy `\left(` (nawiasy okrągłe) a po jego zamknięciu `\right)`. Zamiast nawiasów okrągłych można umieścić inne nawiasy lub proste kreski, jak w następnym przykładzie.

**Przykład 7.**

$$X = \begin{pmatrix} a_{11} & a_{12} \\ b_{21} & b_{22} \end{pmatrix} \qquad X = \begin{bmatrix} a_{11} & a_{12} \\ b_{21} & b_{22} \end{bmatrix}$$
$$X = \left\{ \begin{array}{cc} a_{11} & a_{12} \\ b_{21} & b_{22} \end{array} \right\} \qquad X = \left| \begin{array}{cc} a_{11} & a_{12} \\ b_{21} & b_{22} \end{array} \right|$$

### Wzory wielolinijkowe

Do pisania wzorów w kilku liniach mamy w  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -u środowisko matematyczne `eqnarray` które powstało na bazie środowiska `array`. Otrzymujemy układ równań wyrównany np. do znaku równości:

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \qquad (3)$$

$$-\frac{x^7}{7!} + \dots \qquad (4)$$

pisząc:

```
\begin{eqnarray}
```

```
\sin x & = & x-\frac{x^3}{3!} + \frac{x^5}{5!}-\backslash
& & -\frac{x^7}{7!} + \cdots
\end{eqnarray}
```

W środowisku `eqnarray` każda linia jest opatrywana kolejnym numerem. Jeśli nie chcemy, by któraś z linii była zanumerowana wtedy przed złamaniem jej umieszczamy polecenie `\nonumber`.

Jeżeli nasz wielolinijkowy wzór ma nie mieć żadnego numeru, wtedy umieszczamy go w środowisku `eqnarray*`

### Ułamki

Polecenie `\frac{}{}` tworzy we wzorach eksponowanych ułamek, w którym czcionka w liczniku i mianowniku jest takiej samej wielkości jak w całym dokumencie. Wzór matematyczny w linii tekstu jest zapisany w postaci ułamka odpowiednio zmniejszonymi czcionkami. Pakiet `amsmath` dostarcza nam polecenia `\tfrac{}{}` i `\dfrac{}{}` które dają nam możliwość wyboru wielkości ułamka (pierwsze - ułamek wielkości tekstu, drugie - takiej wielkości jak we wzorze eksponowanym).

### Przykład 8.

Możemy w linii tekstu napisać duży ułamek  $\frac{2x-4}{y-3}$  przy pomocy polecenia `\dfrac{}{}` zamiast małego ułamka  $\frac{2x-4}{y-3}$  produkowanego przez polecenie `\frac{}{}`.

### Symbole Newtona

Tak samo jak w przypadku ułamków symbole Newtona (też z pakietem `amsmath`) można pisać w dwóch rozmiarach, bez względu na to gdzie się znajdują.

### Przykład 9.

Można umieścić duży symbol w tekście  $\binom{a}{b}$  przy pomocy polecenia `\dbinom{}{}` zamiast małego  $\binom{a}{b}$  który w tekście produkuje polecenie `\binom{}{}`. Można także we wzorze wyeksponowanym napisać mały symbol Newtona poleceniem `\tbinom{}{}` zamiast automatycznie produkowanego dużego poleceniem `\binom{}{}`.

$$\binom{a-b}{c+d} \quad \binom{a-b}{c+d}$$

### Znak całki

Standardowo granice całkowania są umieszczane obok znaku całki. W `amsmath` mamy polecenie `\limits` które umieszcza granice pod i nad znakiem całki.

### Przykład 10.

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

`$$\int_a^b f(x)dx \quad \int\limits_a^b f(x)dx$$`

## 8. Formatowanie strony dokumentu

### 8.1. Wybieramy rozmiar czcionki

Standardową czcionką dla anglojęzycznego dokumentu L<sup>A</sup>T<sub>E</sub>X-owego jest 10-cio punktowa prosta czcionka rodziny Computer Modern Roman. Dostępne są też wielkości 11 i 12 punktów (1 cal = 72,27 pt, 1 cm = ok. 28,45 pt). Aby używać języka polskiego wczytujemy pakiet `inputenc` wraz z opcją kodowania T1 lub QX np. `\usepackage[QX]{inputenc}`.

#### Przykład 1.

Kolejne linie tekstu napisane są czcionkami 10, 11 i 12 punktowymi.

Ala ma kota. Ala ma kota.

Ala ma kota.

## 9. Pudełka

L<sup>A</sup>T<sub>E</sub>X oferuje użytkownikom trzy rodzaje pudełek: LR - "left-right" pudełko w którym zawartość jest umieszczana horyzontalnie od lewej do prawej, pudełko paragrafu i pudełko liniowe.

Dla uzyskania LR pudełka mamy do dyspozycji następujące komendy: `\mbox`, `\makebox`, `\fbox` i `\framebox`. Składnia jest następująca:

```
\mbox{tekst}      \makebox[szerokość][pozycja]{tekst}
\fbox{tekst}      \framebox[szerokość][pozycja]{tekst}
```

Parametr szerokość musi zawierać oprócz liczby także jednostkę, natomiast jako parametr pozycja mogą być podane następujące litery:

- l - wyrównywanie tekstu do lewego marginesu,
- r - wyrównywanie tekstu do prawego marginesu,
- s - rozstrzelenie tekstu po całym pudełku.

Następujący tekst

Zawartość pudełka

Zawartość pudełka

Zawartość pudełka

Zawartość pudełka

otrzymamy pisząc kolejno:

```
\mbox{Zawartość pudełka}
\makebox[2cm]{Zawartość pudełka}
\fbox{Zawartość pudełka}
\framebox[6cm][l]{Zawartość pudełka}
```

W ostatnim pudełku tekst był wyrównany do lewego marginesu. W następnych przykładach wyrównamy go do prawego marginesu i rozstrzelimy poszczególne wyrazy po całej szerokości pudełka odpowiednio umieszczając literę r lub s jako argument "pozycja".

`\framebox[7cm][r]{Zawartość tego pudełka}`

Zawartość tego pudełka

`\framebox[7cm][s]{Zawartość tego pudełka}`

otrzymamy:

Zawartość                      tego                      pudełka

Następny przykład pokazuje jak umieścić mały prostokąt na wyrazie.

`\framebox[2mm]{centered}`

centered

Wykorzystując polecenie `\makebox` możemy też uzyskać znak rzekniętej litery S w następujący sposób:

`\makebox[0pt][1]{/}S`

§

Używając polecenia `\raisebox` możemy podnieść lub opuścić słowa względem linii bazowej. Używa się go w następujący sposób:

Linia bazowa `\raisebox{1ex}{trochę wyżej}` a teraz `\raisebox{-1ex}{nieco niżej}` i z powrotem na tym samym poziomie.

Linia bazowa trochę wyżej a teraz nieco niżej i z powrotem na tym samym poziomie.

W pudełkach możemy umieszczać całe paragrafy używając polecenia `\parbox` lub środowiska `minipage` o następującej składni

`\begin{minipage}[pozycja]{szerokość}tekst\end{minipage}`

b - równanie do dolnego brzegu pudełka.

t - równanie do górnego brzegu pudełka.

`\parbox[pozycja]{szerokość}{tekst}`

Oba te polecenia umieszczają tekst w pudełku którym możemy dowolnie manipulować. Następne przykłady pokażą kilka zastosowań tych poleceń.

Następująco sformatowany tekst:

To jest pudełko o szerokości 3.5 cm, wyśrodkowane pionowo do ...

BIEŻĄCEJ LINII.

Trudno jest właściwie sformatować wąską stronę, w wielu wypadkach otrzymujemy komunikaty o nadmiarach w linii. Polecenie `\sloppy` pozwala wielu z nich unikać.

otrzymaliśmy pisząc

```

\parbox{3.5cm}{\sloppy To jest pudełko o szerokości 3.5 cm,
wyśrodkowane pionowo do}
\hfill BIEŻĄCEJ LINII \hfill
\parbox{5.5cm}{Trudno jest właściwie sformatować wąską
stronę, w wielu wypadkach otrzymujemy komunikaty o nadmiarach w linii.
Polecenie \sloppy pozwala wielu z nich uniknąć.}

```

Aby umieścić tekst w pudełku o z góry zadanej szerokości należy wykorzystać środowisko `minipage`. Następny przykład pokaże tekst w pudełku o szerokości 5 cm.

Środowisko `minipage` tworzy pionowe pudełko o zadanej szerokości w naszym przypadku jest to 5 cm.

Środowisko `minipage` można również łączyć np. z poleceniem `\parbox` co pokazuje następny przykład.

Środowisko `minipage` tworzy pionowe pudełko o zadanej szerokości 5 cm. Położone jest tak, że tekst w ostatniej jego linii leży na wysokości linii bazowej.

W środku wąskie (szerokości 3 cm) pudełko które jest wycentrowane względem linii bazowej

Zawartość pudełka umieszczonego po prawej stronie jest wyrównana do linii bazowej w ten sposób, że jego pierwsza linia z nią się pokrywa. W obu przypadkach używania środowiska `minipage` nie można zapominać o zaznaczeniu pozycji w jakiej ma być umieszczony tekst.

## 10. Pakiet `amsmath`, wzory wielolinijkowe

Pakiet `amsmath` zawiera definicje wielu pożytecznych poleceń, między innymi środowiska ułatwiające pisanie wielolinijkowych wzorów.

### 10.1. Środowisko `gather`

Środowisko `gather` stosuje się do zapisywania wielu wzorów jeden pod drugim. Wzory są umieszczane w osobnych liniach, każdy jest wyśrodkowany i znumerowany. Każda linia, z wyjątkiem ostatniej, ma być zakończona znakiem `\\`. Nie może być pustych linii wewnątrz środowiska.

#### Przykład 1.

$$x^2 + 2xy - y^2 = 3 - xy \quad (5)$$

$$3x - 2y^3 + 13 = 2 - x^2 \quad (6)$$

$$x + y - 1 = 3xy \quad (7)$$



```

\begin{gather}
x^2+2xy-y^2=3-xy\\
3x-2y^3+13=2-x^2\\
x+y-1=3xy
\end{gather}

```

Jeżeli chcemy, aby któraś z linii nie miała numeru, wówczas przed jej złamaniem umieszczamy polecenie `\notag` lub `\nonumber`. Jeśli nie chcemy numerować żadnej z linii, wówczas używamy środowiska `gather*`

## 10.2. Środowisko `multline`

W środowisku `multline` kolejne linie wzoru rozmieszczone są w taki sposób, że pierwsza jest dosunięta do lewego marginesu, ostatnia do prawego a środkowe w miarę możliwości wycentrowane. Wzór jest numerowany jednym numerem umieszczonym w ostatniej linii. Jeśli nie chcemy numerować wzoru to umieszczamy go w środowisku z gwiazdką (`multline*`). Każda linia, z wyjątkiem ostatniej, ma być zakończona znakiem `\\`. Nie może być pustych linii wewnątrz środowiska.

