

Rozpoczynamy pracę z L^AT_EX-em

materiały pomocnicze do zajęć

Wprowadzenie do L^AT_EX-a

Zofia Walczak

Wydział Matematyki UŁ

październik 2006

1. Struktura dokumentu w L^AT_EX-u

Aby dokument tekstowy został przeczytany i przetworzony przez L^AT_EX-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_EX-a.

Klasa dokumentu to jest pakiet zawierający definicje i deklaracje m. inn. szerokości i wysokości strony, sposobu numerowania rozdziałów itp. Standardowo w L^AT_EX-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^AT_EX-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 T_EX-em.

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

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

Znakiem backslash (\) rozpoczyna się każda komenda T_EX-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 L^AT_EX-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 L^AT_EX-a

Podczas pracy L^AT_EX znajduje się w jednym z trzech trybów:

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

L^AT_EX 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 L^AT_EX natrafi na tekst matematyczny (zawarty pomiędzy znakami dolara) wtedy automatycznie przełącza się na tryb matematyczny. W tryb LR L^AT_EX 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 L^AT_EX-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 dwukropek, 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 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 \LaTeX 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
rozsypano się po kątach
straszenie 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 piesze 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 stron.

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 \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 łamana 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 \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 `\vspace{odległość}` (lub `\vspace*{odległość}`). Parametr `odległość` musi być liczbą (dodatnią lub ujemną) z mianem (np. 1cm, 3ex, 1pt). Ujemny parametr wstawia ujemny odstęp (zmniejsza odstęp istniejący). Komenda z gwiazdką wstawia dodatkowy pionowy odstęp od góry strony.

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

\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" \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 kresczką (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 \LaTeX 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

\LaTeX 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.

$$\frac{\sqrt{\pi R^2 + xyz}}{\sqrt[5]{\pi R^2 + xyz}} \quad (2)$$

A teraz tylko sam znak pierwiastka $\sqrt{}$.

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
\underbrace{a \cdot a \cdot a \cdot \dots \cdot a}_n $
```

Sposób pisania wszelkiego rodzaju znaków matematycznych można znaleźć w tabelach umieszczanych w każdym podręczniku do L^AT_EX-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 ograniczników można znaleźć w tabelach dowolnego poręcznika.

L^AT_EX 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.

$$\left(\left(\left(\left(\left\{ \left\{ \left\{ \left\{ \left\{ \right. \right) \right) \right) \right) \right) \right) \right) \right) \right)$$

Możemy również skorzystać z tego, że L^AT_EX 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 następnej `\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.

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

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

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 \dots , 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 - wyśrodkowany, 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 L^AT_EX-u środowisko matematyczne `eqnarray` które powstało na bazie środowiska `array`. Otrzymujemy układ równań wyrównany np. do znaku równości:

$$\begin{array}{rcl} \sin x & = & x - \frac{x^3}{3!} + \frac{x^5}{5!} - \\ & & - \frac{x^7}{7!} + \dots \end{array} \quad (3)$$
$$(4)$$

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 dokumentu L^AT_EX-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).

Przykład 1.

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

Ala ma kota. Ala ma kota.

Ala ma kota.

L^AT_EX posiada gotowe deklaracje różnych rozmiarów czcionek od bardzo małej (`\tiny`) do bardzo dużej (`\Huge`).

Przykład 2.

Stopień pisma				
tekstu głównego	10pt	11pt	12pt	
<code>\tiny</code>	5pt	6pt	6pt	najmniejsza
<code>\scriptsize</code>	7pt	8pt	8pt	bardzo mała
<code>\footnotesize</code>	8pt	9pt	10pt	nieco większa
<code>\small</code>	9pt	10pt	11pt	mała
<code>\normalsize</code>	10pt	11pt	12pt	rozmiar normalny
<code>\large</code>	12pt	12pt	14pt	duża
<code>\Large</code>	14pt	14pt	17pt	większa
<code>\LARGE</code>	17pt	17pt	20pt	jeszcze większa
<code>\huge</code>	20pt	20pt	25pt	jeszcze nieco większa
<code>\Huge</code>	25pt	25pt	25pt	największa

Bardziej wprawni użytkownicy L^AT_EX-a mogą zdefiniować sobie nową wielkość fontu korzystając z definicji postaci `\newfont{\mojfo}{plr10 at 20pt}`. Tekst pisany nowym fontem ma postać następującą:

Przykład 3.

Tekst 20-to punktową czcionką:

Ala ma kota i psa

Tekst czcionką 30-to punktową:

Ala ma kota i psa

Tekst czcionką 40-to punktową:

Ala ma kota i psa

A oto jak otrzymaliśmy powyższe przykłady

