

Podstawy informatyki

WYKŁAD nr 04

Fizyka Techniczna, WFT PP

Michał Hermanowicz

Zakład Fizyki Obliczeniowej, Instytut Fizyki, Politechnika Poznańska

Rok akademicki 2018/2019



- 1 Wprowadzenie
- 2 Pliki źródłowe
 - Format i przykład
 - Preambuła, pakiety i klasy dokumentu
 - Treść
 - Wyróżnienia, otoczenia, listy, tabele
 - Rysunki
 - Wyrażenia matematyczne
- 3 Spisy bibliograficzne
- 4 Tworzenie prezentacji
- 5 Analiza przykładowych dokumentów
- 6 Podsumowanie: pytania i dyskusja

Plan ramowy przedmiotu

Nr wykładu	Poruszane zagadnienia
I	Organizacja; forma i warunki zaliczenia; wprowadzenie
II	Powłoka bash i elementy programowania
III	Przetwarzanie danych #1
IV	Przetwarzanie danych #2
V	Reprezentacja danych (wykresy 2D i 3D) – <i>gnuplot</i>
VI	System składu tekstu L ^A T _E X
VII	Pół-otwarty test zaliczeniowy

Każdemu z wykładów odpowiadają ćwiczenia realizowane na zajęciach w pracowni komputerowej.

Czym jest L^AT_EX? Za stroną domową projektu*:

L^AT_EX is a high-quality typesetting system; it includes features designed for the production of technical and scientific documentation. L^AT_EX is the de facto standard for the communication and publication of scientific documents. L^AT_EX is available as free software.

* <http://www.latex-project.org>

Czym jest L^AT_EX? Za stroną domową projektu*:

L^AT_EX is a high-quality typesetting system; it includes features designed for the production of technical and scientific documentation. L^AT_EX is the de facto standard for the communication and publication of scientific documents. L^AT_EX is available as free software.

* <http://www.latex-project.org>

- L^AT_EX – «Lah-tech» lub «Lay-tech»;
- profesjonalny system składu tekstu: książek, artykułów, sprawozdań, prac dyplomowych (inżynierskich, magisterskich, doktorskich, ...), prezentacji multimedialnych i innych rodzajów publikacji;
- L^AT_EX **nie jest** ani edytorem, ani procesorem tekstu;
- pierwotnie T_EX (język/kompilator), który uzupełniony przez zbiór makr (Leslie Lamport) stanowi system L^AT_EX i jest dostępny w wielu *dystrubucjach*.

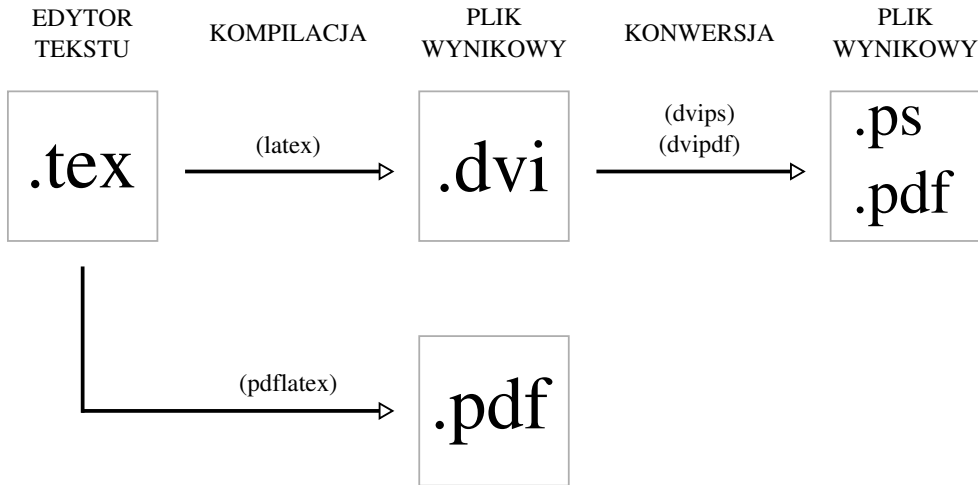
Skąd czerpać informacje?