### Przykład 2.

$$\begin{aligned}
 x_1 + x_2 + x_3 + 2k(x_1 + x_2 + x_3) + 3z(x_1 + x_2 + x_3) \\
 + x_4 + x_5 + x_6 + x_7 - 4y(x_1 + x_2 + x_3) \\
 + x_8 + x_9 - z(x_1 + x_2 + x_3) \quad (8)
 \end{aligned}$$

```

\begin{multline}
x_1+x_2+x_3 +2k(x_1+x_2+x_3)+3z(x_1+x_2+x_3)\\
+x_4+x_5+x_6+x_7 -4y(x_1+x_2+x_3)\\
+x_8+x_9 -z(x_1+x_2+x_3 )
\end{multline}

```

## 10.3. Środowisko `align`

Środowisko `align` pozwala na napisanie wzorów wyrównanych w kolumnach. Odstępy między kolumnami są dopasowywane przez program według z góry zadanych reguł.

### Przykład 3.

$$x + y + z = 23 \quad (9)$$

$$2x - 3y + z = x^2 - 4y^2 - 1 \quad (10)$$

$$y^4 - 4z^2 = 21 \quad (11)$$

```

\begin{align}
x+y+z &= 23\\
2x-3y+z &=x^2-4y^2-1\\
y^4-4z^2 &=21
\end{align}

```

Znak `&` wskazuje miejsce wyrównywania, każda linia musi być złamana znakiem `\\`. Każda linia jest opatrzona numerem. Jeśli nie chcemy w ogóle numerować wzoru, wtedy mamy do dyspozycji formę `align*`, natomiast gdy chcemy zanumerować tylko jedną linię, wtedy w pozostałych, przed zakończeniem linii umieszczamy polecenie `\nonumber` lub `\notag`.

## 10.4. Środowisko `alignat`

Środowisko `alignat` również pozwala na rozmieszczenie wzorów w wyrównanych kolumnach, jednak, inaczej niż w przypadku `align`, samo nie decyduje o wielkości odstępu między wyrównywanymi kolumnami. Użytkownik musi tę odległość podać sam. Należy również podać, w ilu kolumnach będziemy umieszczać i wyrównywać wyrażenia. Każda linia jest zaopatrzona w numer, jeśli nie chcemy w ogóle numerować wzoru, wtedy mamy do dyspozycji formę `alignat*`, natomiast gdy chcemy zanumerować tylko jedną linię, wtedy, jak w innych środowiskach, przed zakończeniem linii umieszczamy polecenie `\nonumber` lub `\notag`. Środowisko to jest szczególnie użyteczne gdy chcemy uzyskać kilka kolumn z krótkimi, wyrównanymi równościami.

**Przykład 4.**

$$f(x) = x + 2y \quad g(x) = x - y \quad (12)$$

$$h(x) = x^2 + 1 \quad k(x) = y^2 \quad (13)$$

```
\begin{alignat}{2}
f(x) &= x+2y & \quad & \quad g(x) = x-y \\
h(x) &= x^2+1 & \quad & \quad k(x) = y^2
\end{alignat}
```

**Uwaga:** Jeżeli wyrównywanych jest  $n$  kolumn w środowisku `alignat`, to wówczas musimy umieścić w każdej linii  $2n - 1$  znaków `&`.

## 10.5. Środowisko `cases`

Korzystając z możliwości pakietu `amsmath` bardzo łatwo napiszemy następujący wzór:

$$f(x) = \begin{cases} x & \text{gdy } x > 5 \\ 0 & \text{gdy } 0 \geq x \geq 5 \\ -x & \text{dla pozostałych} \end{cases} \quad (14)$$

```
\begin{equation}
f(x)=
\begin{cases}
x & \text{gdy } \$x>5\$ \\
0 & \text{gdy } \$0\ge x\ge 5\$ \\
-x & \text{dla pozostałych}
\end{cases}
\end{equation}
```

Po przeanalizowaniu powyższego przykładu zauważamy, że użyte zostały dwa nowe polecenia. Pierwsze to wywołanie środowiska `cases` a drugie to polecenie `\tekst`. Jest ono równoważne poleceniu `\textrm` w L<sup>A</sup>T<sub>E</sub>X-u. Środowisko `cases` jest szczególnym przypadkiem środowiska `array`.

## 11. Pakiet amscd

Pisanie różnego rodzaju diagramów umożliwia pakiet `amscd`. Aby otrzymać np. diagram następujący

$$\begin{array}{ccc} A & \longrightarrow & B \\ \downarrow & & \downarrow \\ D & \longequal{\quad} & H \end{array}$$

musimy, po wczytaniu w preambule dokumentu pakietu napisać:

```

 $\mathbb{S}$ 
\begin{CD}
A @>>> B \\
@VVV @VVV \\
D @= H
\end{CD}
 $\mathbb{S}$ 

```

Ten diagram jest tablicą składającą się z dwóch rodzajów rzędów: z poziomych ze strzałkami poziomymi i pionowych ze strzałkami pionowymi. Poziome strzałki otrzymujemy w następująco:

- `@>>>` - pozioma strzałka skierowana w prawo,
- `@<<<` - pozioma strzałka skierowana w lewo,
- `@=` - dwie poziome linie,
- `@—` - dwie linie pionowe,

a strzałki pionowe następująco:

- `@VVV` - pionowa strzałka skierowana w dół,
- `@AAA` - pionowa strzałka skierowana w górę.

W każdym przypadku możemy umieścić nad strzałkami dowolną etykietę umieszczając ją pomiędzy znakami `;` lub `|` oraz między literami `V` lub `A`. Użycie opisanych poleceń pokażemy na przykładzie.

Aby rysować bardziej skomplikowane diagramy należy użyć albo programu rysującego albo pakietów specjalistycznych, które można znaleźć w archiwach CTAN.

### Przykład 1.

$$\begin{array}{ccccc} A & \longrightarrow & B & & \\ \downarrow & & & & \downarrow \\ D & \longequal{\quad} & H & & \\ \\ A & \xrightarrow{G} & B & \xleftarrow{Z_1} & R \\ \downarrow n & & k \uparrow & & j \downarrow \\ D & \longequal{\quad} & M & \xleftarrow{\alpha} & H \end{array}$$

Powyższe diagramy otrzymaliśmy następująco:

```

$$
\begin{CD}
A @>>> B \\
@VVV @VVV \\
D @= H
\end{CD} \quad
$$
\vspace{.3cm}
$$
\begin{CD}
A @>G> B @<Z_1<< R \\
@VVnV @AkAA @VjVV \\
D @= M @<<{\alpha}<< H
\end{CD}
$$

```

A teraz jeszcze inny przykład prostego diagramu.

**Przykład 2.**

$$\begin{array}{ccccc}
 A & \xrightarrow{\quad G \quad} & B & \xleftarrow{\quad Z_1 \quad} & R & \longrightarrow & H \\
 \downarrow n & & \uparrow k & & & & \downarrow j \\
 D & \xlongequal{\quad} & M & \longleftarrow & & & H
 \end{array}$$

## 12. Tworzenie list

W L<sup>A</sup>T<sub>E</sub>X-u dostępne są trzy środowiska tworzące listy: **itemize**, **enumerate** i **description**. Tworzy się je w sposób następujący

```

\begin{itemize}
\item tekst
\item tekst
.....
\end{itemize}

\begin{enumerate}
\item tekst
\item tekst
.....
\end{enumerate}

\begin{description}
\item tekst

```

```
\item tekst
.....
\end{description}
```

W każdym z tych środowisk tekst jest wcięty od lewego marginesu i poprzedzony etykietą lub numerem, w zależności od środowiska.

## 12.1. Środowisko `itemize` i `enumerate`

W środowisku wypunktowującym (`itemize`) tekst kolejnych punktów poprzedzony jest grubą czarną kropką jako etykietą. Kropka taka jest wywoływana w T<sub>E</sub>X-u poleceniem `\bullet` (w trybie matematycznym). Każdy następny punkt jest oddzielony od poprzedniego dodatkowym odstępem pionowym. A oto przykład działania tego środowiska.

### Przykład 1.

- To jest kilka zdań napisanych w celu pokazania działania różnych środowisk typu lista.
- To jest kilka zdań napisanych w celu pokazania działania różnych środowisk typu lista.
- To jest kilka zdań napisanych w celu pokazania działania różnych środowisk typu lista.

A oto jak otrzymaliśmy tę listę:

```
\begin{itemize}
\item To jest kilka zdań napisanych w celu pokazania działania
różnych środowisk typu lista.
\item To jest kilka zdań napisanych w celu pokazania działania
różnych środowisk typu lista.
\item To jest kilka zdań napisanych w celu pokazania działania
różnych środowisk typu lista.
\end{itemize}
```

W środowisku numerującym (`enumerate`) tekst kolejnych punktów poprzedzony jest kolejnym numerem jako etykietą. Każdy następny punkt jest oddzielony od poprzedniego dodatkowym odstępem pionowym. A oto przykład działania tego środowiska.

1. To jest punkt pierwszy.
2. To jest punkt drugi.
3. To jest punkt trzeci.
4. To jest punkt czwarty.

A oto jak otrzymaliśmy tę listę

```
\begin{enumerate}
\item To jest punkt pierwszy.
\item To jest punkt drugi.
\item To jest punkt trzeci.
\item To jest punkt czwarty.
\end{enumerate}
```

W obu tych środowiskach możemy zmienić etykiety umieszczając własną w nawiasach kwadratowych po wywołaniu polecenia `\item` na przykład `\item[A]`:

A To jest punkt pierwszy.

1. To jest punkt drugi.

b To jest punkt trzeci.

2. To jest punkt czwarty.

lub

• To jest kilka zdań napisanych w celu pokazania działania różnych środowisk typu lista.

H. To jest kilka zdań napisanych w celu pokazania działania różnych środowisk typu lista.

\* To jest kilka zdań napisanych w celu pokazania działania różnych środowisk typu lista.

W obu tych środowiskach każdy kolejny punkt jest od poprzedniego oddalony nieco więcej niż wynosi odstęp między wierszami. Za wielkość tego odstępu odpowiada parametr `\itemsep` który możemy zmienić w zależności od potrzeb. Aby pozostawić tylko standardowy odstęp międzywierszowy możemy użyć polecenia `\setlength` w następujący sposób

```
\setlength{\itemsep}{0ex plus 0.2ex minus 0.1ex}
```

i umieścić go zaraz po wywołaniu środowiska poleceniem `begin` (patrz następny przykład)

### **Przykład 2.**

Pierwsza lista jest utworzona bez ingerencji w wielkość odstępu a druga po zlikwidowaniu odstępu dodatkowego.