```
\newfont{\mojfo}{cmr10 at 20pt}  
\newfont{\mojfon}{cmr10 at 30pt}  
\newfont{\mojfont}{cmr10 at 40pt}
```

```
{\mojfo Ala ma kota i psa}
```

```
{\mojfon Ala ma kota i psa}
```

```
{\mojfont Ala ma kota i psa}
```

8.2. Odstępy między wierszami

Domyślnie \LaTeX składa tekst wstawiając między kolejnymi wierszami paragrafu odstępy równe wysokości wiersza (pojedyncze). W zależności od wielkości zastosowanego fontu (czcionki) wynoszą one 10, 12 lub więcej punktów.

Za odstęp między wierszami odpowiada polecenie `\baselineskip`. Jeżeli chcemy by odstępy były większe niż pojedynczy, należy zmienić jego domyślną wartość za pomocą znanego polecenia `\setlength` w następujący sposób:

```
\setlength{\baselineskip}{24pt}
```

Wartość ta jest zależna od wielkości bieżącego fontu, zatem należy pamiętać, że wraz ze zmianą wielkości fontu powinien zmienić się odstęp między wierszami. Zależność tą można zobaczyć w poniższym przykładzie. Pierwszy akapit tekstu jest złożony czcionką tej samej wielkości co cały dokument ale z podwójnym odstępem (24pt) a drugi czcionką `\small`.

Wartość ta jest zależna od wielkości bieżącego fontu, zatem należy pamiętać, że wraz ze zmianą wielkości fontu zmienia się odstęp między wierszami. Zależność tą można zobaczyć w poniższym przykładzie. Pierwszy akapit tekstu jest złożony czcionką tej samej wielkości co cały dokument ale z podwójnym odstępem (24pt) a drugi czcionką `\small`.

Wartość ta jest zależna od wielkości bieżącego fontu, zatem należy pamiętać, że wraz ze zmianą wielkości fontu zmienia się odstęp między wierszami. Zależność tą można zobaczyć w poniższym przykładzie. Pierwszy akapit tekstu jest złożony czcionką tej samej wielkości co cały dokument ale z podwójnym odstępem (24pt) a drugi czcionką `\small`.

Widzimy, że mimo zmiany wielkości czcionki odstępy między wierszami pozostały takie same. Wartość `\baselineskip` przyjmuje wielkość domyślną po każdej zmianie wielkości

czcionki (tzn. wszystkie ustawienia odstępów przy pomocy polecenia `\setlength` zerują się).

Aby powrócić do ustawień pierwotnych można dwukrotnie zmienić wielkość czcionki np. następująco: `\small\normalsize`. Należy jednak pamiętać, by umieścić je w oddzielnym paragrafie, gdyż wielkość odstępu jest ustalana dopiero na końcu, po przeczytaniu przez program całego akapitu. W przeciwnym wypadku ważne dla całego akapitu byłyby ustawienia umieszczone na jego końcu.

Aby odstępy między wierszami były dobierane prawidłowo wraz ze zmianą wielkości fontu należy zmienić wartość czynnika `\baselinestretch`. Rzeczywista wielkość odstępu między wierszami jest wyliczana przez program według wzoru

`\baselinestretch x \baselineskip`

Domyślnie wartość `\baselinestretch` jest równa 1. Użytkownik może zmienić wielkość odstępu między wierszami zmieniając jego wartość na dowolną liczbę dziesiętną poleceniem

`\renewcommand{\baselinestretch}{liczba}`

Jeżeli w miejsce `liczba` wstawimy na przykład liczbę 1.5 to odstęp między wierszami będzie zwiększony o 50%.

Zobaczmy jak w tym przypadku będą wyglądały poprzednie akapity tym razem z odstępem powiększonym o 50%.

Wartość ta jest zależna od wielkości bieżącego fontu, zatem należy pamiętać, że wraz ze zmianą wielkości fontu zmienia się odstęp między wierszami. Zależność tą można zobaczyć w poniższym przykładzie. Pierwszy akapit tekstu jest złożony czcionką tej samej wielkości co cały dokument ale z podwójnym odstępem (24pt) a drugi czcionką `\small`.