- Strona domowa projektu: <http://www.latex-project.org>
- *Nie za krótkie wprowadzenie do systemu L^AT_EX* – skrypt (PL; licencja: GNU GPL), <http://www.ctan.org/tex-archive/info/lshort/polish/lshort2e.pdf>
- Inne (zasoby Internetu)

Oprogramowanie:

- TeX Live (<http://tug.org/texlive>) – dystrybucja dostępna na wiele platform systemowych (m.in. GNU/Linux oraz systemy MS Windows),
- MiKTeX (<http://miktex.org/>) – dystrybucja dla systemów MS Windows,
- inne dystrybucje (teTeX, proTeXt i in.),
- edytory/środowiska do pracy z L^AT_EX: Texmaker, TeXstudio, **Vim** i inne.

Wprowadzenie – cykl pracy



Dlaczego nie edytor tekstu zamiast \LaTeX ?

Dlaczego nie edytor tekstu zamiast \LaTeX ?

Edytory służą do innych celów.

Dlaczego nie edytor tekstu zamiast L^AT_EX?

Edytory służą do innych celów.

Dlaczego nie procesor tekstu zamiast L^AT_EX?

Powodów (zalet) jest kilka:

- autor skupia się wyłącznie na **treści** dokumentu pozostawiając jego formatowanie *innym*,
- system pozwala w łatwy i automatyczny sposób wygenerować spisy treści, rysunków, tabel, a także odnośniki, skorowidze i bibliografie,
- profesjonalny wygląd generowanych dokumentów – zgodny z zasadami sztuki drukarskiej,
- oprogramowanie dostępne na wiele platform systemowych/sprzętowych, generowane pliki wynikowe są przenośne i poprawnie odczytywane na różnych urządzeniach i w różnych systemach operacyjnych.

Prosty plik źródłowy L^AT_EX

Plik z rozszerzeniem .tex

jest plikiem tekstowym (ASCII). Zawiera treść dokumentu oraz instrukcje sterujące – rozpoczynające się od znaku \ (***backslash***). Plik może zawierać komentarze (ignorowane przez kompilator) – te muszą być poprzedzone symbolem %. Inne znaki specjalne: \$ { } # - & (w tekście poprzedzane symbolem \).

Prosty plik źródłowy L^AT_EX

Plik z rozszerzeniem .tex

jest plikiem tekstowym (ASCII). Zawiera treść dokumentu oraz instrukcje sterujące – rozpoczynające się od znaku \ (**backslash**). Plik może zawierać komentarze (ignorowane przez kompilator) – te muszą być poprzedzone symbolem %. Inne znaki specjalne: \$ { } # - & (w tekście poprzedzane symbolem \).

Składnia instrukcji:

```
\instrukcja[arg. opcjonalne]{argumenty}
```

Prosty plik źródłowy L^AT_EX

Plik z rozszerzeniem .tex

jest plikiem tekstowym (ASCII). Zawiera treść dokumentu oraz instrukcje sterujące – rozpoczynające się od znaku \ (**backslash**). Plik może zawierać komentarze (ignorowane przez kompilator) – te muszą być poprzedzone symbolem %. Inne znaki specjalne: \$ { } # _ & (w tekście poprzedzane symbolem \).

Składnia instrukcji:

```
\instrukcja[arg. opcjonalne]{argumenty}
```

Minimalistyczny przykład pliku źródłowego: dokument.tex

```
\documentclass{article}    % Początek preambuły
\usepackage{polski}
\usepackage[utf8]{inputenc}

\begin{document}           % Początek części głównej
  Prosty dokument tekstowy.
\end{document}
```

Preambuła i klasy dokumentu

Wybrane (standardowe) klasy L^AT_EX:

- `article` – artykuły naukowe i inne, raporty/sprawozdania, dokumentacje, opracowania;
- `book` – dłuższe książki z rozdziałami, podrozdziałami;
- `report` – raport z rozdziałami, praca dyplomowa;
- `letter` – listy;
- `beamer` – prezentacje (*slajdy*).