1. To jest punkt pierwszy.

2. To jest punkt drugi.

3. To jest punkt trzeci.

4. To jest punkt czwarty.

1. To jest punkt pierwszy.

2. To jest punkt drugi.

3. To jest punkt trzeci.

4. To jest punkt czwarty.

## 12.2. Środowisko description

Środowisko **description** umożliwia nam umieszczenie tekstu jako etykiety kolejnych punktów listy.

**temat** To jest kilka zdań napisanych w celu pokazania działania różnych środowisk typu lista.

**przykład** To jest kilka zdań napisanych w celu pokazania działania różnych środowisk typu lista.

**inne zastosowanie** To jest kilka zdań napisanych w celu pokazania działania różnych środowisk typu lista.

Oczywiście taka lista nie wygląda najlepiej ale z reguły nie umieszczają się długiego tekstu jako etykiety. Za rozmiary etykiety odpowiada parametr `\labelwidth` i można go zmienić poleceniem `\setlength`.

Możemy również zdefiniować sobie swoją własną listę określając typ etykiety, jej szerokość, rozmiar wcięcia, odstępy między poszczególnymi punktami itd. Możemy wtedy użyć środowiska `list` w następujący sposób

```
\begin{list}{rodzaj-etykiety}{lista-deklaracji}
kolejne punkty listy \end{list}
```

Parametr *rodzaj-etykiety* zawiera definicję etykiety, a *lista-deklaracji* zawiera ustawienia parametrów opisujących naszą listę.

### Przykład 3.

Poniżej trzy przykłady list o różnych etykietach.

```

* To jest punkt pierwszy. \newcounter{numer}
                           \begin{list}{*}
* To jest punkt drugi.   {\usecounter{numer}}
                           \item To jest punkt pierwszy.
* To jest punkt trzeci.  \item To jest punkt drugi.
                           \item To jest punkt trzeci.
* To jest punkt czwarty. \item To jest punkt czwarty.
                           \end{list}
                           \begin{list}
⇒-1 To jest punkt pierwszy. {\$\Rightarrow$\-arabic{numer}}
                           {\usecounter{numer}}
⇒-2 To jest punkt drugi.   \item To jest punkt pierwszy.
                           \item To jest punkt drugi.
⇒-3 To jest punkt trzeci.  \item To jest punkt trzeci.
                           \end{list}
                           \newcounter{numerek}
A-I To jest punkt pierwszy. \begin{list}{A--\Roman{numerek}}
                           {\usecounter{numerek}}
A-II To jest punkt drugi.  \item To jest punkt pierwszy.
                           \item To jest punkt drugi.
A-III To jest punkt trzeci. \item To jest punkt trzeci.
                           \end{list}
```

### 12.3. Listy zagnieżdżone

Wszystkie środowiska możemy zagnieżdżać jedno w drugim aż do czwartego poziomu włącznie.

- To będzie poziom pierwszy.
    - A to już poziom drugi.
    - I jeszcze raz poziom drugi.
      - \* I poziom trzeci.
        - I na koniec poziom czwarty.
        - I jeszcze raz poziom czwarty.
      - \* I znowu poziom trzeci.
    - Wracamy na drugi poziom
  - I wreszcie na poziomie pierwszym.
- 

1. To będzie poziom pierwszy.
    - (a) A to już poziom drugi.
    - (b) I jeszcze raz poziom drugi.
      - i. I poziom trzeci.
        - A. I na koniec poziom czwarty.
        - B. I jeszcze raz poziom czwarty.
      - ii. I znowu poziom trzeci.
    - (c) Wracamy na drugi poziom
  2. I wreszcie na poziomie pierwszym.
- 

- To będzie poziom pierwszy.
  1. A to już poziom drugi.
  2. I jeszcze raz poziom drugi.
    - I poziom trzeci.
      - (a) I na koniec poziom czwarty.
      - (b) I jeszcze raz poziom czwarty.
    - I znowu poziom trzeci.
  3. Wracamy na drugi poziom
- I wreszcie na poziomie pierwszym.

Przedstawione wyżej listy powstały w sposób następujący:  
Pierwsza:



```

\begin{itemize}
  \item To będzie poziom pierwszy.
\begin{itemize}
\item A to już poziom drugi.
\item I jeszcze raz poziom drugi.
  \begin{itemize}
\item I poziom trzeci.
  \begin{itemize}
    \item I na koniec poziom czwarty.
    \item I jeszcze raz poziom czwarty.
  \end{itemize}
  \end{itemize}
  \item I znowu poziom trzeci.
  \end{itemize}
\item Wracamy na drugi poziom
\end{itemize}
  \item I wreszcie na poziomie pierwszym.
\end{itemize}

```

Druga:

```

\begin{enumerate}
  \item To będzie poziom pierwszy.
\begin{enumerate}
\item A to już poziom drugi.
\item I jeszcze raz poziom drugi.
  \begin{enumerate}
\item I poziom trzeci.
  \begin{enumerate}
    \item I na koniec poziom czwarty.
    \item I jeszcze raz poziom czwarty.
  \end{enumerate}
  \end{enumerate}
  \item I znowu poziom trzeci.
  \end{enumerate}
\item Wracamy na drugi poziom
\end{enumerate}
  \item I wreszcie na poziomie pierwszym.
\end{enumerate}

```

I trzecia:

```

\begin{itemize}
  \item To będzie poziom pierwszy.
\begin{enumerate}
\item A to już poziom drugi.
\item I jeszcze raz poziom drugi.
  \begin{itemize}
\item I poziom trzeci.
  \begin{enumerate}
    \item I na koniec poziom czwarty.
    \item I jeszcze raz poziom czwarty.
  \end{enumerate}
  \end{itemize}
\end{enumerate}

```

```

        \item I znowu poziom trzeci.
    \end{itemize}
\item Wracamy na drugi poziom
\end{enumerate}
    \item I wreszcie na poziomie pierwszym.
\end{itemize}

```

## 12.4. Zmiana etykiet

Etykiety są generowane przez komendy wewnętrzne dla każdego środowiska oddzielnie. Dla środowiska `itemize` są to komendy

```
\labelitemi, \labelitemii, \labelitemiii, \labelitemiv,
```

a dla środowiska `enumerate` komendy

```
\labelenumi, \labelenumii, \labelenumiii, \labelenumiv,
```

Ilość 'i' na końcu komendy pokazuje do którego poziomu ona się odnosi. Aby zmienić etykietę listy `itemize` z czarnej kropki na przykład na znak plus należy użyć komendy `\renewcommand` w następujący sposób:

```
\renewcommand{\labelitemi}{+}
```

Wówczas na pierwszym poziomie zamiast kropki będziemy mieli znak plus. Etykiety na innych poziomach pozostaną niezmienione.

- + Tym razem każdy element listy poprzedzony jest znakiem plus jako etykietą.
- + Wprowadzany tekst może być dowolnej długości. Etykieta pojawia się na początku pierwszej linii.
- + Kolejne elementy listy są od siebie oddzielone dodatkowym odstępem pionowym.

- + To będzie poziom pierwszy ze znakiem + jako etykietą.
  - A to już poziom drugi.
  - I jeszcze raz poziom drugi.
    - \* I poziom trzeci.
      - I na koniec poziom czwarty.
      - I jeszcze raz poziom czwarty.
    - \* I znowu poziom trzeci.
  - Wracamy na drugi poziom
- + I wreszcie na poziomie pierwszym.

Podobnie możemy zmieniać standardowe etykiety w środowisku `enumerate`. Wartość licznika może być zmieniona przy pomocy następujących komend:

```
\arabic, \Alph, \alph, \roman, \Roman.
```

**Przykład 4.**

Aby zmienić etykietę pierwszego poziomu numerowania na numer i nawias ('1') należy przededefiniować licznik w następujący sposób:

```
\renewcommand{\labelenumi}{\arabic{enumi}}
```

Otrzymamy wtedy:

- 1) Etykietami elementów listy są kolejne liczby.
- 2) Numerowanie zaczyna się od 1 za każdym użyciem tego środowiska.

**Zadanie 13.**

Zmienić tak etykiety środowiska `itemize`, aby na pierwszym poziomie była długa kreska (m-dash) otrzymywana z trzech krótkich kresek, na drugim poziomie średnio długa (n-dash, z dwóch krótkich kresek) i na trzecim hyphen - jedna krótka kreska.

**Zadanie 14.**

Zmień tak środowisko `enumerate`, aby pierwszy poziom był numerowany (I), (II) itd., drugi I-1, I-2, ... II-1, II-2, itd.

**Zadanie 15.**

Zmień tak środowisko `enumerate`, aby etykieta składała się ze znaku § i numeru z kropką.

- §1. Etykietami elementów listy są kolejne liczby.
- §2. Numerowanie zaczyna się od 1 za każdym użyciem tego środowiska.

Aby wrócić do standardowych etykiet należy powtórzyć komendę

```
\renewcommand{\labelenumi}{\theenumi.}
```

Następne etykiety będą już standardowe.

1. Etykietami elementów listy są kolejne liczby.
2. Numerowanie zaczyna się od 1 za każdym użyciem tego środowiska.

**Zadanie 16.**

Napisać numerowaną listę w sposób następujący:

**I. Wstęp****A. Zastosowania**

Motywacje do prowadzenia badań w pokrewnych dziedzinach.

**B. Organizacja**

Wyjaśnienie celu badań i sposobu ich przeprowadzenia.

**II. Spis literatury****III. Proponowane tematy**

### 13. Zmiana sposobu numerowania. Domyślny sposób numerowania

Na kolejnych stronach pokażemy domyślny sposób numerowania wzorów i części dokumentu stosowany przez L<sup>A</sup>T<sub>E</sub>X. Będziemy korzystać tylko z jego możliwości i ustawień klasy `article` (w klasie `mwart` niektóre numery mogą wyglądać inaczej), a następnie będziemy zmieniać sposób numerowania.

