

Wstęp

Statystyczny użytkownik myszki komputerowej żyje w przekonaniu, iż stworzenie dokumentu tekstowego wysokiej jakości jest stosunkowo proste. Jest prawdopodobne, że przekonanie to zostało mu zaszczerpione przez edytory typu WYSIWIG (What You See is What You Get). Klasycznym przykładem takiego programu jest MS Word. Umożliwia on m. in. modyfikację kroju czcionki oraz rozmiaru marginesów w taki sposób aby dokument był przyjemny dla oka. Czy wizualna atrakcyjność dokumentu oznacza jednak, że dokument jest wysokiej jakości? Ciężko powiedzieć... nie napisaliśmy jeszcze co to znaczy “dokument wysokiej jakości”.

Otóż, dokument powinien być tak sformatowany, aby umożliwić czytelnikowi całkowite skupienie się na tekście. Nie powinien zawierać elementów, które rozpraszają lub odwracają jego uwagę od tego co jest najważniejsze – od treści. Aby zrealizować ten cel należy wybrać: krój czcionki (szeryfowa, bezszeryfowa), ustalić liczbę wyrazów w linii (50, 66 a może 80?), dokonać podziału wyrazów, które nie mieszczą się w linii, ustalić położenie wykresów i tabel na kartce papieru (górze, dół) oraz wiele innych. Nie jest to proste, zwłaszcza jeśli zmienimy coś w gotowym tekście wymuszając przesunięcie rysunków i tabel na kartce.

Aby zachować wysoki poziom zasad typograficznych i jednocześnie ułatwić autorowi skupienie się na treści dokumentu stworzono program **LaTeX** (wł. zestaw makr i instrukcji do obsługi programu **TeX**). Nazwę **LaTeX** czytamy **la-tech**. Program rozdziela funkcję autora od zecera (osoby dokonującej składu tekstu) i zwalnia nas z odpowiedzialności za prawidłową typografię.

Pisanie w **LaTeXu** w pewnym stopniu przypomina pisanie w języku HTML. “Program” składa się z właściwego tekstu i zestawu instrukcji, które mówią o tym jak nasz tekst ma wyglądać po “kompilacji”. Dzięki ogromnej liczbie pakietów i szablonów dostępnych w systemie **LaTeX** można tworzyć wiele różnych typów dokumentów. Na dzisiejszych zajęciach skorzystamy z podstawowego typu **article**.

W celu poszerzenia wiedzy, autorzy polecają pozycję: *Nie za krótkie wprowadzenie do systemu LaTeX_{2 ϵ}* .

Pierwszy dokument

Przejdziemy teraz do stworzenia pierwszego dokumentu w **LaTeXu**. Najpierw musimy zadeklarować klasę dokumentu. Robimy to za pomocą polecenia

`\documentclass[opcje]{typ}`, które umożliwia nam wybranie typu dokumentu (`article`, `report`, `book`, `letter`) oraz dodatkowych opcji, np.

- rozmiaru pisma (`10pt`, `11pt`, itp.),
- rozmiaru papieru (`a4paper`, `letterpaper`),
- czy strona tytułowa ma być wygenerowana osobno (`titlepage`, `notitlepage`),
- liczby kolumn (`onecolumn`, `twocolumn`),
- czy tekst ma być przygotowany do wydruku dwustronnego (`oneside`, `twoside`), itd.

Opcje ujęte w nawiasy kwadratowe (`[]`) nie są obligatoryjne – jeśli nie zdefiniujemy żadnej, zostaną użyte wartości domyślne.

W edytorze tekstu wpisz poniższy kod:

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

\begin{document}
  Stół z powyżamywanymi nogami!
\end{document}
```

zapisz go w pliku z rozszerzeniem `.tex`, np. `dokument.tex` i skompiluj za pomocą polecenia