Każda z możliwych klas dokumentu ma zbiór predefiniowanych opcji, które można jednak zmieniać za pomocą opcjonalnego nawiasu kwadratowego w preambule, to znaczy:

```
\documentclass[arg. opcjonalne]{klasa}.
```

Możliwe opcje obejmują między innymi: rozmiar strony (np. `a4paper`, `a5paper`, `legalpaper`); rozmiar czcionki (domyślnie `10pt`); skład jedno lub dwu kolumnowy (`onecolumn`, `twocolumn`); druk jedno lub dwustronny (`oneside`, `twoside`). Opcje **oddzielamy przecinkami**.

Preambuła, pakiety i style

Dalsza część preambuły to **pakiety**, które uaktywniamy dla danego dokumentu poleceniem:

```
\usepackage[opcje]{pakiet}
```

Zadaniem pakietów jest rozszerzenie systemu o nowe możliwości – związane np. z definicją kodowania znaków, załączaniem rysunków, etc. Generalny wygląd dokumentu określa **styl**, który pozwala zdefiniować **nagłówek** oraz **stopkę**. Definicja stylu odbywa się podobnie jak poprzednio:

```
\pagestyle{styl}
```

lub

```
\thispagestyle{styl}
```

Standardowe style to: **plain** (domyślny, tj. pusty nagłówek i numer strony w stopce); **headings** (numer strony w nagłówku i pusta stopka); **empty** (nagłówek i stopka są puste).

Treść dokumentu

Znaki *białe* występujące w tekście (również wielokrotne) są interpretowane w procesie kompilacji jako pojedyncze *spacje*; akapity oddziela się pustym wierszem. Łamanie tekstu na wiersze wykonywane jest automatycznie – możemy jednak zdefiniować, które z odstępów (znaków białych) mają pozostać niezłamane zastępując je znakami *tyldy* (~).

Cudzysłów:

w języku polskim – „tekst” (dwa przecinki na początku i dwa apostrofy na końcu).

Treść dokumentu

Znaki *białe* występujące w tekście (również wielokrotne) są interpretowane w procesie kompilacji jako pojedyncze *spacje*; akapity oddziela się pustym wierszem. Łamanie tekstu na wiersze wykonywane jest automatycznie – możemy jednak zdefiniować, które z odstępów (znaków białych) mają pozostać niezłamane zastępując je znakami *tyldy* (~).

Cudzysłów:

w języku polskim – „tekst” (dwa przecinki na początku i dwa apostrofy na końcu).

Rodzaje *poziomych kresek*:

Treść dokumentu

Znaki *białe* występujące w tekście (również wielokrotne) są interpretowane w procesie kompilacji jako pojedyncze *spacje*; akapity oddziela się pustym wierszem. Łamanie tekstu na wiersze wykonywane jest automatycznie – możemy jednak zdefiniować, które z odstępów (znaków białych) mają pozostać niezłamane zastępując je znakami *tyldy* (~).

Cudzysłów:

w języku polskim – „tekst” (dwa przecinki na początku i dwa apostrofy na końcu).

Rodzaje *poziomych kresek*:

- łącznik/dywiz (-)

Treść dokumentu

Znaki *białe* występujące w tekście (również wielokrotne) są interpretowane w procesie kompilacji jako pojedyncze *spacje*; akapity oddziela się pustym wierszem. Łamanie tekstu na wiersze wykonywane jest automatycznie – możemy jednak zdefiniować, które z odstępów (znaków białych) mają pozostać niezłamane zastępując je znakami *tyldy* (~).

Cudzysłów:

w języku polskim – „tekst” (dwa przecinki na początku i dwa apostrofy na końcu).

Rodzaje *poziomych kresek*:

- łącznik/dywiz (-)
- minus (—)

Treść dokumentu

Znaki *białe* występujące w tekście (również wielokrotne) są interpretowane w procesie kompilacji jako pojedyncze *spacje*; akapity oddziela się pustym wierszem. Łamanie tekstu na wiersze wykonywane jest automatycznie – możemy jednak zdefiniować, które z odstępów (znaków białych) mają pozostać niezłamane zastępując je znakami *tyldy* (*~*).

Cudzysłów:

w języku polskim – „tekst” (dwa przecinki na początku i dwa apostrofy na końcu).

