

# Jak używać $\LaTeX$ 'a i przeżyć

Tomasz Kwiatkowski

ver. 0.9, 14 gru 1998

## 1 Pliki wejściowe $\LaTeX$ 'a

Na wejściu  $\LaTeX$  wymaga podania zwykłego pliku tekstowego, zawierającego kody ASCII. Można go stworzyć w dowolnym edytorze typu `vi`, `pico` czy `joe`. Musi on zawierać tekst dokumentu wzbogacony komendami które powiedzą  $\LaTeX$ 'owi, jak go sformatować.

Jeśli piszemy po polsku, musimy dodatkowo zadbać o dostępność polskich liter w używanym edytorze. Pracując w środowisku graficznym X-Windows należy więc otworzyć `x` terminal, który został poinformowany, że będziemy w nim chcieli wyświetlać polskie litery. W pracowni komputerowej OA UAM taki spolszczony `x` terminal uzyskamy pisząc w linii komend `xterm-pl &`. Następnie musimy uruchomić swój ulubiony edytor tekstu. Jeśli np. korzystamy z edytora `joe`, wówczas wywołujemy go z opcją `-asis` pisząc:

```
joe -asis nazwa.zbioru.
```

Po otwarciu edytora polskie litery uzyskamy wciskając jednocześnie *prawy* ALT i odpowiednią literę.

### 1.1 Z pustego w próżne, czyli 5 spacji = 1 spacja

Tzw. „białe znaki”, czyli spacje lub tabulatory traktowane są przez  $\LaTeX$ 'a tak, jak pojedynczy odstęp. Tak więc nie ma różnicy czy do pliku wstawimy jedną, czy pięć spacji. Ta cecha może wywołać zdumienie użytkowników edytorów tekstu typu WYSIWYG (WordPerfect, StarWriter, Microsoft Word) lecz — jak większość rzeczy w  $\LaTeX$ 'u — ma ona swoje uzasadnienie. Do przesuwania wyrazów w wierszu służą odpowiednie komendy formatujące, np. komenda `\hspace{2cm}` wstawia pusty odstęp o długości 2 cm.

Pusta linia pomiędzy dwoma wierszami tekstu oznacza koniec paragrafu. I znowu — kilka kolejnych, pustych linii traktowanych jest tak samo, jak pojedyncza pusta linia. Do uzyskiwania większych odstępów np. między paragrafami

służy komenda `\vspace`. Przykładowo: `\vspace{1cm}` wstawia w pionie pusty dostęp o wysokości 1 cm.

## 1.2 10 znaków specjalnych

W L<sup>A</sup>T<sub>E</sub>X'u istnieje zestaw znaków, które sterują jego pracą. Gdybyśmy któryś z nich wstawili w tekst, na ogół nie zostałby on wydrukowany lecz spowodował jakąś reakcję ze strony L<sup>A</sup>T<sub>E</sub>X'a. Do znaków tych zaliczamy:

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

Uważniejszy czytelnik spyta zapewne, jak znaki te zostały zakodowane, skoro jednak zostały wydrukowane w tym tekście. Jest to bardzo proste, wystarczy każdy z nich poprzedzić „w-tył-ciachem”, o tak:

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

Pozostałe trzy znaki specjalne można uzyskać w inny sposób, np. „w-tył-ciach” kodujemy jako `\backslash`.

## 1.3 Cudzysłowy, myślniki i minusy

Polski cudzysłów uzyskamy pisząc `„słowa w cudzysłowie”`. L<sup>A</sup>T<sub>E</sub>X przetworzy to w: „słowa w cudzysłowie”.