Wartość ta jest zależna od wielkości bieżącego fontu, zatem należy pamiętać, że wraz ze zmianą wielkości fontu zmienia się odstęp między wierszami. Zależność tą można zobaczyć w poniższym przykładzie. Pierwszy akapit tekstu jest złożony czcionką tej samej wielkości co cały dokument ale z podwójnym odstępem (24pt) a drugi czcionką `\small`.

W przypadku polecenia `\baselinestretch` aby powrócić do poprzednich ustawień należy ustawić jego wartość na domyślną czyli 1 oraz wstawić polecenie zmiany wielkości czcionki np. `\small\normalsize`.

8.3. Wyróżnienia w tekście

Tekst możemy także wyróżnić poprzez pochylenie, pogrubienie lub podkreślenie. \LaTeX proponuje nam następujące deklaracje wyróżniające tekst przez pochylenie `\em` i równoważną jej `\emph`. Sposób ich użycia ilustruje następujący

Przykład 4.

To będzie przykład użycia pierwszej deklaracji *pochylającej litery tekstu*

To będzie przykład użycia drugiej deklaracji *pochylającej litery tekstu*

A oto jak otrzymaliśmy powyższy tekst:

To będzie przykład użycia pierwszej deklaracji `{\em pochyłającej litery tekstu}\`

To będzie przykład użycia drugiej deklaracji `\emph{pochylającej litery tekstu}`

Dla podkreślenia jakiegoś wyrazu linią możemy użyć komendy `\underline` w następujący sposób:

Przykład 5.

To będzie przykład tekstu `\underline{podkreślonego}`

To będzie przykład tekstu podkreślonego

Dokument który tworzymy może wymagać użycia czcionek różnych kształtów, wielkości i grubości. Typowa instalacja T_EX-owa zawiera domyślnie trzy rodziny fontów: Computer Modern Romam (`\rmfamily`), czcionki imitujące maszynę do pisania (`\ttfamily`) (Computer Modern Typewriter) oraz czcionki bezszeryfowe (`\sffamily`) (Computer Modern Sansserif).

Czcionki należące do każdej z tych rodzin mogą posiadać kształt prosty (`\upshape`), pochyły (`\itshape`), pochylony (`\slshape`) oraz kapitaliki (`\scshape`). Mogą być również normalnej grubości (`\mdseries`) lub pogrubione (`\bfseries`).

Dla każdej z omówionych wyżej deklaracji kroju i odmian T_EX posiada komendy które ułatwiają wprowadzanie tekstu. I tak:

1. Krój pisma (rodzina)

- Normalny prosty tekst krojem szeryfowym otrzymamy dzięki komendzie `\textrm{tekst}` (lub `\rmfamily`) (jest to szczególnie przydatne gdy chcemy dodać komentarz w środowisku matematycznym).
- Tekst pisany czcionką bezszeryfową uzyskamy dzięki `\textsf{...}` lub `\sffamily`.
- Tekst pisany literami podobnymi do maszyny do pisania (grotesk) uzyskamy dzięki `\texttt{...}` lub `\ttfamily`.

2. Grubość i szerokość pisma

- Tekst pisany literami pogrubionymi uzyskamy dzięki `\textbf{...}` lub `\bfseries`.
- Tekst pisany pismem jasnym uzyskamy pisząc `\textmd{...}` lub `\bfseries`.

3. Odmiana pisma

- Tekst pisany literami prostymi (przełącza np. z pochylego) uzyskamy pisząc `\textup{...}` lub `\upshape`.
- Tekst pisany kapitalikami otrzymamy pisząc `\textsc{...}` lub `\scshape`.
- Tekst pisany kursywą otrzymamy pisząc `\textit{...}` lub `\itshape`.
- Tekst pisany literami pochyłymi uzyskamy dzięki `\textsl{...}` lub `\slshape`.

Przykład 6.

Przykład użycia wyżej wymienionych komend.

1. Przykład tekstu w środowisku matematycznym

$$x^2 + y^2 = 25 \quad \text{równanie okręgu}$$

2. *Tekst pisany literami pochyłymi* a teraz prostymi
A teraz zastosowanie `\upshape` i `\itshape`
To tekst pisany kursywą a to pismem prostym.
3. TEKST PISANY KAPITALIKAMI i zastosowanie `\scshape`
TEKST PISANY KAPITALIKAMI.
4. TEKST PISANY LITERAMI PODOBNYMI DO MASZYN DO PISANIA
5. *Tekst pisany italikami*
6. **Tekst pisany literami pogrubionymi**
7. TEKST PISANY CZCIONKĄ BEZSZERYFOWĄ
8. *Tekst pisany czcionką pochyłą*

A oto jak otrzymaliśmy te przykłady:

Przykład tekstu w środowisku matematycznym