**Uwaga:** Tekst ilustrujący temat jest napisany pismem prostym zaś tekst wypełniający rozdziały kursywą.

$$\langle \mathbf{e}_1, -\mathbf{e}_0 \rangle, \langle \mathbf{e}_2, -\mathbf{e}_0 \rangle, \langle \mathbf{e}_2, -\mathbf{e}_1 \rangle, \langle -\mathbf{e}_1, \mathbf{e}_0 \rangle, \langle -\mathbf{e}_2, \mathbf{e}_0 \rangle, \langle \mathbf{e}_1, -\mathbf{e}_2 \rangle. \quad (15)$$

### 14. Następny rozdział

*Teraz będzie trochę tekstu aby wypełnić drugi rozdział. Teraz będzie trochę tekstu aby wypełnić drugi rozdział. Teraz będzie trochę tekstu aby wypełnić drugi rozdział. Teraz będzie trochę tekstu aby wypełnić drugi rozdział.*

*Teraz będzie trochę tekstu aby wypełnić drugi rozdział. Teraz będzie trochę tekstu aby wypełnić drugi rozdział. Teraz będzie trochę tekstu aby wypełnić drugi rozdział. Teraz będzie trochę tekstu aby wypełnić drugi rozdział.*

$$a_{ij} = \begin{cases} 1 & h(R_i) \cap R_j \neq \emptyset, \\ 0 & h(R_i) \cap R_j = \emptyset, \end{cases} \quad (16)$$

*I jeszcze trochę tekstu. I jeszcze trochę tekstu. I jeszcze trochę tekstu. I jeszcze trochę tekstu. I jeszcze trochę tekstu. I jeszcze trochę tekstu. I jeszcze trochę tekstu. I jeszcze trochę tekstu.*

$$A = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}. \quad (17)$$

Jak widać wzory numerowane są po kolei pojedynczym numerem. Zobaczmy czy numeracja się zmieni gdy do dokumentu dodamy podrozdział.

#### 14.1. To jest podrozdział

*I znowu nieco tekstu tym razem w podrozdziale. I znowu nieco tekstu tym razem w podrozdziale. I znowu nieco tekstu tym razem w podrozdziale. I znowu nieco tekstu tym razem w podrozdziale. I znowu nieco tekstu tym razem w podrozdziale. I znowu nieco tekstu tym razem w podrozdziale. I znowu nieco tekstu tym razem w podrozdziale.*

$$\{\alpha_1 \leq 0\}, \{\alpha_2 \geq 0\}, \quad (18)$$

Nic się nie zmieniło.

**Wprowadzenie następnego poziomu organizacji tekstu nie powoduje zmiany w stylu numerowania wyrażeń matematycznych.**

Zmienimy teraz numerowanie wzorów tak, aby w każdym rozdziale numeracja zaczynała się od początku. Za numerowanie wzorów odpowiada licznik (counter) który dla wzorów umieszczonych w środowisku `equation` nazywa się `equation`, należy więc zastosować polecenie `\setcounter{equation}{numer}`.

Aby zmienić sposób numerowania tak, by każdy wzór miał numer składający się dwóch liczb: numeru rozdziału i kolejnego numeru wzoru należy przeddefiniować polecenie `\theequation`.

Do przeddefiniowywania poleceń T<sub>E</sub>X-a służy komenda o następującej składni `\renewcommand{co_zmieniamy}{na_co_zmieniamy}`.

Należy więc napisać:

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

Za numerowanie części dokumentu (section) odpowiada polecenie `\thesection` które też zmieniamy komendą `\renewcommand`.

Jeżeli licznik dla wzorów i dla rozdziałów ustawimy poleceniami:

```
\setcounter{section}{numer}
\setcounter{equation}{numer}
```

to wzory będą numerowane podwójnie — numerem rozdziału i numerem kolejnym.

Wracamy teraz do sposobu numerowania stosowanego od początku w tym skrypcie.

Aby otrzymać kropkę po numerze rozdziału przeddefiniujemy polecenie `\thesection`

```
\renewcommand{\thesection}{\arabic{section}.}
```

Aby wzory były numerowane podwójnie przeddefiniujemy polecenie `\theequation`

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

## 15. Inna numeracja wzorów

Ala

$$\{\alpha_1 \leq 0\}, \{\alpha_2 \geq 0\}, \quad (15.19)$$

A to inna równość

$$A = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}. \quad (15.20)$$

## 16. I jeszcze jeden rozdział ze wzorami

$$\{\alpha_1 \leq 0\}, \{\alpha_2 \geq 0\}, \quad (16.21)$$

To działa też dla wzorów wielolinijkowych.

$$a_{ij} = \begin{cases} 1 & h(R_i) \cap R_j \neq \emptyset, \\ 0 & h(R_i) \cap R_j = \emptyset, \end{cases} \quad (16.22)$$

## 17. To jeszcze jeden przykładowy rozdział

$$\{\alpha_1 \leq 0\}, \{\alpha_2 \geq 0\}, \quad (17.23)$$

To działa też dla wzorów wielolinijkowych.

$$a_{ij} = \begin{cases} 1 & h(R_i) \cap R_j \neq \emptyset, \\ 0 & h(R_i) \cap R_j = \emptyset, \end{cases} \quad (17.24)$$

**Zadanie 17.** Zanumerować wzory trzema liczbami: numerem rozdziału, podrozdziału i kolejnym tak jak w przykładzie poniżej.

**Przykład 1.**

### 17.1. To jest podrozdział

I numerowanie wzorów trzema liczbami.

$$\{\alpha_1 \leq 0\}, \{\alpha_2 \geq 0\}, \quad (17.1.25)$$

**Zadanie 18.** Zanumerować kolejne rozdziały dużymi literami.

A teraz wzory będą numerowane trzema liczbami: numerem rozdziału, podrozdziału i kolejnym numerem wzoru.

$$\{\alpha_1 \leq 0\}, \{\alpha_2 \geq 0\}, \quad (17.17.1.26)$$

I jeszcze jeden wzór

$$\Gamma_{(\dots ikik\dots)} = \alpha_k - \alpha_i, (k = 1, \dots, i-1, i+1, \dots, n), \quad (17.17.1.27)$$

$$\Gamma_{(\dots ii\dots)} = -\alpha_i \quad (17.17.1.28)$$

**Czytelnik zgadnie, dlaczego wzory są numerowane czterema cyframi i co zrobić, by numerowane były pojedynczo jak poniżej.**

$$\{\alpha_1 \leq 0\}, \{\alpha_2 \geq 0\}, \quad (29)$$

## 18. Struktury definiowane przez użytkownika

W literaturze matematycznej znajdujemy różnego rodzaju twierdzenia, definicje, uwagi itp. zawierające ważne informacje dotyczące prezentowanej treści. Do składania tekstu twierdzeń i definicji możemy używać środowiska zdefiniowanego przez polecenie `\newtheorem` w następujący sposób:

```
\newtheorem{typ_środowiska}{etykieta}[z licznikiem]
```

W tak określonym środowisku etykieta i numer są pogrubione i proste a tekst pisany jest czcionką pochyłą (italikiem). Domyślnie każde środowisko utworzone poleceniem `\newtheorem` jest numerowane osobno i w sposób ciągły w całym dokumencie.

Jeżeli chcielibyśmy, by nasze nowe środowisko (np. twierdzenie) było numerowane od początku w każdym nowym rozdziale wtedy jako argument [z licznikiem] powinniśmy umieścić wyraz `section` lub `chapter`.

Niekiedy chcemy umieścić obok etykiety środowiska również jego dodatkową nazwę. Wtedy tę nazwę umieszczamy w nawiasach kwadratowych bezpośrednio po otwarciu środowiska.

**Twierdzenie 1 (Twierdzenie o kocie)** *Ala ma kota.*

```
\begin{thm}[Twierdzenie o kocie]
Ala ma kota.
\end{thm}
```

W literaturze matematycznej przyjęte jest by treść twierdzeń, lematów czy spostrzeżeń różniła się krojem pisma od treści definicji. Wówczas po otwarciu środowiska musimy umieścić polecenie zmieniające domyślną czcionkę (italik) na czcionkę prostą. Możemy to zrobić pisząc `\rm{ tekst}`, `\textup{ tekst}` lub `\textrm{ tekst}`.

**Twierdzenie 2** *Ala ma kota.*

**Definicja 1** Król to jest bardzo ważna osoba w państwie a paż mu towarzyszy.

**Lemat 3 (Bardzo ważny)** *Król to jest bardzo ważna osoba w państwie a paż mu towarzyszy.*

```
\begin{thm}
Ala ma kota.
\end{thm}
\begin{defin}\rm{
Król to jest bardzo ważna osoba w państwie
a paż mu towarzyszy.}
\end{defin}
\begin{lemm}[Bardzo ważny]
Król to jest bardzo ważna osoba w państwie
a paż mu towarzyszy.
\end{lemm}
```

Powyższe przykłady pokazują skutki użycia opisanych wyżej poleceń. Wcześniej, w preambule zostały zdefiniowane środowiska definicji (`defin`), lematu (`lemm`) i twierdzenia (`thm`). Lematy mają być numerowane po kolei łącznie z

twierdzeniami. Po rozpoczęciu nowego rozdziału numeracja twierdzeń i lematów rozpocznie się od nowa natomiast definicje będą numerowane dalej. Aby to osiągnąć w preambule umieściliśmy następujące definicje:

```
\newcounter{thm}[section]
\newtheorem{thm}{Twierdzenie}
\newtheorem{defin}{Definicja}
\newtheorem{lemm}[thm]{Lemat}
```

Domyślnie każde środowisko zdefiniowane przy pomocy polecenia `\newtheorem` numerowane jest oddzielnie. Jeżeli chcemy, by dwa lub więcej było numerowanych wspólnie, nazwę licznika musimy umieścić w nawiasach kwadratowych między nazwą środowiska a jego etykietą:

```
\newcounter{aaa}
\newtheorem{aaa}{Aksjomat}
\newtheorem{bb}[aaa]{Uwaga}
```

Otrzymamy wtedy

**Aksjomat 1** *To jest aksjomat pierwszy o numerze 1.*

**Uwaga 2** *A to uwaga pierwsza o numerze 2.*

**Aksjomat 3** *A to aksjomat drugi o numerze 3.*

**Uwaga 4** *A to uwaga druga ale już o numerze 4.*

Oczywiście możemy w znany już sposób zmienić wygląd licznika (np. dodać mu kropkę na końcu).

## 19. Definiowanie własnych komend i środowisk

**Twierdzenie 4** *Ala ma kota i psa.*

**Definicja 2** Król i paż poszli spać.

**Lemat 5 (Też istotny)** *Kot i pies nie lubią się.*

Aby skrócić czas potrzebny do wpisania poleceń otwierających i/lub zamykających środowisko lub innych poleceń o długich nazwach można zdefiniować swoją komendę. Składnia takiej definicji jest następująca:

```
\newcommand{\nowa_nazwa}{nazwa_poprzednia}
```

Można na przykład zdefiniować następująco otwarcie i zamknięcie środowiska `equation`



```
\newcommand{\be}{\begin{equation}}
\newcommand{\ee}{\end{equation}}
```

i wtedy otwieramy środowisko `\be` a zamykamy `\ee`. Należy jednak uważać by nie nadać nazwy już istniejącej. Jeżeli tak się zdarzy, otrzymamy od programu stosowny komunikat.