Nieco bardziej złożony jest problem myślników. Można je uzyskać poprzez *pauzę* lub *półpauzę*. Pauzę zapisujemy w L<sup>A</sup>T<sub>E</sub>X'u jako `---` (trzy znaki minus na klawiaturze) a półpauzę jako `--`. Po przetworzeniu da nam to, odpowiednio, `—` oraz `-`. Jest jeszcze tzw. *dywiza*, kodowana w postaci pojedynczego minusa, którą wykorzystujemy w wyrazach złożonych. Użycie tych kresek w praktyce ilustruje poniższy przykład (podwójne „w-tył-ciachy” stosuje się tu do oznaczenia końca wiersza): tekście:

```
--- Kto tam? \\
--- Zawisza-Czarny. \\
--- Zostaw miecz przed drzwiami. \\
```

Po przetworzeniu tego tekstu L<sup>A</sup>T<sub>E</sub>X'em otrzymamy:

```
— Kto tam?
— Zawisza-Czarny.
— Zostaw miecz przed drzwiami.
```

Kwestia poprawnego kodowania dywiz, półpauz i pauz jest obecnie szczególnie istotna. Coraz częściej można napotkać książki i czasopisma składane przez półprofesjonalistów, w których namiętnie używa się dywizy w każdej z powyższych sytuacji.

W  $\LaTeX$ 'u jest jeszcze jedna kreseczka o długości innej, niż powyższe, a mianowicie znak minusa. Stosujemy go w środowisku matematycznym i kodujemy za pomocą pojedynczego minusa (powrócimy do tej kwestii przy omawianiu pisania wzorów matematycznych).

## 1.4 Komentarze

Chcąc zamieścić w tekście swoje uwagi w charakterze komentarza, który nie będzie przetwarzany przez  $\LaTeX$ 'a, należy poprzedzić je znakiem `%`. Poniższy fragment tekstu źródłowego to wyjaśnia:

```
%
% Tu wyjaśniamy, jak pisać różne kreseczki
%
Nieco bardziej złożony jest problem myślników.
Można go uzyskać poprzez
\emph{pauzę} lub \emph{półpauzę}. % Rozwinąć
Pauzę zapisujemy w \LaTeX{}'u jako
\verb|---| (trzy znaki minus na klawiaturze)
a półpauzę jako \verb|--|. Po przetworzeniu
```

## 1.5 Zabezpieczanie grup wyrazów przed podziałem

Spójrzmy na poniższy rezultat przetworzenia tekstu  $\LaTeX$ 'em:

System komputerowego składu tekstu  $\TeX$  został stworzony przez D. Knuth'a. W celu ułatwienia posługiwania się nim w praktyce, L. Lamport rozszerzył go o zestaw gotowych makroinstrukcji i stylów dokumentów, co dało początek  $\LaTeX$ 'owi.

Jak widać inicjały imion obu panów zostały oddzielone od ich nazwisk, które trafiły do następnych wierszy. Aby tego uniknąć, pomiędzy skrót imienia i nazwisko należy wstawić *tyldę*: L.~Lamport, D.~Knuth.  $\LaTeX$  traktuje słowa połączone tyldą jak jeden wyraz i nie rozdziela je pomiędzy dwa wiersze. Podobnie należałoby np. zapisać skrótową nazwę Obserwatorium Astronomicznego UAM: OA~UAM, co da nam pewność, że oba wyrazy znajdują się w tym samym wierszu.

## 1.6 Sposób zapisu komend L<sup>A</sup>T<sub>E</sub>X'a

Każdą komendę L<sup>A</sup>T<sub>E</sub>X'a poprzedza „w-tył-ciach”; mamy więc przykładowo komendę `\newpage`, która wymusza rozpoczęcie nowej strony. Wiele komend wymaga podania dodatkowych parametrów — umieszcza się je wówczas w nawiasach klamrowych. Przykładowo, wspomniana już przez nas komenda `\hspace{1cm}` wymaga podania poziomego odstępu w wybranej jednostce miary, w tym przypadku odstęp ten ma wynosić 1cm. Jeśli komenda kończy się literą, a nie np. zamykającym nawiasem klamrowym, musimy po niej zostawić spację. Nie należy więc pisać

```
\newpageTu zaczyna sie nowa strona tekstu,
```

choć można napisać

```
\hspace{1cm}Tu wstawiono odstęp.
```

## 2 Struktura pliku wejściowego

Plik wejściowy dla L<sup>A</sup>T<sub>E</sub>X'a (a dokładniej, dla wersji L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>), rozpoczynamy komendą

```
\documentclass{...},
```

gdzie zamiast przecinków wstawić można *styl dokumentu*. O dostępnych w L<sup>A</sup>T<sub>E</sub>X'u stylach powiemy niżej. Dokumenty przeznaczone dla poprzedniej wersji 2.09 L<sup>A</sup>T<sub>E</sub>X'a odróżniają się już na pierwszy rzut oka tym, że rozpoczynają się komendą `\documentstyle{...}`. Na ogół można je jednak przetworzyć L<sup>A</sup>T<sub>E</sub>X'em 2<sub>ε</sub> bez większych problemów.