```
$$x^2 + y^2 = 25 \quad \text{równanie okręgu}$$
\textit{Tekst pisany literami pochyłymi} \textup{a teraz prostymi}
\\ A teraz zastosowanie \verb|\upshape| i \verb|\itshape|
\\ \itshape To tekst pisany kursywą \upshape a to pismem prostym.
\textsc{Tekst pisany kapitalikami} i zastosowanie \verb|\scshape|
\scshape Tekst pisany kapitalikami.
\texttt{Tekst pisany literami podobnymi do maszyny do pisania}
\textit{Tekst pisany italikami}
\textbf{Tekst pisany literami pogrubionymi}
\textsf{Tekst pisany czcionką bezszeryfową}
\textsl{Tekst pisany czcionką pochyłą}
```

9. Pozycjonowanie tekstu

9.1. Wyśrodkowanie

Tekst na stronie może być położony w różny sposób. Zazwyczaj jest wyrównany do obu marginesów ale w niektórych przypadkach możemy chcieć розміścić go inaczej. Aby tekst umieścić pośrodku strony mamy do dyspozycji środowisko `center`. Składnia jego jest następująca

```

\begin{center}
linia pierwsza \\
linia druga \\
....\\
linia n
\end{center}

```

Jeżeli wyśrodkujemy cały paragraf tekstu i nie zależy nam na podzieleniu go na poszczególne linie to możemy zaniechać łamania linii. Cały tekst będzie wtedy podzielony na odpowiednią ilość linii i każda z nich wycentrowana. Zastosowanie tego środowiska pokazuje:

Przykład 1.

Tekst wyśrodkowany bez znaków łamania linii.

Jeżeli centrujemy cały paragraf tekstu i nie zależy nam na podzieleniu go na poszczególne linie to możemy zaniechać łamania linii. Cały tekst będzie wtedy podzielony na odpowiednią ilość linii. Jeżeli centrujemy cały paragraf tekstu i nie zależy nam na podzieleniu go na poszczególne linie to możemy zaniechać łamania linii. Cały tekst będzie wtedy podzielony na odpowiednią ilość linii. Cały tekst będzie wtedy podzielony na odpowiednią ilość linii.

Tekst wyśrodkowany ze znakami łamania linii.

Jeżeli centrujemy cały paragraf tekstu
i zależy nam na podzieleniu go na poszczególne linie
to nie możemy zaniechać łamania linii.
Cały tekst będzie wtedy podzielony na odpowiednią ilość linii.
Jeżeli centrujemy cały paragraf tekstu i zależy nam
na podzieleniu go na poszczególne linie to nie możemy
zaniechać łamania linii. Cały tekst będzie wtedy podzielony
na odpowiednią ilość linii.
Cały tekst będzie wtedy podzielony na odpowiednią ilość linii.

Jeżeli chcemy wyśrodkować tylko jedną linię to możemy użyć komendy `\centerline{tekst}`.

9.2. Wyrównywanie

Wyrównywanie tekstu paragrafu do jednego z marginesów możliwe jest dzięki środowisku `flushleft` i `flushright`. Składnia jest następująca