Komendy które mają być używane w całym dokumencie umieszcza się w *preamble*. Umieszczone wewnątrz pewnego środowiska działają tylko w nim.

Użytkownik może zdefiniować własne środowisko poleceniem `\newenvironment`. Istniejące środowisko zmieniamy poleceniem `\renewenvironment`.

Środowiska zdefiniowane przez użytkownika (tak jak i nowe polecenia zdefiniowane przez `\newcommand`) mogą posiadać do 9 dodatkowych argumentów. Nowe środowisko i nowe polecenie z dodatkowymi argumentami definiujemy następująco:

```
\renewenvironment{nazwa }[larg] [arg_opcj] {początek}{koniec}
\renewcommand{nazwa}[larg] [arg_opcj] {początek}{koniec}
```

Podamy teraz przykłady środowisk bez argumentów. Obszerniejsze informacje o definiowaniu środowisk z argumentami czytelnik znajdzie w literaturze. Środowiska tworzone przy pomocy poleceń `\newcommand` i `\renewcommand` opisywaliśmy już wcześniej zajmujemy się więc tylko poleceniem `\newenvironment` i `\renewenvironment`. Zdefiniujemy środowisko w którym tekst będzie węższy niż pozostały (z obu stron) i będzie pisany kapitalikami.

```
\newenvironment{nowe}{\begin{quote}\scshape}\end{quote}}
```

TEN TEKST JEST UMIESZCZONY W NOWYM, ZDEFINIOWANYM PRZEZ UŻYTKOWNIKA ŚRODOWISKU O NAZWIE `nowe`. ŚRODOWISKO TO ROZPOCZYNA SIĘ POLECENIEM `\begin{nowe}` A KOŃCZY `\end{nowe}`.

Wewnątrz środowiska definiowanego przez użytkownika można umieszczać inne, już zdefiniowane wcześniej (w tym przypadku było to środowisko `quote`).

## 20. Pakiet `theorem`

### 20.1. Zmiana nagłówka twierdzenia

Pakiet `theorem` autorstwa Franka Mittelbacha pozwala nam na globalną zmianę wyglądu tekstu twierdzenia lub definicji. Można zmienić wygląd nazwy i numeru twierdzenia, krój czcionki jaką ta nazwa jest napisana oraz krój użyty do napisania samej treści twierdzenia. Ma to duże znaczenie dla czytelności tekstu, szczególnie tekstu matematycznego.

Zobaczmy na przykładach jak to można uzyskać w L<sup>A</sup>T<sub>E</sub>X-u. Wszystkie deklaracje zmian dotyczące wyglądu twierdzeń i definicji należy umieszczać w preambule dokumentu.

Deklaracje zmieniające wygląd tytułu twierdzenia:

```
\theoremstyle{plain}  
\theoremstyle{break}  
\theoremstyle{marginbreak}  
\theoremstyle{changebreak}  
\theoremstyle{change}  
\theoremstyle{margin}
```

- \* Styl `plain` emuluje oryginalny L<sup>A</sup>T<sub>E</sub>X-owy styl twierdzenia i dodatkowo jeszcze dodaje przed i po treści twierdzenia niewielki odstęp.
- \* Styl `break` – etykieta i numer umieszczone są w osobnej linii a tekst twierdzenia rozpoczyna się od nowej.
- \* Styl `marginbreak` – numer twierdzenia jest umieszczony na marginesie, po etykiecie zaś następuje złamanie linii. Tekst umieszczany jest w nowej linii.
- \* Styl `changebreak` – tak samo jak `break` tylko najpierw jest drukowany numer a potem etykieta.
- \* Styl `change` – etykieta i numer twierdzenia są zamienione miejscami.
- \* Styl `margin` – numer jest umieszczony na marginesie ale bez zmiany linii.

Ich zastosowanie pokazują przykłady 1, 2, 3 i 4.

### Przykład 1. Styl `break`

#### **Wniosek 1**

*To jest zdanie napisane w środowisku `Cor` w stylu `break`.*

### Przykład 2. Styl `marginbreak`

#### **2 Lemacik**

*To jest nowy lemat.*

### Przykład 3. Styl `plain`

**Przykład 1** *To jest zdanie napisane w środowisku **Exa** w stylu **plain** (domyślnym).*

**Przykład 4.** Styl `marginbreak`

### **3 Lemacik (Ben King)**

*To jest zdanie napisane w środowisku **Lem** w stylu **marginbreak**, z dodatkowym argumentem z nazwą twierdzenia.*

**Przykład 5.** Styl `break`

**4 Definicjka (Bardzo ważna definicja)** *To jest zdanie napisane w środowisku **Def** w stylu **break**, z dodatkowym argumentem.*

```
\begin{Lem}[Ben King]
To jest zdanie napisane w środowisku \texttt{Lem}
  w stylu \texttt{marginbreak},
z dodatkowym argumentem z nazwą twierdzenia.
\end{Lem}
```

```
\begin{Def}[Bardzo ważna definicja]
To jest zdanie napisane w środowisku \texttt{Def}
  w stylu \texttt{break},
z dodatkowym argumentem.
\end{Def}
```

## **20.2. Zmiana czcionki w tekście twierdzeń**

Uważny czytelnik spostrzeże, że treści w środowiskach w przykładach 1,2, 4 i 5 są napisane tym samym krojem czcionki (pochyła antykwa) zaś w przykładzie 3 innym (italik). Do zmiany kroju w treści twierdzenia służy polecenie `\theorembodyfont`. Jako argument należy podać nazwę rodziny czcionek lub kroju, który chcemy użyć. Może to być np. `\rmfamily`, krój `\scshape` lub `\itshape`.

Możemy również zmienić krój czcionki w nagłówku twierdzenia poleceniem `\theoremheaderfont`. Ponieważ w jednym dokumencie nie stosuje się różnych rodzajów nagłówków więc może ono wystąpić tylko raz i tylko w preambule, przed pozostałymi definicjami dotyczącymi twierdzeń. Aby uzyskać w całym dokumencie nagłówki twierdzeń, definicji itp. np. pisane kapitalikami, należy w preambule umieścić komendę:

```
\theoremheaderfont{\scshape}
```

Jeżeli teraz treść twierdzeń ma być pisana pismem pochyłym a treść definicji pismem prostym i styl ma być np. `break` to w preambule muszą znaleźć się następujące definicje:

```
{\theoremstyle{break}
  {\theorembodyfont{\slshape}
    \newtheorem{twier}{Twierdzenie}}}
{\theoremstyle{break}
  {\theorembodyfont{\rm}
    \newtheorem{defi}{Definicja}}}
```

## 21. Odnośniki i cytowania

L<sup>A</sup>T<sub>E</sub>X posiada bardzo wygodny mechanizm służący do poruszania się po definicjach, twierdzeniach i wzorach a nawet po treści na konkretnych stronach.

Każde z wymienionych środowisk powinno być zaopatrzone w etykietę którą tworzy następujące polecenie `\label{nazwa}`. Słowo "nazwa" może składać się z kombinacji liter, cyfr lub symboli (z wyjątkiem symboli specjalnych), na przykład `sec:spis1` (nie może zawierać znaków diakrytycznych czyli także polskich liter). Do takiej etykiety odwołujemy się umieszczając w stosownym miejscu polecenie `\ref{nazwa}`. Można również odwołać się do konkretnej strony dokumentu na której dowolny tekst został uznany jako nazwa etykiety `verb`—poleceniem `\pageref{nazwa}`. I tak na przykład jeśli na stronie pierwszej dokumentu umieścimy etykietę `\label{przy:x}` to w dalszym ciągu dokumentu możemy się do tego miejsca odwołać przez `\pageref{przy:x}`.

**Uwaga:** Aby wszystkie odwołania zostały utworzone, należy uruchomić L<sup>A</sup>T<sub>E</sub>X-a dwukrotnie!

### 21.1. Przykłady

Ten rozdział został zaopatrzony w etykietę i można będzie się do niego odwołać w dalszej części. Teraz umieścimy kilka równości do których się będziemy odwoływać później.

$$x^2 + y^2 = r^2 \tag{21.30.}$$

I jeszcze inna równość

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \tag{21.31.}$$

I na tym skończymy pierwszy rozdział o odnośnikach i cytowaniach.

---

<sup>6</sup>tekścik

## 22. Rozdział drugi

W tym rozdziale będziemy odwoływać się do znaczników (etykiet) umieszczonych w rozdziale poprzednim. Wykorzystamy polecenie `\ref{etykieta}`.

W rozdziale 21.1. umieściliśmy dwa bardzo ważne wzory. Wzór (21.30.) przedstawia równanie okręgu o środku w punkcie  $(0, 0)$  i promieniu  $r$  a wzór (21.31.) przedstawia równanie elipsy o środku w punkcie  $(0, 0)$  i półosiach  $a$  i  $b$ .

## 23. Tworzenie spisu literatury

Każda praca naukowa powinna zawierać spis literatury. W zależności od tego gdzie jest ona publikowana (a więc składana w klasie `article`, `report` lub `book`) spis literatury umieszczany jest na osobnych stronach lub bezpośrednio po zakończeniu artykułu. I tak w klasie `article` spis literatury umieszczany jest bezpośrednio pod tekstem artykułu, w klasie `report` na następnej stronie a w klasie `book` na następnej stronie nieparzystej.

Umieszczamy go w środowisku o następującej składni:

```
\begin{thebibliography}{przykładowa etykieta}
\bibitem[etykieta]{klucz} treść
.....
\end{thebibliography}
```

Bez opcjonalnego argumentu `etykieta` L<sup>A</sup>T<sub>E</sub>X umieszcza kolejne numery w nawiasach kwadratowych jako etykiety poszczególnych pozycji w spisie literatury. Jeżeli chcemy, by kolejne pozycje spisu miały inne etykiety niż numery, wówczas każdą taką etykietę umieszczamy w miejscu argumentu `etykieta`. Aby cały spis został poprawnie sformatowany należy umieścić w miejscu `przykładowa etykieta` najdłuższą występującą w nim etykietę. Obowiązkowy argument `klucz` może składać się z dowolnej kombinacji liter, cyfr i znaków z wyjątkiem przecinka. Tekst argumentu `klucz` umieszczamy w instrukcji `\ref{klucz}` gdy chcemy w tekście dokumentu odwołać się do konkretnej pozycji spisu literatury (tu się odwołujemy do strony 36). W tym miejscu zostaje umieszczony numer strony na której znajdował się nasz znacznik.