Następnie można wpisać komendy, które zmieniają wygląd całego dokumentu (np. zmieniają rozmiary marginesów, odstęp pomiędzy wierszami, itp.) a także wczytać dodatkowe *pakiety*, które rozszerzają możliwości podstawowej wersji L<sup>A</sup>T<sub>E</sub>X'a. Pakiety wczytujemy komendą

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

Kiedy mamy już zdefiniowany całościowy wygląd dokumentu, możemy zacząć wprowadzać tekst, rozpoczynając go komendą

```
\begin{document}
```

Cały obszar pliku pomiędzy `\documentclass{...}`, a `\begin{document}`, zwany jest *preambułą*.

Po zakończeniu pisania treści dokumentu zmieszanej z komendami formatującymi, powiadamy L<sup>A</sup>T<sub>E</sub>X'a o zakończeniu dokumentu komendą

```
\end{document}
```

Wszystko, co pojawi się poniżej, będzie ignorowane.

Najkrótszy poprawny pliku wejściowy L<sup>A</sup>T<sub>E</sub>X'a wygląda następująco:

```
\documentclass{article}
\begin{document}
Small is beautiful.
\end{document}
```

Jak widać, brak tu polskich liter, gdyż korzystanie z nich wymaga wczytania dodatkowych pakietów.

## 3 Format dokumentu

### 3.1 Style dokumentu

W L<sup>A</sup>T<sub>E</sub>X'u mamy kilka gotowych stylów dokumentu. Podajemy je jako argument komendzie `\documentclass`

```
\documentclass[opcja1,opcja2,...]{styl dokumentu}
```

Nie nazywa się ona `\documentstyle`, choć tak było by poprawniej, gdyż `documentstyle` występował w programie L<sup>A</sup>T<sub>E</sub>X 2.09, a chcemy je od siebie odróżnić).

W Tabeli 1 przedstawiono przegląd stylów dokumentów, opisanych w niniejszym wprowadzeniu. Dodatkowe parametry *opcje* pozwalają dopasować każdy ze stylów do aktualnych potrzeb. Opis niektórych parametrów zawiera Tabela 2.

Przykład: Plik wejściowy dla L<sup>A</sup>T<sub>E</sub>X'a mogłaby rozpoczynać komenda:

```
documentclass[11pt,twoside,a4paper]{article}
```

która instruuje L<sup>A</sup>T<sub>E</sub>X'a by złożył tekst w stylu artykułu czcionką o rozmiarze 11 punktów, i zastosował format dogodny do dwustronnego druku na papierze A4.

### 3.2 Pakiety dodatkowe

Często okazuje się, że podstawowe możliwości L<sup>A</sup>T<sub>E</sub>X'a są niewystarczające. Jeśli chcemy wstawić w tekst grafikę, użyć kolorów lub choćby znaków polskiego al-

fabetu, musimy posłużyć się odpowiednimi pakietami rozszerzającymi. Wczytuje się je komendą

```
usepackage[opcja1,opcja2,...]{pakiet}
```

Niektóre pakiety znajdują się w podstawowej dystrybucji L<sup>A</sup>T<sub>E</sub>X'a, inne, czasem nawet bardzo specjalistyczne, można odszukać w bogatych archiwach internetowych. Przykładem tej różnorodności jest np. pakiet `calendar`, służący do tworzenia różnych kalendarzy w wielu językach, pakiet `crosswd` do tworzenia krzyżówek, pakiet `cchess` do drukowania rozgrywek w chińskich szachach lub pakiet do drukowania menu restauracji.