Rodzaje *poziomych kresek*:

- łącznik/dywiz (–)
- minus (—)
- półpauza (—)

Treść dokumentu

Znaki *białe* występujące w tekście (również wielokrotne) są interpretowane w procesie kompilacji jako pojedyncze *spacje*; akapity oddziela się pustym wierszem. Łamanie tekstu na wiersze wykonywane jest automatycznie – możemy jednak zdefiniować, które z odstępów (znaków białych) mają pozostać niezłamane zastępując je znakami *tyldy* (~).

Cudzysłów:

w języku polskim – „tekst” (dwa przecinki na początku i dwa apostrofy na końcu).

Rodzaje *poziomych kresek*:

- łącznik/dywiz (-)
- minus (—)
- półpauza (–)
- myślnik (—)

Treść dokumentu

Podział treści dokumentu w klasie `article`:

- `\section{Tytuł rozdziału}`
- `\subsection{Tytuł podrozdziału}`
- `\subsubsection{Tytuł (pod)podrozdziału}`

Można także użyć instrukcji `\part{Tytuł części}`, a w klasie `book` używa się ponadto instrukcji `\chapter{Tytuł rozdziału}`. Tytuły poszczególnych rozdziałów i podrozdziałów są podstawą do konstruowania spisu treści, który generuje się poleceniem `\tableofcontents`. Podobnie spisy rysunków i tabel – odpowiednio: `\listoffigures` i `\listoftables`.

Treść dokumentu

Podział treści dokumentu w klasie `article`:

- `\section{Tytuł rozdziału}`
- `\subsection{Tytuł podrozdziału}`
- `\subsubsection{Tytuł (pod)podrozdziału}`

Można także użyć instrukcji `\part{Tytuł części}`, a w klasie `book` używa się ponadto instrukcji `\chapter{Tytuł rozdziału}`. Tytuły poszczególnych rozdziałów i podrozdziałów są podstawą do skonstruowania spisu treści, który generuje się poleceniem `\tableofcontents`. Podobnie spisy rysunków i tabel – odpowiednio: `\listoffigures` i `\listoftables`.

Stronę tytułową można skonstruować korzystając z polecenia `\maketitle` poprzedzonego instrukcjami: `\title{Tytuł}`, `\author{Autor}`, `\date{data}`. Datę można sformułować manualnie lub skorzystać z instrukcji `\today`, czyli: `\date{\today}`.

Wyróżnienia i otoczenia

Wyróżnienie fragmentu tekstu możliwe jest np. poprzez użycie instrukcji `\emph{tekst}`. Wybrany fragment można także złożyć używając innej czcionki, np. `\texttt{czcionka maszynowa}` lub efektów: *kursywa* (`\textit{tekst}`), **pogrubienie** (`\textbf{tekst}`).

Otoczenia

Istotnym elementem formatującym treść są tzw. **otoczenia**. Są to fragmenty tekstu zamknięte w następującej konstrukcji:

```
\begin{nazwa}
```

Fragment tekstu, który może zawierać rysunki, tabele, etc.

```
\end{nazwa}
```

Przykład wyróżnienia z użyciem otoczenia

```
\begin{emph}
```

Tekst, który zostanie wyróżniony.

```
\end{emph}
```


Przykład listy i wyliczenia

Lista – kod źródłowy

Lista:

```
\begin{itemize}
\item element pierwszy,
\item element drugi,
\item element trzeci,
\item element czwarty.
\end{itemize}
```

Lista – efekt działania

Lista:

- element pierwszy,
- element drugi,
- element trzeci,
- element czwarty.

Wyliczenie – kod źródłowy

Lista:

```
\begin{enumerate}
\item element pierwszy,
\item element drugi.
\end{enumerate}
```

Wyliczenie – efekt działania

Lista:

- 1 element pierwszy,
- 2 element drugi.

Cytowania

Cytowania

Do cytowania fragmentów tekstu możemy użyć kursywy, ale służą do tego także instrukcje `\quote` lub `\quotation`.

Cytowanie fragmentu – kod źródłowy

Normalny tekst, pod którym
umieszczamy krótkie cytowanie:

```
\begin{quote}
```

Cytowany fragment tekstu.

```
\end{quote}
```

Cytowanie fragmentu – efekt działania

Normalny tekst, pod którym umieszczamy
krótkie cytowanie:

Cytowany fragment tekstu.