## 24. Odwołania do spisu literatury

W tym rozdziale będziemy odwoływać się do poszczególnych pozycji spisu literatury. Służy do tego polecenie `\cite{klucz}`.

Na końcu dokumentu utworzymy dwa spisy literatury (ich ilość nie jest ograniczona, jednak w dokumencie umieszcza się tylko jeden spis literatury).

Poszczególne pozycje w spisie zaopatrzymy w klucze aby można się było do nich odwołać.

Teraz odwołamy się do pierwszej [1] pozycji w pierwszym spisie literatury (następująco: `\cite{aa}`), a w tym miejscu do pozycji drugiej [2] (`\cite{bb}`).

A teraz odwołujemy się do drugiego spisu [BC] i [CC] (pisząc `\cite{bc}` i `\cite{cc}`).

---

Sporządzimy teraz przykładowy spis literatury. Jak wyjaśniliśmy wyżej, poszczególne elementy bibliografii można numerować liczbami lub nadać im etykiety literowe lub literowo-liczbowe. Pierwszy sposób jaki pokażemy to zwykłe numerowanie (domyślne). Umieścimy również odpowiednie klucze, zgodnie z definicją pozycji bibliograficznej. Odwołania umieszciliśmy wcześniej.

## Literatura

- [1] S.A. Bleiler and A.J. Casson, Automorphisms of surfaces after Nielsen and Thurston, Cambridge Univ. Press, Cambridge, 1988.
- [2] J. Cantwell and L. Conlon, *Foliation Cones*, in Proceedings of the Kirbyfest, 35–86, Geometry and Topology Monographs, 1999.

## Literatura

- [BC] S.A. Bleiler and A.J. Casson, Automorphisms of surfaces after Nielsen and Thurston, Cambridge Univ. Press, Cambridge, 1988.
- [CC] J. Cantwell and L. Conlon, *Foliation Cones*, in Proceedings of the Kirbyfest, 35–86, Geometry and Topology Monographs, 1999.

W pierwszym spisie pierwsza pozycja bibliograficzna wyglądała następująco: `\bibitem{bb}` a w drugim `\bibitem[BC]{bc}`. Widać od razu, że różnią się one etykietą umieszczoną w nawiasie kwadratowym. Jeśli jej nie ma to domyślnie pozycje literatury są numerowane kolejnymi liczbami.

I na koniec tego tematu uwaga. Spis literatury w artykule umieszczamy tam, gdzie ma się pojawić (czyli na samym jego końcu) gdyż środowisko `thebibliography` pojawia się tam gdzie je sami umieścimy.

## 25. Spis treści

Spis treści umieszczamy w dokumencie poleceniem `\tableofcontents`. Spis pojawi się w tym miejscu, gdzie umieścimy polecenie. Zazwyczaj spis umieszcza się na początku lub na końcu dokumentu, u nas znajduje się na końcu.

Spis treści tworzony automatycznie przez  $\LaTeX$  zawiera numery i tytuły rozdziałów oraz numery stron na których się one zaczynają. Standardowo w klasie `book` i `report` w spisie znajdować się będą numery i tytuły do podrozdziałów (subsection) włącznie a w klasie `article` do poppodrozdziałów (subsubsection) włącznie.

Tworząc spis treści  $\LaTeX$  czyta informacje o rozdziałach podczas pierwszego przebiegu i zapisuje je w pliku o rozszerzeniu `.toc` a następnie, podczas drugiego przebiegu drukuje aktualny spis treści.

W tym dokumencie spis treści umieściliśmy na początku, bezpośrednio po tytule.

## 26. Rozdział o nazwie „Dodatek”

W niektórych dokumentach musimy umieścić rozdział, który jest słabo związany z poprzednimi ale jest w tym dokumencie niezbędny gdyż na przykład zawiera bardzo ważne informacje dodatkowe. W takim przypadku mamy do dyspozycji polecenie `\appendix`. Można też treść dodatku umieścić pomiędzy poleceniami `\begin{appendix}` i `\end{appendix}`. Liczniki części (section) w klasie `article` i rozdziałów (chapter) w klasach `book` i `report` zostają wyzerowane i zastąpione dużymi literami alfabetu. Przykład użycia tego polecenia czytelnik znajdzie na końcu dokumentu.

## 27. Przypisy

Przypisy umieszczamy w tekście komendą `\footnote{tekst przypisu}`.

W miejscu jej umieszczenia pojawia się numer przypisu a na dole strony jego treść opatrzona tym samym numerem. Treść przypisów na stronie jest od reszty tekstu oddzielona poziomą kreską. Pierwsza linia przypisu<sup>1</sup> jest odsunięta od marginesu o wielkość wcięcia akapitowego w dokumencie. Standardowe znaczniki przypisów to małe, podniesione w górę liczby. Domyślnie wszystkie przypisy w dokumencie w klasie `article` są numerowane w sposób ciągły a w klasach `book` i `report` rozpoczynają się od pierwszego wraz z rozpoczęciem nowego rozdziału. Za pomocą komendy `\footnote` możemy utworzyć przypis tylko wtedy, gdy  $\LaTeX$  jest w trybie akapitowym. Przypis nie pojawi się gdy umieścimy go w trybie matematycznym lub kiedy  $\LaTeX$  jest w trybie LR (w środowisku `\parbox`). Chociaż możemy komendy `\footnote` użyć w środowisku `minipage`, jednak przypis pojawi się u dołu ministrony a nie na dole bieżącej strony.

---

<sup>1</sup>jest odsunięta od marginesu o wielkość wcięcia akapitowego w dokumencie

Chociaż możemy komendy `\footnote` użyć w środowisku `minipage`<sup>a</sup>, jednak przypis pojawi się u dołu ministrony a nie na dole bieżącej strony. To jest przykład użycia przypisu w środowisku `minipage`.

---

<sup>a</sup>jednak przypis pojawi się u dołu ministrony

Komenda `\footnote` musi pojawić się bezpośrednio po wyrazie którego dotyczy, nie należy poprzedzać jej spacją, kropką lub przecinkiem.

### 27.1. Przypisy niestandardowe

Jeżeli użytkownik życzy sobie, by w klasie `article` przypisy w każdym rozdziale numerowały się od początku, wówczas może bezpośrednio przed lub po rozpoczęciu każdego rozdziału wyzerować licznik przypisów poleceniem `\setcounter{footnote}{0}`.

Przypisy domyślnie numerowane są liczbami arabskimi. Możemy w znany już sposób zmienić styl numerowania.

```
\renewcommand{\thefootnote}{\styl_numerowania{footnote}}
```

Oprócz znanych już stylów numerowania

```
\arabic, \roman, \Roman, \alph i \Alph
```

mamy też dziewięć symboli które wywołujemy poleceniem `\fnsymbol`. Są to symbole następujące (w kolejności występowania w zbiorze)

`*`, `†`, `‡`, `§`, `¶`, `||`, `**`, `††`, `‡‡`.

Jeżeli styl numerowania przypisów jest ustawiony na symboliczny wówczas po poleceniu `\footnote` należy umieścić w nawiasach kwadratowych numer symbolu, który ma być napisany na przykład zmienimy teraz sposób numerowania przypisów na symboliczny i umieścimy na dole przypis oznaczony znakiem paragrafu (§).

Teraz będzie przypis<sup>§</sup> oznaczony znakiem paragrafu. Aby numeracja przypisów znowu była tak jak ustawiona domyślnie, to po wszystkich zmianach musimy do niej powrócić w znany sposób czyli pisząc:

```
\renewcommand{\thefootnote}{\arabic{footnote}}
```

Ponieważ nie zerowaliśmy licznika przypisów więc następny powinien mieć numer kolejny. Sprawdźmy to<sup>2</sup> Odpowiedź jest łatwa, to jest drugi przypis,

---

<sup>§</sup>oznaczony znakiem paragrafu.

<sup>2</sup>Jaki jest numer tego przypisu?



gdyż pozostałe były bądź umieszczone w środowisku `minipage` bądź oznaczone były symbolem. Jeżeli chcemy, by nasz przypis miał kolejny numer (w tym przypadku 4) to musimy odpowiednio ustawić licznik przypisów poleceniem `\setcounter`.

W trybach pracy  $\text{\LaTeX}$ -a, w których nie działa komenda `\footnote` możemy przypis umieścić korzystając jednocześnie z dwóch komend:

`\footnotemark[numer]` i `\footnotetext[numer]{tekst}`. W tym przypadku wstawiamy ręcznie numer przypisu w środowisku LR a jego tekst umieszczamy bezpośrednio po zamknięciu tego środowiska czyli już w trybie akapitowym. Po każdorazowym użyciu tych dwu poleceń należy pamiętać, o odpowiednim zwiększeniu licznika przypisów, by nie pojawiły się przypisy o tym samym numerze.

No i jeszcze przykład użycia omówionych poleceń.

Umieścimy przypis w pudełku które nie jest ministroną. Odnośnik do przypisu umieścimy tutaj<sup>3</sup>. Bezpośrednio po zamknięciu pudełka umieścimy tekst przypisu. Ten przypis będzie kolejnym numerowanym przypisem w tym dokumencie.

## 28. Tworzenie tabel

Tabele możemy tworzyć wykorzystując wykorzystując następujące środowiska:

```
\begin{array}[pozycja]{opis kolumny} rzędy \end{array}
\begin{tabular}[pozycja]{opis kolumny} rzędy \end{tabular}
\begin{tabular*}{szerokość}[pozycja]{opis kolumny}rzędy\end{tabular*}
```

Jeśli umieścimy tabelę w środowisku `\begin{center}` i `\end{center}` będzie ona wyśrodkowana. Kolejne kolumny tabeli oddzielamy znakiem `&` a każdy wiersz kończymy znakiem `\\`.

Treść tabeli umieszczamy pomiędzy

```
\begin{tabular}{rcl}
treść tabeli
\end{tabular}
```

Parametry `r`, `l`, `c` oznaczają sposób wyrównywania tekstu w kolumnach i są obowiązkowe.

Prostą tabelę pokazuje następujący przykład

---

<sup>3</sup>A to jest treść tego przypisu.

**Przykład 1.**

Miejsce	Drużyna	Gole	Punkty
1	Legia	12	36
2	Górnik	10	30
3	Widzew	8	24
4	ŁKS	7	21

A oto jak utworzyliśmy tę tabelę:

```
\begin{tabular}{rlcc}
Miejsce & Drużyna & Gole & Punkty \\ \ [0.2ex]
1 & Legia & 12 & 36 \\ \
2 & Górnik & 10 & 30 \\ \
3 & Widzew & 8 & 24 \\ \
4 & ŁKS & 7 & 21
\end{tabular}
```

Wielkość w nawiasach kwadratowych w pierwszej linii to dodatkowy odstęp między wierszem pierwszym i drugim.