```

\begin{flushleft}
linia pierwsza \\
linia druga \\
....\\
linia n
\end{flushleft}

```

wyrównuje do lewego marginesu i

```

\begin{flushright}
linia pierwsza \\
linia druga \\
....\\
linia n
\end{flushright}

```

wyrównuje do prawego marginesu. Taki sam efekt uzyskamy zastępując środowisko `flushleft` i `flushright` komendami `\raggedleft` i `\raggedright`. W przypadku tych ostatnich każda linia musi być zakończona znakiem łamania (`\\`).

Uwaga: W żadnym z tych środowisk \LaTeX nie przenosi wyrazów.

Przykład 2.

Każda linia tego tekst będzie
wyrównana do prawego
marginesu.

a każda linia tego tekst będzie
wyrównana do lewego
marginesu.

Każda linia tego tekst będzie
wyrównana do lewego
marginesu.

a każda linia tego tekst będzie
wyrównana do prawego
marginesu.

Zadanie 13. Wypróbuj działanie środowiska `center`.

Zadanie 14. Wypróbuj działanie środowisk `flushleft` i `flushright` oraz `\raggedleft` i `\raggedright`. Porównaj efekty.

Jeżeli jakaś część tekstu ma być wyróżniona w ten sposób, że odległość od lewego i prawego marginesu będzie taka sama to możemy użyć środowiska `quote` lub `quotation` o następującej składni:

```

\begin{quote} tekst \end{quote}
\begin{quotation} tekst \end{quotation}

```

Różnica między tymi dwiema formami jest taka, że w środowisku `quote` akapity nie zaczynają się od wcięcia i są od siebie oddzielone dodatkowym odstępem, zaś w środowisku `quotation` tekst każdego akapitu rozpoczyna się wcięciem. W obu przypadkach dodawany jest dodatkowy odstęp przed i za wyróżnionym tekstem.

Zadanie 15.

Spróbuj uzyskać dokładnie tak sformatowany tekst.

Dla lepszego uwidocznienia różnic pomiędzy środowiskami ten tekst będzie powtórzony dwukrotnie.

To będzie przykład użycia środowiska `quote`. Jeżeli jakaś część tekstu ma być wyróżniona w ten sposób, że odległość od lewego i prawego marginesu będzie taka sama to możemy użyć środowiska `quote` lub `quotation`.

Różnica między tymi dwiema formami jest taka, że w środowisku `quote` akapity nie zaczynają się od wcięcia i są od siebie oddzielone dodatkowym odstępem, zaś w środowisku `quotation` tekst każdego akapitu rozpoczyna się wcięciem.

W obu przypadkach dodawany jest dodatkowy odstęp przed i za wyróżnionym tekstem.

To będzie przykład użycia środowiska `quotation`. Jeżeli jakaś część tekstu ma być wyróżniona w ten sposób, że odległość od lewego i prawego marginesu będzie taka sama to możemy użyć środowiska `quote` lub `quotation`.

Różnica między tymi dwiema formami jest taka, że w środowisku `quote` akapity nie zaczynają się od wcięcia i są od siebie oddzielone dodatkowym odstępem, zaś w środowisku `quotation` tekst każdego akapitu rozpoczyna się wcięciem.

W obu przypadkach dodawany jest dodatkowy odstęp przed i za wyróżnionym tekstem.

Do pisania poezji możemy użyć środowiska `verse`. Korzysta się z niego w następujący sposób

```
\begin{verse}
pierwsza linia pierwszej zwrotki\\
druga linia pierwszej zwrotki\\
...
ostatnia linia pierwszej zwrotki

pierwsza linia drugiej zwrotki\\
druga linia drugiej zwrotki\\
...
ostatnia linia drugiej zwrotki
\end{verse}
```

A oto wiersz pt. Odrodzenie A. Puszkina w tłumaczeniu J. Tuwima (tekst umieszczony w środowisku `verse`):

```
Artysta-wandal pędzlem sennym
Powleka arcytwór czernidłem,
Potem, bezmyślnie, skarb bezcenny
Pokrywa własnym malowidłem.