*Opisać pakiety służące do uzyskiwania polskich liter. Wspomnieć o pakietach do włączania grafiki (`epsfig`, `graphixc`).*

### 3.3 Style strony

L<sup>A</sup>T<sub>E</sub>X udostępnia trzy predefiniowane kombinacje nagłówek i stopek, zwane *stylami strony*. Możemy je zmieniać komendą

```
pagestyle{styl}
```

w której parametr *styl* może być jednym z trzech słów kluczowych: *plain*, *he-*

Tablica 1: Style dokumentu

---

<code>article</code>	do pisania prac naukowych, artykułów, prac semestralnych, raportów z ćwiczeń obserwacyjnych, dokumentacji oprogramowania itp.
<code>report</code>	do dłuższych tekstów, zawierających kilka rozdziałów, broszur, prac magisterskich, doktoratów, itp.
<code>book</code>	do składania „prawdziwych” książek
<code>letter</code>	do pisania listów
<code>slides</code>	do folii dla grafoskopu. Ten styl używa dużych liter bezszeryfowych i może być zastąpiony pakietem <code>FoilT<sub>E</sub>X</code> , dającym inny wygląd folii.

---

Tablica 2: Opcje stylów dokumentów

---

<code>10pt, 11pt, 12pt</code>	Ustala rozmiar czcionki w dokumencie w tzw. <i>punktach amerykańskich, pt</i> (1pt = 0.35146mm). Gdy nie określono rozmiaru czcionek, L <sup>A</sup> T <sub>E</sub> X zastosuje czcionki 10-cio punktowe.
<code>a4paper</code>	Określa rozmiar stosowanych kartek papieru. Domyślnie przyjmowaną wielkością jest <code>letterpaper</code> , używany powszechnie w USA. Innymi możliwymi formatami są: <code>a5paper</code> , <code>b5paper</code> , <code>executivepaper</code> i <code>legalpaper</code> .
<code>landscape</code>	Odwraca stronę o 90°, pozwalając na druk np. szczególnie szerokich tabel bądź wykresów. By uzyskać taki wynik należy plik <code>*.dvi</code> przetworzyć w PostScript komendą <code>dvips -t landscape -f plik.dvi &gt; plik.ps</code> . Dopiero wówczas można otrzymać na wydruku wymagany efekt.
<code>fleqn</code>	Zamiast pośrodku, wzory matematyczne będą wyrównane do lewego marginesu.
<code>leqno</code>	Numeracja wzorów matematycznych pojawi się z lewej strony, zamiast z prawej.
<code>titlepage, notitlepage</code>	Opcja używana tylko w stylu <i>article</i> . Powoduje, że tytuł i streszczenie pojawią się na osobnych stronach, a nie na pierwszej stronie właściwego tekstu.
<code>twocolumn</code>	Służy do składu tekstu w dwóch kolumnach, zwiększając jego czytelność.
<code>twoside, oneside</code>	Określa, czy ma powstać dwustronna ( <i>twoside</i> ), czy jednostronna ( <i>oneside</i> ) wersja dokumentu. Wersje te różnicuje np. szerokość marginesów. Użycie parametru <i>twoside</i> nie znaczy jednak, że z drukarki wyjdą dwustronnie zadrukowane kartki. Wymaga to specjalnych drukarek lub osobnego druku parzystych i nieparzystych stron.

---

*adings*, *empty*. Ich znaczenie wyjaśniono w Tabeli 3. W praktyce do definiowania

Tablica 3: Predefiniowane style stron w L<sup>A</sup>T<sub>E</sub>X'u

---

*plain* drukuje numery stron na dole strony, pośrodku stopki. Jest to opcja domyślna.

*headings* drukuje tytuł bieżącego rozdziału oraz numer strony w nagłówku, stopka natomiast pozostaje pusta.

*empty* zarówno stopka jak i nagłówek pozostają puste.