Kreski pionowe w tabeli uzyskuje się przez wstawienie znaku — pomiędzy parametry określające położenie tekstu.

**Przykład 2.**

Miejsce	Drużyna	Gole	Punkty
1	Legia	12	36
2	Górnik	10	30
3	Widzew	8	24
4	ŁKS	7	21

```
\begin{tabular}{|r|l|c|c|}
Miejsce & Drużyna & Gole & Punkty \\ \ [0.2ex]
1 & Legia & 12 & 36 \\ \
2 & Górnik & 10 & 30 \\ \
3 & Widzew & 8 & 24 \\ \
4 & ŁKS & 7 & 21
\end{tabular}
```

Kreski poziome w tabeli uzyskuje się przez wstawienie polecenia `\hline` pomiędzy rzędy.

**Przykład 3.**

Miejsce	Drużyna	Gole	Punkty
1	Legia	12	36
2	Górnik	10	30
3	Widzew	8	24
4	ŁKS	7	21

```

\begin{tabular}{|r|l|c|c|}
Miejsce & Drużyna & Gole & Punkty \\ \ [0.2ex]
\hline
1 & Legia & 12 & 36 \\
\hline
2 & Górnik & 10 & 30 \\
\hline
3 & Widzew & 8 & 24 \\
\hline
4 & ŁKS & 7 & 21
\end{tabular}

```

Linie podwójną otrzymujemy przez wstawienie dwóch poleceń `\hline`.

#### Przykład 4.

Miejsce	Drużyna	Gole	Punkty
1	Legia	12	36
2	Górnik	10	30
3	Widzew	8	24
4	ŁKS	7	21

## 29. Wstawianie tabel do tekstu

Tabelę można wstawić do tekstu umieszczając ją bezpośrednio w tym miejscu w którym powstała lub umieszczając ją w środowisku `table`. Umożliwia nam ono dodanie podpisu do tabeli jak również zapewnia automatyczne numerowanie tworzonych przez nas tabel. Umożliwia również odwoływanie się w tekście do konkretnej tabeli przy pomocy polecenia `\ref{etykieta}`. Etykieta powinna znaleźć się wewnątrz środowiska `caption` (podpis). Środowisko `table` ma także argument opcjonalny umieszczany bezpośrednio po otwarciu, który pozwala określić miejsce umieszczenia w tekście naszej tabeli. Domyślnie tabela jest umieszczana u góry strony (argument `t` - top) ale można ją też umieścić na dole strony (`b` - bottom) lub w miejscu jej wystąpienia (`h` - here). Jednak

Miejsce	Drużyna	Gole	Punkty
1	Legia	12	36
2	Górnik	10	30
3	Widzew	8	24
4	ŁKS	7	21

Tabela 1: To jest tabela z argumentem [t]

nie zawsze udaje nam się przekonać program by zrobił to co chcemy i wtedy musimy jakoś go ręcznie oszukać. (Tutaj umieszczamy tabelę.)

Tabelę 1. staraliśmy umieścić w tym miejscu na stronie, gdzie umieściliśmy jej tekst. Niestety, na bieżącej stronie pozostało za mało miejsca i została przeniesiona na następną. W takim przypadku lepiej jest nie umieszczać argumentu h tylko t.

Miejsce	Drużyna	Gole	Punkty
1	Legia	12	36
2	Górnik	10	30
3	Widzew	8	24
4	ŁKS	7	21

Tabela 2: To jest tabela z argumentem [h]

A to ta sama tabela z argumentem h. Teraz nazywa się ona Tabela 2. i jest umieszczona na stronie tam gdzie ją napisaliśmy. Jest jeszcze dodatkowo wyśrodkowana.

Aby to otrzymać należało treść tabeli poprzedzić

```
\begin{table}[h]
\begin{center}
```

a na końcu napisać

```
\caption{To jest tabela z argumentem [h] \label{tab:2}}
\end{center}
\end{table}
```

Jeżeli podpis `\caption{}` umieścimy przed `\begin{tabular}` to opis tabeli będzie ponad jej treścią.

Dodatkowy odstęp pomiędzy nagłówkiem tabeli a samą tabelą otrzymamy przy pomocy polecenia `\vspace`. W naszym przypadku jest to odstęp wielkości 0.2 cm.

Tabela 3: To jest tabela z podpisem na górze

Miejsce	Drużyna	Gole	Punkty
1	Legia	12	36
2	Górnik	10	30
3	Widzew	8	24
4	ŁKS	7	21

## 30. Środowisko `tabbing`

### 30.1. Ustawienia tabulatora do pisania w kolumnach

Do umieszczania danych w prostych tabelach możemy wykorzystać środowisko `tabbing`. Pozwala ono ustawiać tabulator w z góry zaplanowanych miejscach na stronie. Szerokość kolumn ustawiamy w pierwszej linii przychodząc do następnej kolumny przy pomocy znaku `\=`. W kolejnych liniach przechodzimy do następnej kolumny znakiem `\>`. Każdą linię kończymy znakiem `\\`.

Możemy również podać wielkość odstępu między wierszami tabeli, umieszczając po znaku złamania wiersza w nawiasach kwadratowych stosowną liczbę. Zazwyczaj podaje się ułamek lub wielokrotność wysokości litery `x` bieżącego fontu. W pierwszym przykładzie odstęp między pierwszym wierszem i następnymi będzie równy `0.8ex`, czyli mniej niż wysokość litery `x` a w następnym będzie równy `3ex` czyli trzykrotności wysokości litery `x`.

#### Przykład 1.

```
\begin{tabbing}
Rodzaj obuwia\quad\= Jakość\quad\= Kolor\quad\= Cena\\[0.8ex]
Czółenka \> średnia \> brązowy \> niska \\
Sandały \> dobra \> biały \> wysoka \\
Pantofle \> kiepska \> granat \> niska
\end{tabbing}
```

Rodzaj obuwia	Jakość	Kolor	Cena
Czółenka	średnia	brązowy	niska
Sandały	dobra	biały	wysoka
Pantofle	kiepska	granat	niska

Odstęp między pierwszym wierszem i następnymi będzie równy `3ex`.

### Przykład 2.

```
\begin{tabbing}
Rodzaj obuwia\quad\quad= Jakość\quad\quad= Kolor\quad\quad= Cena\ll[3ex]
Czółenka \> średnia \> brązowy \> niska \ll
Sandały \> dobra \> biały \> wysoka \ll
Pantofle \> kiepska \> granatowy \> niska
\end{tabbing}
```

Rodzaj obuwia	Jakość	Kolor	Cena
Czółenka	średnia	brązowy	niska
Sandały	dobra	biały	wysoka
Pantofle	kiepska	granatowy	niska

**Zadanie 19.** Utworzyć tabelę, w której odstęp między wierszami są równe 0.5 cm.

W przykładzie drugim wyraz określający kolor obuwia był długi i nie zmieścił się w zarezerwowanym dla niego miejscu. W środowisku `tabbing`  $\TeX$  rezerwuje w następnych liniach tyle miejsca ile zajmuje wyraz w pierwszej linii.

**Zadanie 20.** Utworzyć tabelę, w której wyrazy w sąsiadujących kolumnach nachodzą na siebie i drugą, która tego błędu nie ma (z tą samą treścią).

Wskazówka: przeanalizować przykład 1.

Jeżeli nie możemy z góry przewidzieć jakiej długości wyrazy będziemy wpisywać w kolejnych wierszach tabeli lepiej posłużyć się tzw. linią wzorca. Zaraz po deklaracji otwierającej środowisko należy umieścić linię wzorca w której podajemy szerokość kolumn. Możemy to zrobić w dwojaki sposób: przy pomocy komendy `\hspace` określić szerokość kolumny lub napisać najdłuższy wyraz który może wystąpić w naszej tabeli. Przykłady pokazują oba te przypadki. \*forma komendy `\hspace` na początku linii jest konieczna, by pierwsza kolumna nie została zredukowana do szerokości 0 cm.

### Przykład 3.

```
\begin{tabbing}
\hspace*{3cm}\= \hspace{2.5cm} \=\hspace{4cm} \= \kill
\> Jabłka \> \> średnie \ll
Gruszki \> luksusowe \> \ll
\> \> Śliwki \> niedojrzałe
\end{tabbing}
```

Jabłka	średnie
--------	---------

Gruszki	luksusowe	Śliwki	niedojrzałe
---------	-----------	--------	-------------

A to jeszcze inny przykład tabeli.

określenie na 4cm	Długi tekst objaśniający	Długi tekst objaśniający	Długi tekst objaśniający
Krótkie określenie	Długi tekst objaśniający	Długi tekst objaśniający	Długi tekst objaśniający
Krótkie określenie	Długi tekst objaśniający	Długi tekst objaśniający	Długi tekst objaśniający

**Zadanie 21.** Utworzyć samodzielnie taką samą tabelę.

**Przykład 4.**

```
\begin{tabbing}
\hspace*{3cm}\=przykładowa kolumna \=\hspace{4cm} \= \kill
\>Jabłka \> Gruszki \\
\+ Dobre \> Niedobre \\
Owoce \> Smaczne
\end{tabbing}
```

	Jabłka	Gruszki
Dobre	Niedobre	
	Owoce	Smaczne

**Zadanie 22.** Sprawdzić co się stanie z tabelką jeśli w linii wzorca nie umieścimy gwiazdki przy `\hspace`.

W przykładach 3 i 4 znalazły się jeszcze dwie niewyjaśnione komendy: `\kill` i znak `\+`. `\kill` umieszczona na końcu linii wzorca nie dopuszcza do jej wydrukowania. Umieszczenie na początku linii znaku `\>` powoduje umieszczenie wprowadzanego tekstu w drugiej kolumnie. Aby podobny efekt otrzymać w

następnym wierszu należy rozpocząć go od znaku `\>`. Jeśli w kilku kolejnych liniach chcemy zaczynać wprowadzać dane od np. drugiej kolumny (tzn. chcemy przesunąć lewy margines tabeli w prawo o jeden skok tabulatora), to w linii poprzedzającej należy umieścić znak `\+` (polecenie zaczyna działać od następnej linii). Można w linii umieścić tyle znaków `\+` ile jest zdefiniowanych skoków tabulatora. Aby w pojedynczej linii zlikwidować efekt komendy `\+` możemy użyć `\<`.