Z czasem się obce farby kruszą,
Odpada obraz ów bezprawny
I znów stoimy przed geniuszem,
Przed jego arcydziełem dawnym.

Tak z duszy mojej umęczonej
Znikają błędy i wahania,
A wstają zjawy upragnione
Niewinnych, czystych dni zarania.
```

10. Części dokumentu

Artykuł przygotowywany do druku w czasopiśmie naukowym składa się z tytułu, nazwiska autora, jego adresu, daty utworzenia (opcjonalnie), krótkiego streszczenia, treści dokumentu oraz spisu literatury. Niekiedy na dole pierwszej strony pojawiają się informacje o grantach lub podziękowania osobom lub instytucjom. Treść artykułu bywa podzielona na rozdziały, podrozdziały itp.

W L^AT_EX-u tytuł, nazwisko autora, jego adres, datę utworzenia (opcjonalnie) i ewentualne podziękowania umieszcza się w preambule dokumentu wykorzystując następujące polecenia:

```
\documentclass{article}
\title{Tytuł dokumentu}
\author{Nazwisko autora \thanks{Podziękowania}
\\ Jego adres}
\date{Data utworzenia}

\begin{document}
\maketitle
```

Aby wszystkie te polecenia zostały wykonane należy bezpośrednio po `\begin{document}` umieścić komendę `\maketitle`. Bez niej podane informacje nie zostaną wydrukowane. Polecenie `\thanks` umieszcza się gdziekolwiek w tekście poleceń `\author`, `\title` lub `\date`. W miejscu jego pojawienia się pojawia się odsyłacz do dołu strony, gdzie pojawia się wprowadzony tekst podziękowań.

Jeśli chcemy, by tytuł i nazwisko autora znalazły się na osobnej stronie to należy zadeklarować to jako parametr w klasie dokumentu

```
\documentclass[titlepage]{article}
```

Streszczenie dokumentu otrzymamy, gdy tekst umieścimy w środowisku

```
\begin{abstract}
treść streszczenia
\end{abstract}
```

W klasie `article` streszczenie pojawia się bezpośrednio ponagłówek z tytułem i nazwiskiem autora na pierwszej stronie dokumentu, w klasie `report` streszczenie umieszczane jest na osobnej stronie. Jeżeli w klasie `article` wybrana jest opcja `titlepage` wtedy streszczenie jest drukowane na oddzielnej stronie. W klasie `book` streszczenie nie występuje.

Każdy dokument powinien być podzielony na logiczne części. W klasie `article` mamy dostępne następujące instrukcje hierarchizujące:

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

W klasie `report` i `book` dodatkowo można użyć instrukcji `\part{...}` i `\chapter{...}`.

We wszystkich przypadkach L^AT_EX dobiera odpowiednie odstępy między rozdziałami (częściami), krój i wielkość pisma w śródtytułach oraz zadba o numerację.

11. Pudełka

L^AT_EX 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 prostokącik na wyrazie.

`\framebox[2mm]{centered}`

centered

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

`\makebox[0pt][l]{/}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, wyrównane 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 uniknąć.

otrzymaliśmy pisząc

```
\parbox{3.5cm}{\sloppy To jest pudełko o szerokości 3.5 cm,
wyrównane 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. Położenie środka wą-
ne jest tak, że tekst w ostat-
niej jego linii leży na wyso-
kości linii bazowej. ko które jest Zawartość pudełka umiesz-
wycelowane czonogo po prawej stronie
względem linii jest wyrównana do linii ba-
bazowej zowej w ten sposób, że je-
go pierwsza linia z nią się
pokrywa. W obu przypad-
kach używania środowiska
`minipage` nie można zapo-
minąć o zaznaczeniu pozycji
w jakiej ma być umieszczo-
ny tekst.