Cytowania

Cytowania

Do cytowania fragmentów tekstu możemy użyć kursywy, ale służą do tego także instrukcje `\quote` lub `\quotation`.

Cytowanie fragmentu – kod źródłowy

Normalny tekst, pod którym
umieszczamy krótkie cytowanie:

```
\begin{quote}  
  Cytowany fragment tekstu.  
\end{quote}
```

Cytowanie fragmentu – efekt działania

Normalny tekst, pod którym umieszczamy
krótkie cytowanie:

Cytowany fragment tekstu.

Wyrównywanie bloków tekstu jest możliwe za pomocą poleceń `\flushleft` oraz `\flushright` – także w formie *otoczenia*!

Przypisy

Do tworzenia przypisów używamy instrukcji `\footnote{Treść przypisu}`. Umieszczeniem i poprawną numeracją przypisów zajmuje się \LaTeX !

Przypis – kod źródłowy

To jest tekst, który zawiera przypis `\footnote{To jest treść przypisu.}`.

To jest tekst, który zawiera przypis¹.

¹To jest treść przypisu.

Tabele – otoczenie tabular

Do wstawiania tabel wykorzystujemy otoczenie tabular o następującej składni:

```
\begin{tabular}{parametry}  
...  
\end{tabular}
```

Możliwe parametry:

- l – wyrównaj zawartość komórki do lewej,
- r – wyrównaj zawartość komórki do prawej,
- c – wyrównaj zawartość komórki do środka,
- p{5cm} – ustaw szerokość kolumny na 5 cm (w przeciwnym razie szerokość jest automatyczna),
- | – oddziela kolumny pionową kreską.

Oddzielanie kreską kolejnych wierszy – instrukcja `\hline`; oddzielanie zawartości komórek – symbol `&`.

Tabele – otoczenie tabular

Przykład tabeli – kod źródłowy

```
\begin{tabular}{lcc}  
Kolumna 1 & Kolumna 2 & Kolumna 3 \\  
Kolumna 1 & Kolumna 2 & Kolumna 3 \\  
Kol 1 & 123 & asd\\  
a & b & c\\  
\end{tabular}
```

Tabele – otoczenie tabular

Przykład tabeli – kod źródłowy

```
\begin{tabular}{lcc}  
Kolumna 1 & Kolumna 2 & Kolumna 3 \\  
Kolumna 1 & Kolumna 2 & Kolumna 3 \\  
Kol 1 & 123 & asd\ \  
a & b & c\ \  
\end{tabular}
```

Przykład tabeli – efekt działania

Kolumna 1	Kolumna 2	Kolumna 3
Kolumna 1	Kolumna 2	Kolumna 3
Kol 1	123	asd
a	b	c

W ostatnim wierszu można pominąć znaki \\\.

Tabele – otoczenie tabular

Przykład tabeli – kod źródłowy

```
\begin{tabular}{|c|c|c|}  
  Kolumna 1 & Kolumna 2 & Kolumna 3 \\  
  Kolumna 1 & Kolumna 2 & Kolumna 3 \\  
  Kol 1 & 123 & asd\ \  
  a & b & c\ \  
\end{tabular}
```


Tabele – otoczenie tabular

Przykład tabeli – kod źródłowy

```
\begin{tabular}{|c|c|c|}  
Kolumna 1 & Kolumna 2 & Kolumna 3 \\  
Kolumna 1 & Kolumna 2 & Kolumna 3 \\  
Kol 1 & 123 & asd\\  
a & b & c\\  
\end{tabular}
```

Przykład tabeli – efekt działania

Kolumna 1	Kolumna 2	Kolumna 3
Kolumna 1	Kolumna 2	Kolumna 3
Kol 1	123	asd
a	b	c

Tabele – otoczenie tabular

Przykład tabeli – kod źródłowy

```
\begin{tabular}{|c|c|c|}  
  \hline  
  Kolumna 1 & Kolumna 2 & Kolumna 3 \\  
  Kolumna 1 & Kolumna 2 & Kolumna 3 \\  
  Kol 1 & 123 & asd\ \  
  a & b & c\ \  
  \hline  
\end{tabular}
```