### Przykład 5.

```
\begin{tabbing}
\hspace*{3cm}\=kolumna wzorcowa =\hspace{3cm}\=\hspace{2cm}\=\kill
Owce \> Barany \+\+\
\+ Słonie \> Jaki  \
Małpy\> Osły\
\< \< Krowy \> Konie\> Lwy\
Woły\> Kozy
\end{tabbing}
```

Owce	Barany		Słonie	Jaki	
				Małpy	Osły
	Krowy		Konie	Lwy	
				Woły	Kozy

**Zadanie 23.** Po przeanalizowaniu poprzednich przykładów utworzyć tabelkę złożoną z 5 kolumn i 5 wierszy, w której w dwóch pierwszych i w ostatnim wierszu dane znajdują się w trzeciej i piątej kolumnie, natomiast w 3 wierszu znajdują się w kolumnie drugiej, czwartej i piątej, a w wierszu 4 we wszystkich kolumnach.

Poleceniem działającym w sposób przeciwny do `\+` jest `\-`. Zastosowanie go przesuwa margines tabeli o jeden skok tabulatora w lewo. Oczywiście nie można przesunąć marginesu poza zerowy skok tabulatora.

Ilość skoków tabulatora czyli ilość kolumn w tabeli możemy w każdej chwili zmienić (dodać lub usunąć kolumnę) przy pomocy `\=`. Kolumna zostanie dodana, jeśli ilość znaków `\>` będzie wystarczająca do przejścia do ostatniego skoku tabulatora, w przeciwnym przypadku usuwa następny skok. Ilustrują to następujące przykłady.

### Przykład 6.

```
\begin{tabbing}
Stara kolumna 1 \= Stara kolumna 2 \
```



```

Lewa kolumna \> Środkowa kolumna \= Ekstra kolumna\\
Nowa kolumna 1 \= Nowa kolumna 2 \> Stara 3\= Ekstra \\
Kolumna 1 \> Kolumna 2 \> Kolumna 3 \\
Stara kolumna 1 \> kolumna 2\\
Stara kolumna 1 \= Stara kolumna 2\> coś \=Coś nowego
\end{tabbing}

```

```

Stara kolumna 1 Stara kolumna 2
Lewa kolumna   Środkowa kolumna Ekstra kolumna
Nowa kolumna 1 Nowa kolumna 2   Stara 3 Ekstra
Kolumna 1      Kolumna 2        Kolumna 3
Stara kolumna 1 kolumna 2
Stara kolumna 1 Stara kolumna 2   coś Coś nowego

```

Niekiedy chcemy tylko na chwilę (w jednej linii) zlikwidować jeden ze stopów tabulatora. Możemy wtedy użyć polecenia `\pushtabs` które usuwa bieżący stop, pozostawiając go w pamięci. Wrócić do poprzednich ustawień można przy pomocy polecenia `\poptabs`. Obie te komendy mogą występować dowolną ilość razy ale musi ich być tyle samo w jednym środowisku `tabbing`.

Tekst w linii może być wyrównywany do lewego marginesu przy pomocy polecenia `\'`, a do prawego przy pomocy polecenia `\'`. Nie może być wówczas w tej linii żadnego znaku `\>` lub `\=`.

**Zadanie 24.** Sporządź następującą tabelę wykorzystując poznane możliwości środowiska `tabbing`.

```

Spodnie:   noszone przez: kobiety
                mężczyzn
                oraz dzieci
                raczej sportowe
Garnitury: eleganckie
(zobacz też spodniomy
                kostiumy)
Kobiety   noszą          sukienki

```

### 31. Inne przykłady zastosowanie środowiska `tabbing`

W pierwszej tabeli w linii wzorca ustawimy szerokości kolumn mniejsze niż potrzeba. Wówczas tekst z jednej kolumny zachodzi na tekst w kolumnie następnej.

```

\begin{tabbing}
\hspace*{3cm}\= \hspace*{2cm}\= \hspace*{5cm}\= \kill
Ala ma kota\> kota ma Ola\> Ala ma kota i Ola ma kota\>Iza ma psa\\
Ala ma \> kota ma \> Ala ma kota i kota\> Iza ma \\
Ala ma kota\> kota ma Ola\> Ala ma kota i Ola ma kota\>Iza ma psa
\end{tabbing}

```

```

Ala ma kota      kota ma OlaAla ma kota i Ola ma kota Iza ma psa
Ala ma           kota ma  Ala ma kota i kota      Iza ma
Ala ma kota      kota ma OlaAla ma kota i Ola ma kota Iza ma psa

```

```

\begin{tabbing}
\hspace*{3cm}\= \hspace*{3cm}\= \hspace*{5.5cm}\= \kill
Ala ma kota \> kota ma Ola \>Ala ma kota i Ola ma kota\>Iza ma psa\\
Ala ma \> kota ma \> Ala ma kota i kota\> Iza ma \\
Ala ma kota \> kota ma Ola \>Ala ma kota i Ola ma kota\>Iza ma psa
\end{tabbing}

```

```

Ala ma kota      kota ma Ola      Ala ma kota i Ola ma kota      Iza ma psa
Ala ma           kota ma           Ala ma kota i kota           Iza ma
Ala ma kota      kota ma Ola      Ala ma kota i Ola ma kota      Iza ma psa

```

W drugim przykładzie odstępy są już dobre. Zauważmy, że szerokość ostatniej kolumny jest równa różnicy między szerokością tekstu a sumą szerokości kolumn pozostałych.

Niekiedy możemy chcieć zlikwidować jakiś skok tabulatora tylko na chwilę (np. na trzy wiersze) a potem wrócić do poprzednich ustawień. W tym celu możemy wykorzystać polecenia: `\pushtabs` które usuwa skoki tabulatora z bieżącej linii i `\poptabs` które je reaktywuje. Zastosowanie tych poleceń pokazuje następujący przykład.

```

\begin{tabbing}
\hspace*{3cm}\= \hspace*{3cm}\= \hspace*{5.5cm}\= \kill
Ala ma kota\>kota ma Ola\>Ala ma kota i Ola ma kota\>Iza ma psa\\
Ala ma \> kota ma \> Ala ma kota i kota\> Iza ma \\
\pushtabs
(zobacz również: \= Iza \\
 \> Ela\\
 \> Ula )\\

```

```
\poptabs
Ala ma kota\>kota ma Ola\>Ala ma kota i Ola ma kota\>Iza ma psa
\end{tabbing}
```

```
Ala ma kota      kota ma Ola      Ala ma kota i Ola ma kota      Iza ma psa
Ala ma          kota ma          Ala ma kota i kota          Iza ma
(zobacz również: Iza
                  Ela
                  Ula )
Ala ma kota      kota ma Ola      Ala ma kota i Ola ma kota      Iza ma psa
```

## A Dodatek pierwszy

W niektórych dokumentach musimy umieścić rozdział, który jest słabo związany z poprzednimi ale jest w tym dokumencie niezbędny gdyż na przykład zawiera bardzo ważne informacje dodatkowe. W takim przypadku mamy do dyspozycji polecenie `\appendix`. Można też treść dodatku umieścić pomiędzy poleceniami `\begin{appendix}` i `\end{appendix}`. Liczniki części (section) w klasie `article` i rozdziałów (chapter) w klasach `book` i `report` zostają wyzerowane i zastąpione dużymi literami alfabetu. Nagłówek tego rozdziału otrzymaliśmy następująco:

```
\appendix
\section{Dodatek}
```

## B Dodatek drugi

To jest drugi dodatek zanumerowany drugą w kolejności literą alfabetu.

### Spis treści

1. Struktura dokumentu w $\LaTeX$ -u	1
2. Tryby pracy $\LaTeX$ -a	2
3. Odstępy poziome i pionowe	3
4. Style strony	4
5. Numerowanie stron	5

<b>6. Format strony</b>	<b>5</b>
6.1. Tekst w dwu kolumnach . . . . .	6
6.2. Łamanie linii tekstu . . . . .	7
6.3. Łamanie stron . . . . .	7
6.4. Formatowanie paragrafu . . . . .	8
6.5. Dzielenie wyrazów . . . . .	8
<b>7. Wprowadzanie wyrażeń i wzorów matematycznych</b>	<b>9</b>
<b>8. Formatowanie strony dokumentu</b>	<b>14</b>
8.1. Wybieramy rozmiar czcionki . . . . .	14
<b>9. Pudełka</b>	<b>14</b>
<b>10. Pakiet amsmath, wzory wielolinijkowe</b>	<b>16</b>
10.1. Środowisko gather . . . . .	16
10.2. Środowisko multiline . . . . .	17
10.3. Środowisko align . . . . .	17
10.4. Środowisko alignat . . . . .	18
10.5. Środowisko cases . . . . .	18
<b>11. Pakiet amscd</b>	<b>19</b>
<b>12. Tworzenie list</b>	<b>20</b>
12.1. Środowisko <b>itemize</b> i <b>enumerate</b> . . . . .	21
12.2. Środowisko <b>description</b> . . . . .	23
12.3. Listy zagnieżdżone . . . . .	24
12.4. Zmiana etykiet . . . . .	26
<b>13. Zmiana sposobu numerowania. Domyślny sposób numerowa-</b>	
<b>nia</b>	<b>28</b>
<b>14. Następny rozdział</b>	<b>28</b>
14.1. To jest podrozdział . . . . .	28
<b>15. Inna numeracja wzorów</b>	<b>29</b>
<b>16. I jeszcze jeden rozdział ze wzorami</b>	<b>29</b>
<b>17. To jeszcze jeden przykładowy rozdział</b>	<b>30</b>
17.1. To jest podrozdział . . . . .	30
<b>18. Struktury definiowane przez użytkownika</b>	<b>30</b>
<b>19. Definiowanie własnych komend i środowisk</b>	<b>32</b>

<b>20. Pakiet theorem</b>	<b>33</b>
20.1. Zmiana nagłówka twierdzenia . . . . .	33
20.2. Zmiana czcionki w tekście twierdzeń . . . . .	35
<b>21. Odnośniki i cytowania</b>	<b>36</b>
21.1. Przykłady . . . . .	36
<b>22. Rozdział drugi</b>	<b>37</b>
<b>23. Tworzenie spisu literatury</b>	<b>37</b>
<b>24. Odwołania do spisu literatury</b>	<b>37</b>
<b>25. Spis treści</b>	<b>38</b>
<b>26. Rozdział o nazwie „Dodatek”</b>	<b>39</b>
<b>27. Przypisy</b>	<b>39</b>
27.1. Przypisy niestandardowe . . . . .	40
<b>28. Tworzenie tabel</b>	<b>41</b>
<b>29. Wstawianie tabel do tekstu</b>	<b>43</b>
<b>30. Środowisko tabbing</b>	<b>45</b>
30.1. Ustawienia tabulatora do pisania w kolumnach . . . . .	45
<b>31. Inne przykłady zastosowanie środowiska     tabbing</b>	<b>49</b>
<b>A Dodatek pierwszy</b>	<b>51</b>
<b>B Dodatek drugi</b>	<b>51</b>