12. Pakiet `amsmath`, wzory wielolinijkowe

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

12.1. Środowisko `gather`

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

Przykład 1.

$$x^2 + 2xy - y^2 = 3 - xy \tag{1}$$

$$3x - 2y^3 + 13 = 2 - x^2 \tag{2}$$

$$x + y - 1 = 3xy \tag{3}$$

```
\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*`

12.2. Środowisko `multline`

W środowisku `multline` kolejne linie wzoru rozmieszczane 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.

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

```
\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}
```

12.3. 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.

$$x + y + z = 23 \tag{5}$$

$$2x - 3y + z = x^2 - 4y^2 - 1 \tag{6}$$

$$y^4 - 4z^2 = 21 \tag{7}$$

```
\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`.

12.4. Środowisko `alignat`

Środowisko `alignat` również pozwala na rozmieszczenie wzorów w wyrównanych kolumnach, jednak, inaczej niż w przypadku `align`, samo nie decyduje o wielkości odstepu 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.

$$f(x) = x + 2y \quad g(x) = x - y \quad (8)$$

$$h(x) = x^2 + 1 \quad k(x) = y^2 \quad (9)$$

```
\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 `&`.

12.5. Środowisko `cases`

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

$$f(x) = \begin{cases} x & \text{gdy } x > 5 \\ 0 & \text{gdy } 0 \geq x \geq 5 \\ -x & \text{dla pozostałych} \end{cases} \quad (10)$$

```
\begin{equation}
f(x)=
\begin{cases}
x & \text{\text{gdy } $x>5$}\\
0 & \text{\text{gdy } $0\ge x\ge 5$}\\
-x & \text{\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 `\text`. Jest ono równoważne poleceniu `\textrm` w \LaTeX-u . Środowisko `cases` jest szczególnym przypadkiem środowiska `array`.

13. Pakiet amscd

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

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

musimy, po wczytaniu w preambule dokumentu pakietu napisać:

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

Ten diagram jest tablicą składającą się z dwóch rodzajów rzędów: z poziomymi ze strzałkami poziomymi i pionowymi 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 `ˆ` lub `ˆ` 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.

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

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

Powyższe diagramy otrzymaliśmy następująco:

```


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


```

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

A teraz jeszcze inny przykład prostego diagramu.

Przykład 2.

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

13.1. Pakiet diagrams

To przykłady diagramów uzyskanych przy pomocy pakietu `diagrams`:

$$\begin{array}{ccccc}
 A * (B * (C * D)) & \xrightarrow{\text{assl}} & (A * B) * (C * D) & \xrightarrow{\text{assl}} & ((A * B) * C) * D \\
 \text{id} * \text{assl} \downarrow & & & & \uparrow \text{assl} * \text{id} \\
 A * ((B * C) * D) & \xrightarrow{\text{assl}} & & & (A * (B * C)) * D
 \end{array}$$

Każda strzałka może być opatrzona etykietą której miejsce zależy od użytego oznaczenia i to:

$\overset{\sim$ na górze, $\underset{\sim}{}$ na dole, $\overset{<}{}$ z lewej, $\overset{>}{}$ z prawej i $\overset{\sim}{}$ pośrodku.

Zastosowanie w poszczególnych rodzajach strzałek pokazuje następująca lista

- dla strzałek **poziomych** polecenie `\rTof_g` and `\rTo{f}{g}` daje \xrightarrow{f}_g ;
- dla strzałek **pionowych** polecenia `\dTo <f >g`, `\dTof_g` i `\dTo{f}{g}` dają $f \downarrow g$;
- dla strzałek **ukośnych** o dodatnim nachyleniu pisząc `\ldTof_g`, `\ldTo <f >g` i `\ldTo{f}{g}` otrzymamy w każdym przypadku \xrightarrow{f}_g (podobnie `\ruTo`).
- ale dla strzałek o nachyleniu **ujemnym** na górze= z prawej i na dole= a lewej, więc `\rdTof_g`, `\rdTo <g >f` i `\rdTo{f}{g}` dają $g \searrow^f$ (podobnie `\luTo`);
- Użycie tyldy spowoduje, że etykieta będzie umieszczona w przerwie w strzałce: `\rTof A \relax ^* B {}_s \lTo_g C`