Tabele – otoczenie tabular

Przykład tabeli – kod źródłowy

```
\begin{tabular}{|c|c|c|}  
  \hline  
  Kolumna 1 & Kolumna 2 & Kolumna 3 \\  
  Kolumna 1 & Kolumna 2 & Kolumna 3 \\  
  Kol 1 & 123 & asd\ \  
  a & b & c\ \  
  \hline  
\end{tabular}
```

Przykład tabeli – efekt działania

Kolumna 1	Kolumna 2	Kolumna 3
Kolumna 1	Kolumna 2	Kolumna 3
Kol 1	123	asd
a	b	c

Tabele – otoczenie tabular

Przykład tabeli – kod źródłowy

```
\begin{tabular}{|c|c|c|}  
  \hline  
  Kolumna 1 & Kolumna 2 & Kolumna 3 \\  
  \hline  
  Kolumna 1 & Kolumna 2 & Kolumna 3 \\  
  \hline  
  Kol 1 & 123 & asd\ \  
  \hline  
  a & b & c\ \  
  \hline  
\end{tabular}
```

Tabele – otoczenie tabular

Przykład tabeli – kod źródłowy

```
\begin{tabular}{|c|c|c|}  
 \hline  
 Kolumna 1 & Kolumna 2 & Kolumna 3 \\  
 \hline  
 Kolumna 1 & Kolumna 2 & Kolumna 3 \\  
 \hline  
 Kol 1 & 123 & asd \\  
 \hline  
 a & b & c \\  
 \hline  
 \end{tabular}
```

Przykład tabeli – efekt działania

Kolumna 1	Kolumna 2	Kolumna 3
Kolumna 1	Kolumna 2	Kolumna 3
Kol 1	123	asd
a	b	c

Tabele – otoczenie tabular i table

W tekście używamy zwykle otoczenia tabular w połączeniu z otoczeniem table.

Tabele – otoczenie tabular i table

W tekście używamy zwykle otoczenia tabular w połączeniu z otoczeniem table.

Składnia:

```
\begin{table}[b]
\caption{Podpis, który pojawi się pod tabelą.}
\label{tab1} % Nazwa służąca do cytowania tabeli
\begin{tabular}{|l|c|c|}
\hline
Kolumna 1 & Kolumna 2 & Kolumna 3 \\
\hline
Kolumna 1 & Kolumna 2 & Kolumna 3 \\
\hline
\end{tabular}
\end{table}
```

Umieszczenie tabeli: [h] (*here*), [t] (*top*), [b] (*bottom*), [p] (strona zawierająca wyłącznie *wstawki*), [!] (ignoruj kontrolę umieszczenia tabeli).

Tabele – otoczenie tabular i table

Przykład tabeli w otoczeniu table

Tabela 1: Przykładowa tabela zawierające przykładowe informacje

Kolumna 1	Kolumna 2	Kolumna 3
Kolumna 2	Kolumna 2	Kolumna 3

Tabele – otoczenie tabular i table

Przykład tabeli w otoczeniu table

Tabela 1: Przykładowa tabela zawierające przykładowe informacje

Kolumna 1	Kolumna 2	Kolumna 3
Kolumna 2	Kolumna 2	Kolumna 3

Jeżeli użyliśmy instrukcji `label`, to możemy zrobić tak (kod źródłowy):

Przykładowy fragment tekstu z odnośnikiem do Tabeli `\ref{tab1}`.

Tabele – otoczenie tabular i table

Przykład tabeli w otoczeniu table

Tabela 1: Przykładowa tabela zawierające przykładowe informacje

Kolumna 1	Kolumna 2	Kolumna 3
Kolumna 2	Kolumna 2	Kolumna 3

Jeżeli użyliśmy instrukcji `label`, to możemy zrobić tak (kod źródłowy):

Przykładowy fragment tekstu z odnośnikiem do Tabeli `\ref{tab1}`.

Efekt działania:

Przykładowy fragment tekstu z odnośnikiem do Tabeli 1.