---

stylu strony stosuje się pakiet `fancyhdr`, który w bardziej elastyczny sposób umożliwia określenie zawartości nagłówków i stopek. Wczytujemy go tak, jak i inne pakiety dodatkowe:

```
\usepackage{fancyhdr}
```

po czym informujemy L<sup>A</sup>T<sub>E</sub>X'a, że chcemy określać styl strony przy użyciu poleceń tego właśnie pakietu:

```
\pagestyle{fancy}
```

Pora na zdefiniowanie nagłówków i stopek. Spójrzmy na poniższy przykład:

```
\lhead{Jan Kowalski}
\chead{}
\rhead{Fotometria planetoid}
\lfoot{}
\cfoot{\thepage}
\rfoot{}
```

Jak łatwo się domyślić, pierwsza litera komend określa miejsce umieszczenia tekstu (*l* — po lewej, *c* — pośrodku, *r* — po prawej), natomiast pozostała część to angielska nazwa nagłówka (*head*) bądź stopki (*foot*). Komenda `\thepage` powoduje wstawienie we wskazanym miejscu numeru bieżącej strony. W powyższym przykładzie w nagłówku strony po lewej znajdzie się imię i nazwisko autora, po prawej zaś tytuł artykułu; pośrodku stopki zostanie umieszczony numer strony. Użycie pakietu `fancyhdr` w przypadku druku dwustronnego wymaga podania innych komend, różnicujących parzyste i nieparzyste strony.

### 3.4 Praca z długimi tekstami

Gdy nasz dokument się rozrasta (np. dopisujemy do pracy semestralnej kolejne rozdziały tak, by przekształciła się ona w pracę magisterską), wygodnie jest mieć możliwość przetwarzania w L<sup>A</sup>T<sub>E</sub>X'u wybranych części całości, bez konieczności dodawania do każdego takiego fragmentu wspólnej preambuły. W takich sytuacjach możemy posłużyć się komendą

```
\include{nazwa pliku}
```

która we wskazanym miejscu wstawi nam odpowiedni plik z dysku, zawierający np. kolejny rozdział. Całość dokumentu może wówczas wyglądać następująco:

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

\usepackage{...}
.....
\usepackage{...}

\includeonly{rozdzial2,rozdzial3}

\begin{document}

\include{wstep}
\include{rozdzial1}
\include{rozdzial2}
\include{rozdzial3}
\include{zakonczenie}
\include{literatura}
```

Proszę zwrócić uwagę na dodatkową komendę `\includeonly{...}`, która instruuje L<sup>A</sup>T<sub>E</sub>X'a, by spośród napisanych już części dokumentu uwzględnił w czasie przetwarzania tylko rozdziały 2 i 3. Może to być wygodne gdy np. chcemy wydrukować fragment pracy do sprawdzenia przez promotora.

Stosując powyższe dwie komendy należy pamiętać, że L<sup>A</sup>T<sub>E</sub>X zawsze rozpoczyna każdy z nich od nowej strony zakładając niejako, że są to większe fragmenty tekstu, np. poszczególne rozdziały. Jeśli chcemy wstawiać mniejsze części tekstu, np. podrozdziały lub wręcz wycięte ze środka fragmenty, należy posłużyć

się komendą