Poziome strzałki mogą być również umieszczane we wzorach w jednej linii. Wtedy piszemy po prostu:

`$ A \rTof \relax ^* B {}_s \lTo_g C $`

i otrzymujemy

$$A \xrightarrow{f} {}^*B_s \xleftarrow{g} C.$$

Można również zmieniać wielkość fontu, który chcemy użyć do napisania wyrażenia nad strzałką.

$$A \xrightarrow[\scriptscriptstyle]{\sum_1^n f_i} B \quad A \xrightarrow[\script]{\sum_1^n f_i} B \quad A \xrightarrow[\text{text}]{\sum_1^n f_i} B \quad A \xrightarrow[\displaystyle]{\sum_1^n f_i} B$$

Powyższy efekt otrzymaliśmy pisząc:

```
\begin{displaymath}
\hss\begin{diagram}[inline,labelstyle=\scriptscriptstyle]
A & \rTo^{\sum^n_1 f_i}_{\rm scriptscript} & B
\end{diagram}
\quad
\begin{diagram}[inline,labelstyle=\scriptstyle]
A & \rTo^{\sum^n_1 f_i}_{\rm script} & B
\end{diagram}
\quad
\begin{diagram}[inline,labelstyle=\textstyle]
A & \rTo^{\sum^n_1 f_i}_{\rm text} & B
\end{diagram}
\quad
\begin{diagram}[inline,labelstyle=\displaystyle]
A & \rTo^{\sum^n_1 f_i}_{\rm display} & B
\end{diagram}
\hss\end{displaymath}
```

$$\begin{array}{ccc} & & f \\ & & \rightarrow \\ A & \xrightarrow[\alpha]{+} & B \\ & \uparrow g & \\ & & \\ h & \downarrow & k \\ & c & \end{array}$$

Inne pakiety wspomagające składanie diagramów w \LaTeX -u to `array`, `kuvio`, `DCpic`, `xymatrix`, `diagxy`.

14. Tworzenie list

W \LaTeX -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.

14.1. Środowisko itemize i enumerate

W środowisku wypunktowującym (**itemize**) tekst kolejnych punktów poprzedzony jest grubą czarną kropką jako etykietą. Kropka taka jest wywoływana w \TeX -u poleceniem $\text{\$}\text{\texttt{bullet}}\text{\$}$ (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.

14.2. 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 umieszcza 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.	<code>\newcounter{numer}</code> <code>\begin{list}{*}</code> <code>{\usecounter{numer}}</code>
* To jest punkt drugi.	<code>\item To jest punkt pierwszy.</code>
* To jest punkt trzeci.	<code>\item To jest punkt drugi.</code>
* To jest punkt czwarty.	<code>\item To jest punkt trzeci.</code>
	<code>\item To jest punkt czwarty.</code> <code>\end{list}</code>
\Rightarrow -1 To jest punkt pierwszy.	<code>\begin{list}</code> <code>{\\$\rightarrow\$\arabic{numer}}</code> <code>{\usecounter{numer}}</code>
\Rightarrow -2 To jest punkt drugi.	<code>\item To jest punkt pierwszy.</code>
\Rightarrow -3 To jest punkt trzeci.	<code>\item To jest punkt drugi.</code>
	<code>\item To jest punkt trzeci.</code> <code>\end{list}</code>
A-I To jest punkt pierwszy.	<code>\newcounter{numerek}</code> <code>\begin{list}{A--\Roman{numerek}}</code> <code>{\usecounter{numerek}}</code>
A-II To jest punkt drugi.	<code>\item To jest punkt pierwszy.</code>
A-III To jest punkt trzeci.	<code>\item To jest punkt drugi.</code>
	<code>\item To jest punkt trzeci.</code> <code>\end{list}</code>

14.3. 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}
        \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}
        \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}
  \item I znowu poziom trzeci.
  \end{itemize}
\item Wracamy na drugi poziom
\end{enumerate}
\item I wreszcie na poziomie pierwszym.
\end{itemize}

```

14.4. 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 1.

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 2.

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

Zadanie 3.

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 4.

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**