Wstawianie rysunków

Wstawianie rysunków do tekstu za pomocą instrukcji:

```
\includegraphics[parametry]{ścieżka dostępu do pliku graficznego}
```

Wstawianie rysunków

Wstawianie rysunków do tekstu za pomocą instrukcji:

```
\includegraphics[parametry]{ścieżka dostępu do pliku graficznego}
```

Wybrane parametry:

- `width`=wartość z jednostką (przykładowe jednostki: px, cm, pt i inne),
- `height`=wartość z jednostką,
- `scale`=skala (przeskalowanie według podanej wartości),
- `angle`=kąt (obrot o kąt – ujemny lub dodatni).

Przydatna instrukcja: `\textwidth` (`\textheight`). **Przykład:** `[width=0.5\textwidth]`.

Wstawianie rysunków

Wstawianie rysunków do tekstu za pomocą instrukcji:

```
\includegraphics[parametry]{ścieżka dostępu do pliku graficznego}
```

Wybrane parametry:

- `width`=wartość z jednostką (przykładowe jednostki: px, cm, pt i inne),
- `height`=wartość z jednostką,
- `scale`=skala (przeskalowanie według podanej wartości),
- `angle`=kąt (obrot o kąt – ujemny lub dodatni).

Przydatna instrukcja: `\textwidth` (`\textheight`). **Przykład:** `[width=0.5\textwidth]`.

Ścieżka dostępu do pliku graficznego może być względna lub bezwzględna.

Wstawianie rysunków – otoczenie figure

Przykład użycia otoczenia figure:

```
\begin{figure}  
\caption{Podpis, który pojawi się pod rysunkiem.}  
\label{rys1} % Nazwa służąca do cytowania rysunku  
\includegraphics[width=0.5\textwidth]{/home/herman/rysunki/rysunek1.png}  
\end{figure}
```

Cytowanie rysunku wykonuje się podobnie jak w przypadku tabel – za pomocą instrukcji `\ref{label}`.

Wyrażenia matematyczne

Do zapisu wyrażeń matematycznych możemy posłużyć się m.in. otoczeniem `math`, `displaymath`, `equation` lub zapisać wyrażenie bezpośrednio pomiędzy dwoma symbolami `$`.

Wyrażenia matematyczne

Do zapisu wyrażeń matematycznych możemy posłużyć się m.in. otoczeniem `math`, `displaymath`, `equation` lub zapisać wyrażenie bezpośrednio pomiędzy dwoma symbolami `$`.

$$\text{\textcolor{red}{\$}} y(x)=ax^2+bx+c \text{\textcolor{red}{\$}}$$

$$y(x) = ax^2 + bx + c$$

Wyrażenia matematyczne

Do zapisu wyrażen matematycznych możemy posłużyć się m.in. otoczeniem `math`, `displaymath`, `equation` lub zapisać wyrażenie bezpośrednio pomiędzy dwoma symbolami `$`.

`$ y(x)=ax^2+bx+c $`

$$y(x) = ax^2 + bx + c$$

```
\begin{equation}
y(x)=ax^2+bx+c
\end{equation}
```

$$y(x) = ax^2 + bx + c \quad (1)$$

Wyrażenia matematyczne

Do zapisu wyrażeń matematycznych możemy posłużyć się m.in. otoczeniem `math`, `displaymath`, `equation` lub zapisać wyrażenie bezpośrednio pomiędzy dwoma symbolami `$`.

`$ y(x)=ax^2+bx+c $`

$$y(x) = ax^2 + bx + c$$

```
\begin{equation}  
y(x)=ax^2+bx+c  
\end{equation}
```

$$y(x) = ax^2 + bx + c \quad (1)$$

W równaniach zapisanych w otoczeniu `equation` można, podobnie jak w przypadku rysunków i tabel, wykorzystać instrukcję `label` w celu późniejszego zacytowania.