```
\input {nazwa pliku}
```

która nie posiada takich ograniczeń. W tym wypadku mamy pełną swobodę, nie możemy jednak korzystać z `\includeonly{}` — chcąc powstrzymać L<sup>A</sup>T<sub>E</sub>X'a od wczytania któregoś z fragmentów, zawartego w komendzie `\input{}`, wystarczy opatrzyć ją znakiem komentarza.

Składanie wzorów matematycznych

## 4 Otoczenie matematyczne

W L<sup>A</sup>T<sub>E</sub>X'u istnieje kilka otoczeń, w których można zapisywać wzory matematyczne. W najprostszym przypadku — gdy wstawiamy w tekst jednego paragrafu krótki wzór — używamy otoczenia *math*, czyli wpisujemy wzór pomiędzy komendy `\begin{math}` i `\end{math}`. Żeby przyspieszyć pisanie, można wzór zawrzeć między dwoma znakami `$`. I wreszcie, jeśli terminal, na którym pracujemy, nie posiada znaków `$`, możemy zamiast tego zapisać wzór pomiędzy znakami `\(` oraz `\)`.

W trójkącie prostokątnym suma kwadratów boków `$a$` i `$b$` da nam kwadrat boku `$c$`. Można to zapisać w postaci wzoru: `$a^2+b^2=c^2$`

co po przetworzeniu L<sup>A</sup>T<sub>E</sub>X'em da nam:

W trójkącie prostokątnym suma kwadratów boków *a* i *b* da nam kwadrat boku *c*. Można to zapisać w postaci wzoru:  $a^2 + b^2 = c^2$

W powyższym przykładzie znak `^` użyto w celu uzyskania wykładnika.

Dłuższe wzory czytelniej jest prezentować w otoczeniu *displaymath*. Oznacza to, że wpisujemy je pomiędzy komendami `\begin{displaymath}` i `\end{displaymath}` (lub, jeśli nie lubimy tak dużo pisać, między `\[` oraz `\]`).

W trójkącie prostokątnym suma kwadratów boków *a* i *b* da nam kwadrat boku *c*. Można to zapisać w postaci wzoru:

$$a^2 + b^2 = c^2$$

Wzór ten znany jest jako twierdzenie Pitagorasa.

Chcąc uzyskać numerację wprowadzanych wzorów należy użyć otoczenia *equation*.

W trójkącie prostokątnym suma kwadratów boków  $a$  i  $b$  da nam kwadrat boku  $c$ . Można to zapisać w postaci wzoru:

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

Wzór 1 znany jest jako twierdzenie Pitagorasa.

Zauważmy, że w powyższym przykładzie użyto też odnośnika do równania. Wewnątrz otoczenia *equation* umieszczono znaną nam już komendę `\label{...}`, za pomocą której nadano równaniu etykietę *Pitagoras*, do której odwołano się w dalszej części tekstu. L<sup>A</sup>T<sub>E</sub>X sam ustala numer porządkowy równania co zwalnia nas od obowiązku wpisywania go w czasie odwoływania się doń — o ileż prostsze jest pamiętanie, że mamy odwołać się do równania *Pitagoras* niż do równania (1.1) (dodatkową zaletą tego systemu jest automatyczna zmiana numerów równań i odwołań po zmianie kolejności prezentacji równań).

Wymienimy teraz niektóre różnice między otoczeniem matematycznym a tekstowym. Przykładowo, w otoczeniu matematycznym:

1. Do uzyskiwania odstępów np. między kilkoma częściami wzoru, używa się specjalnych komend (np. `\`, czy `\quad`) o których będziemy jeszcze mówili
2. W ramach jednego otoczenia może znaleźć się tylko jeden paragraf, zatem puste linie nie mają żadnego znaczenia
3. Każda litera jest traktowana jako zmienna i tak też jest drukowana; jeśli chcemy uzyskać normalne litery tekstu, należy je wpisać używając komendy `\text{rm}{...}`.

Przykład naruszenia jednej z tych zasad i wynikające z tego efekty prezentuje poniższy przykład (w trakcie przetwarzania tekstu źródłowego, L<sup>A</sup>T<sub>E</sub>X poinformował o niewłaściwie użytym znaku końca linii):

$$2a + 3x = 4c3a + x = 10d$$

## 5 Podstawowe operacje

### 5.1 Cztery działania

Jak widzieliśmy na przykładach, zapis dodawanie i odejmowania jest trywialny. Mnożenie i dzielenie uzyskujemy następująco:

$ab$   
 $a \cdot b$   
 $a \times b$   
 $ab$   
 $a/b$   
 $\frac{a}{b}$

Zauważmy, że zapis w linii 1 i 4 dał ten sam rezultat. Użycie symbolu `\*` ma tę przewagę, że w sytuacji, gdy L<sup>A</sup>T<sub>E</sub>X postanowi złamać wzór pomiędzy symbolami  $a$  i  $b$ , wówczas na końcu starej linii doda symbol mnożenia  $\times$ .

Dalsze przykłady pokazują nieco bardziej skomplikowane wzory:

$$(a + b) \cdot (a - b) = a^2 - b^2$$

$$(a + b) \cdot \frac{x}{x + \frac{1}{y}}$$

$$\frac{1 + a \times (b + \frac{c}{d+1})}{x + y}$$

## 5.2 Indeksy górne i dolne, znaki relacji itp.

Indeksy górne zapisujemy tak samo, jak wykładniki potęgi (czyli za pomocą symbolu `^`; indeksy dolne uzyskujemy stosując symbol `_`:

$$a_1 \quad x_{ij}^2 \quad e^{-xt} \quad e^{x^2} \neq e^2$$

W powyższym przykładzie występuje jeden z operatorów relacji, znak *nierówne*. Inne, często stosowane operatory relacji, zawarto poniżej:

$$a < b \quad b > c \quad x^2 \leq 1 \quad y \geq 0 \quad x \sim 0$$

$$z \simeq 1 \quad x \approx y \quad x \ll y \quad y \gg x$$

Wykaz wszystkich dostępnych operatorów relacji zawiera tabela ??.

Zobaczymy jeszcze, jak różnią się zwykłe liczby, zapisane w otoczeniu tekstowym i matematycznym:

$$-1 + 3 \ 2 = 7$$

$$-1 + 3 \ 2 = 7$$

Wniosek z powyższego jest prosty: jedynie pojedyncze liczby dodatnie można zapisywać w otoczeniu tekstowym; liczby ujemne i jakiegokolwiek relacje między liczbami wymagają już otoczenia matematycznego.

Greckie litery zapisujemy korzystając z ich angielskich nazw, np.

$$\alpha, \mu, \rho, \Sigma$$

Wykaz wszystkich dostępnych liter alfabetu greckiego zawierają tabele ?? i ??.

### 5.3 Inne, często używane funkcje

Pierwiastkowanie zapisujemy za pomocą komendy

$$\backslash\text{sqrt}[n]\{\text{argument}\}$$

gdzie parametr  $n$ , oznaczający stopień pierwiastka, można pominąć dla pierwiastka kwadratowego.

$$\sqrt{x} \quad \sqrt{x^2 + \sqrt{y}} \quad \sqrt{x^2 + y^2}$$

Pozostałe, często używane funkcje, np. trygonometryczne, logarytmiczne, mają zdefiniowane odpowiednie komendy. Mamy więc `\sin`, `\arctan`, `\log` itp. Zrobiono tak, gdyż funkcje te zapisujemy we wzorach normalną czcionką, a nie pochyłą, jak oznaczane są zmienne. Komplet dostępnych funkcji prezentujemy poniżej:

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

### 5.4 Granice ciągu, pochodne, sumy i całki

Zasady zapisu granicy ciągu najlepiej wyjaśni poniższy przykład:

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$$

Zobaczmy teraz, że ten sam wzór, zapisany w otoczeniu *math* (czyli pomiędzy znakami  $\$$ ), będzie wyglądał inaczej:

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$$

Dzieje się tak, ponieważ otoczenie *math* stworzono do prezentacji *krótkich* wzorów wewnątrz paragrafu, i dlatego stara się ono oszczędzać miejsce, zapisując wzory — w miarę możliwości — na jednym poziomie.

Pochodne zapisujemy, korzystając z akcentów

$$y = x^2 \quad y' = 2x \quad y'' = 2$$

$$y = x^2 \quad \frac{dy}{dx} = 2x \quad \frac{d^2y}{dx^2} = 2$$

Zauważmy, że zapisując pochodną jako iloraz różniczek, konieczne było użycie normalnej niepochyłonej czcionki; stąd też komenda `\textrm{...}`, która składa umieszczone w nawiasie symbole normalną czcionką. Można ułatwić sobie pisanie tej, dość długiej, komendy `\textrm{...}`, definiując odpowiednią, nową, komendę. Nie będziemy jednak teraz tego wyjaśniać.

Do uzyskiwania sum i całek stosujemy, odpowiednio, komendy `\sum` i `\int`

$$\sum_{i=1}^n \int_0^{\frac{\pi}{2}}$$
$$\sum_i z_i + \int_0^1 f(x)dx$$