Wyrażenia matematyczne – przykłady

- Potęga (indeks górny) – x^2 ,
- indeks dolny – x_2 ,
- dodawanie/odejmowanie – $x \pm y$,
- mnożenie – $x \cdot y$, $x \times y$,
- ułamek – $\frac{a}{b}$,
- pierwiastek – $\sqrt[3]{2}$,
- wektor – \vec{a} ,
- litery greckie – α , β , ...
- granica – $\lim_{x \rightarrow 0}$,
- całka – \int_0^{π} ,

- suma – $\sum_{i=1}^n$,
- iloczyn – \prod_{ϵ} ,
- większe lub równe – $x \geq y$,
- mniejsze lub równe – $x \leq y$,
- a nie jest równe b – $a \neq b$,
- plus/minus – $x \pm y$,
- *daszek* – \hat{a} ,
- proporcjonalność – $x \propto y$,
- strzałka w prawo – $x \rightarrow y$,
- (...).

Bibliografia

Do tworzenia bibliografii służy instrukcja thebibliography o następującej składni:

```
\begin{thebibliography}  
  \bibitem{JK} J. Kowalski, Tytuł, Czasopismo, Numer, Rok.  
  \bibitem{AN} A. Nowak, Tytuł, Czasopismo, Numer, Rok.  
\end{thebibliography}
```

Bibliografia

Do tworzenia bibliografii służy instrukcja `thebibliography` o następującej składni:

```
\begin{thebibliography}  
  \bibitem{JK} J. Kowalski, Tytuł, Czasopismo, Numer, Rok.  
  \bibitem{AN} A. Nowak, Tytuł, Czasopismo, Numer, Rok.  
\end{thebibliography}
```

Następnie w kodzie źródłowym możemy napisać tak:

Tekst z cytowaniem do artykułu J. Kowalskiego `\cite{JK}` oraz A. Nowaka `\cite{AN}`.

Bibliografia

Do tworzenia bibliografii służy instrukcja `thebibliography` o następującej składni:

```
\begin{thebibliography}  
  \bibitem{JK} J. Kowalski, Tytuł, Czasopismo, Numer, Rok.  
  \bibitem{AN} A. Nowak, Tytuł, Czasopismo, Numer, Rok.  
\end{thebibliography}
```

Następnie w kodzie źródłowym możemy napisać tak:

Tekst z cytowaniem do artykułu J. Kowalskiego `\cite{JK}` oraz A. Nowaka `\cite{AN}`.

Efektem po skompilowaniu jest:

Tekst z cytowaniem do artykułu J. Kowalskiego [1] oraz A. Nowaka [2].

Bibliografia

Do tworzenia bibliografii służy instrukcja `thebibliography` o następującej składni:

```
\begin{thebibliography}  
  \bibitem{JK} J. Kowalski, Tytuł, Czasopismo, Numer, Rok.  
  \bibitem{AN} A. Nowak, Tytuł, Czasopismo, Numer, Rok.  
\end{thebibliography}
```

Następnie w kodzie źródłowym możemy napisać tak:

Tekst z cytowaniem do artykułu J. Kowalskiego `\cite{JK}` oraz A. Nowaka `\cite{AN}`.

Efektem po skompilowaniu jest:

Tekst z cytowaniem do artykułu J. Kowalskiego [1] oraz A. Nowaka [2].

Ta metoda jest wygodna w krótszych dokumentach, artykułach, etc. W obszerniejszych pracach używa się częściej pakietu BibTeX – o nim wkrótce.

Tworzenie prezentacji – klasa beamer

```
\documentclass[Red, mathserif, xcolor=pdftex, table, aspectratio=169]{beamer}
\usetheme{Boadilla}
\usecolortheme{seahorse}
\usepackage{polski}
\usepackage[utf8]{inputenc}

\title[Podstawy informatyki / wykład nr~04]{Podstawy informatyki\\vspace{0.4cm}
\small{WYKŁAD nr~04}\\vspace{0.2cm}\small{Fizyka Techniczna, WFT PP}}

\author[M.~Hermanowicz (IF PP)]{Michał Hermanowicz}
\institute[{Zakład Fizyki Obliczeniowej, Instytut Fizyki, Politechnika Poznańska}]
\date{Rok akademicki 2017/2018}

\begin{document}
\section{Wprowadzenie}
\frame{
  \frametitle{Wprowadzenie}
  (...)
}
\end{document}
```


Przykładowe dokumenty:

- najprostszy plik L^AT_EX,
- artykuł/sprawozdanie,
- prezentacja,
- praca dyplomowa.

Czas na pytania i dyskusję