```
pdflatex dokument.tex
```

Kompilator utworzy kilka plików, m. in. plik `dokument.pdf`, który zawiera nasz tekst.

Otwórz plik `dokument.pdf` i naciesz oczy wynikiem swojej pracy. Ale co to? Okazuje się, że brakuje kilku liter! Podstawowe możliwości **LaTeXa** nie sięgają bowiem języka polskiego.

Pakiety

Domyślne reguły można (i często trzeba) rozszerzyć wykorzystując dodatkowe pakiety. Przykładowo, kompletną obsługę języka polskiego włączamy dodając pakiety:

- `\usepackage[T1]{fontenc}` - zawiera czcionkę z polskimi znakami,

- `\usepackage[utf8]{inputenc}` - definiuje kodowanie (utf8 - Linux, Cp1250 - Windows) znaków w pliku źródłowym,
- `\usepackage[english, polish]{babel}` - definiuje język dokumentu,
- `\usepackage{polski}` - wymusza polskie reguły składu dokumentu.

Pakiety, z których planujemy korzystać umieszczamy w t.zw. *preamble*: za instrukcją `\documentclass`, a przed instrukcją `\begin{document}`. Zmodyfikuj teraz kod źródłowy tak, aby polskie znaki wydrukowały się prawidłowo.

Strona tytułowa

Ponieważ tekst wygląda teraz elegancko, warto pochwalić się swoim osiągnięciem świata. Włącz opcję utworzenia osobnej strony tytułowej i dodaj (za instrukcjami dołączającymi pakiety) instrukcje “sterujące” jej zawartością, np.:

```
\author{Gall Anonim}
\title{Kroniki}
\date{1113-1116}
```

W “ciele” dokumentu (między instrukcjami `\begin{document}` i `\end{document}`) umieść instrukcję

```
\maketitle
```

i skompiluj ponownie tekst. Sprawdź co się stanie jeśli opcja osobnej strony tytułowej nie będzie włączona.

Podział logiczny dokumentu

Sekcje

Rzadko mamy do czynienia z tekstami pozbawionymi struktury logicznej. Przemysłana kompozycja dokumentu pozwala uporządkować treść i ułatwia jej zapamiętanie. W LaTeXu podstawowy podział dokumentu można otrzymać wykorzystując instrukcje:

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

Podziel wybrany fragment tekstu za pomocą wymienionych poleceń i sprawdź jak działają. Zbuduj teraz spis treści odpowiadający utworzonym rozdziałom. W tym celu dodaj instrukcję

```
\tableofcontents
```

na początku ciała dokumentu.

Listy i numeracje

Wypunktowania i numeracje uzyskuje się wykorzystując tzw. otoczenia. Charakteryzują się one dwiema instrukcjami: otwierającą `\begin{}` i zamykającą `\end{}`. W przypadku listy nieuporządkowanej napiszemy:

```
\begin{itemize}
  \item A
  \item B
\end{itemize}
```

Z kolei dla listy uporządkowanej

```
\begin{enumerate}
  \item A
  \item B
\end{enumerate}
```

Utwórz teraz zagnieżdżoną listę:

1. A1
 2. B1
- A2

- B2
 - A3
 - B3
 - C2
3. C1
 1. D2
 2. E2
 4. D1

Wzory i równania matematyczne

Dobry tekst naukowy nie może obyć się bez równań i wzorów (choć znaleźli by się i tacy, którzy twierdzą inaczej). W **LaTeXu** istnieje kilka sposobów dodawania wyrażeń matematycznych. Podstawowy z nich umożliwia wstawianie wyrażeń w tej samej linii co tekst (z ang. często mówi się *inline*)

wzór $E=mc^2$ powstał \ldots

Kolejny sposób to wykorzystanie otoczenia, które pozwala na wyświetlenie równania w osobnym wierszu, np.

dla trójkąta prostokątnego zachodzi:

```
\[
  c ^ 2 = a ^ 2 + b ^ 2
\]
```

Dla utrzymania porządku często warto numerować równania

```
\begin{equation}
  c ^ 2 = a ^ 2 + b ^ 2
\end{equation}
```

Aby sprawnie posługiwać się jakimś językiem trzeba znać jego składnię. Przedstawimy teraz podstawowe symbole i techniki wykorzystywane w **LaTeXu** przy wpisywaniu równań matematycznych.

Greckie litery są jednym z podstawowych elementów. Są dość łatwe do zapamiętania ze względu na zapis “fonetyczny”. Małe litery uzyskamy stosując:

```
\[
  \alpha, \beta, \gamma, \delta, \epsilon, \varepsilon, \zeta, \eta, \theta,
```

Duże, natomiast:

```
\[
  \Gamma, \Delta, \Theta, \Lambda, \Pi, \Sigma, \Upsilon, \Phi, \Psi, \Omega,
```

Indeksy górne i dolne otrzymuje się przez użycie symboli \wedge i $_$:

```
\[ n^3=m^{a^2-d} \]
\[ \alpha_{1,2}=x_1+y_{1,2} \]
```

Zapis pierwiastków wymaga z kolei instrukcji `\sqrt{}`

```
\[ R = \sqrt{x ^ 2 + y ^ 2 + z ^ 2} \]
```

Symbol wektora uzyskuje się przez

```
\[ \vec{z} = \vec{x} \times \vec{y} \]
```

Podkreślenia i klamry wykorzystywane do komentowania poszczególnych części wzorów:

```
\[ \overline{x + i \cdot y} \cdot \underline{v - i \cdot w} \]
\[ \underbrace{M_{\mu \nu} + 1}_{\neq 0} \rightarrow \overbrace{N_{\nu \mu}}^{\neq 0}
```

Nazwy funkcji powinny być zapisane zwykłą czcionką:

```
\[ 1 = \sin^2(x) + \cos^2(x) \]
```

a nie pochyloną:

```
\[ 1 = \sin^2(x) + \cos^2(x) \]
```

Ułamki piętrowe uzyskujemy przez

```
\[ \frac{x ^ 3 + \sqrt{x^2}}{x ^ 2 + x - 1} \]
```

Nawiasy możemy wprowadzić wprost

```
\[ y=((\frac{1}{1 + x})^2 - 1) \]
```

Nie jest to jednak elegancki sposób. W przypadku równań, które wymagają nawiasów różnej wysokości lepiej jest napisać

```
\[ y=\left( \left( \frac{1}{1 + x} \right)^2 - 1 \right) \]
```

Całki zapisujemy:

```
\[ \Gamma(z)=\int_0^{\infty}x^{z-1} e^{-x} dx \]
```

Natomiast sumy:

```
\[ S=\sum_{n=1}^{\infty} a \cdot q^{n-1} \]
```

Tablice (macierze, wyznaczniki itd...) zapisujemy stosując otoczenie `array`.

```
\[
\left|
\begin{array}{ccc}
a_{11} & a_{12} & a_{13} \\
a_{21} & a_{22} & a_{23} \\
a_{31} & a_{32} & a_{33}
\end{array}
\right|
\]
```

Parametry `{ccc}` oznaczają, że utworzona tablica będzie miała trzy kolumny, z których każda będzie zawierała tekst wyrównany do środka. Dwie pozostałe możliwości to `r` – wyrównanie do prawej i `l` – wyrównanie do lewej.

Otoczenie te można także wykorzystać do zapisu warunków

```
\[
\delta(x)=\left\{
\begin{array}{lr}
0 & x \neq 0 \\
+\infty & x = 0
\end{array}
\right.
\]
```

Ćwiczenia

Zapisz poniższe równania wykorzystując system **LaTeX**. Niektóre symbole, których należy użyć, nie były wprowadzone wcześniej. W sieci Internet można znaleźć listy dostępnych symboli. W naszej pracowni można skorzystać ze strony: https://www.sharelatex.com/learn/Mathematical_expressions

Należy pamiętać, że symbole rzadsze są dostępne dopiero po dołączeniu odpowiednich pakietów. Przykładowo, symbol całki podwójnej `\iint` uzyskamy po dołączeniu pakietu `amsmath`.

$$i\hbar \frac{\partial}{\partial t} \Psi(\vec{r}, t) = -\frac{\hbar^2}{2m} \nabla^2 \Psi(\vec{r}, t) + V(\vec{r}) \Psi(\vec{r}, t)$$

$$\oint_{\partial \Sigma} \mathbf{B} \cdot d\mathbf{l} = \mu_0 \iint_{\Sigma} \mathbf{J} \cdot d\mathbf{S} + \mu_0 \epsilon_0 \frac{d}{dt} \iint_{\Sigma} \mathbf{E} \cdot d\mathbf{S}$$

$$\begin{aligned} \text{Var} \left(\hat{Z}(x_0) - Z(x_0) \right) &= \\ &= \underbrace{c(x_0, x_0)}_{\text{Var}(Z(x_0))} - \underbrace{\begin{pmatrix} c(x_1, x_0) \\ \vdots \\ c(x_n, x_0) \end{pmatrix} \begin{pmatrix} c(x_1, x_1) & \cdots & c(x_1, x_n) \\ \vdots & \ddots & \vdots \\ c(x_n, x_1) & \cdots & c(x_n, x_n) \end{pmatrix}^{-1} \begin{pmatrix} c(x_1, x_0) \\ \vdots \\ c(x_n, x_0) \end{pmatrix}}_{\text{Var}(\hat{Z}(x_0))} \end{aligned}$$

Tabele

Istnieje wiele bibliotek rozszerzających możliwości tworzenia tabel w systemie **LaTeX**. Pokażemy najprostszy przykład z wykorzystaniem otoczenia `tabular`. Otoczenie

to tworzy tabelę w sposób automatyczny, tzn. szerokości i wysokości poszczególnych komórek są dobierane do ich zawartości. Jedyne parametry na jakie mamy wpływ to liczba kolumn i sposób ich justowania. Przeanalizuj poniższy kod:

```
\begin{tabular}{l|c|c|c}
& symb. & symb. & symb. \\ \hline \hline
symb. & X & 0 & 0 \\ \hline
symb. & 0 & X & X \\ \hline
symb. & X & 0 & X \\ \hline
\end{tabular}
```

Jaki wpływ na wygląd tabeli ma parametr `{l|c|c|c}`?

Otoczenie `tabular` pozwala także na tworzenie wielokolumnowych rubryk. Służy do tego polecenie `\multicolumn{1_kolumn}{justowanie}{tekst}`.

```
\begin{tabular}{c|c|c}
\multicolumn{3}{c}{symb.} \\ \hline \hline
X & 0 & 0 \\ \hline
0 & X & X \\ \hline
X & 0 & X \\ \hline
\end{tabular}
```

Użycie powyższej konstrukcji ma pewną wadę. Tabele dodane do dokumentu za pomocą otoczenia `tabular` nie zawsze znajdują się tam gdzie byśmy tego chcieli. Aby mieć kontrolę nad położeniem tabeli należy użyć otoczenia `table`, które ma jeden argument opcjonalny mówiący o tym gdzie na stronie ma się znaleźć wstawka z tabelą. Dodatkowo, otoczenie to pozwala na dodanie podpisu do naszej tabeli. Służy do tego instrukcja `\caption{podpis}`. Przykładowo, jeśli chcemy aby tabela znalazła się dokładnie w miejscu użycia, napiszemy:

```
\begin{table}[h]
\begin{tabular}{c|c|c}
X & 0 & 0 \\ \hline
0 & X & X \\ \hline
X & 0 & X \\ \hline
\end{tabular}
\caption{Kółko i krzyżyk.}
\end{table}
```

Pozostałe parametry to:

- `t` – umieść wstawkę na górze strony,
- `b` – umieść wstawkę na dole strony,
- `p` – umieść wstawkę na oddzielnej stronie ze wstawkami.

Ćwiczenia

Wykonaj poniższą tabelę:

Tabela			
Lewa kol.		Prawa kol.	
	Środkowa kol.		
kom	kom	kom	kom
kom	kom	kom	kom
kom	kom	kom	kom

Figure 1:

Grafika

Wstawienie grafiki w *LaTeXu* ogranicza się do dołączenia pakietu `\usepackage{graphicx}` oraz użycia instrukcji `\includegraphics[parametry]{ścieżka_do_grafiki}`. Przykładowo, jeśli chcemy wstawić obrazek, którego szerokość będzie równa 95% szerokości tekstu napiszemy:

```
\includegraphics[width=.95\textwidth]{rysunek.pdf}
```

Podobnie jak to było w przypadku tabel, kontrolę nad położeniem rysunku i możliwość dodawania podpisów otrzymamy po zastosowaniu specjalnego otoczenia. W przypadku grafiki będzie to otoczenie `figure`:

```
\begin{figure}[!h]
  \includegraphics[width=.95\textwidth]{rysunek.pdf}
  \caption{Rysunek.}
\end{figure}
```

Odwołania

System LaTeX umożliwia wstawienie odwołania do niemal każdego obiektu posiadającego numer. Oznacza to, że możemy odwołać się do dowolnego rozdziału, równania, tabeli czy rysunku. Obiekt, do którego planujemy się odwołać oznaczamy za pomocą polecenia `\label{nazwa_obiektu}`. Natomiast w miejscu, w którym odwołujemy się, wstawiamy polecenie `\ref{nazwa_obiektu}`. Zamiast numeru obiektu, możemy podać stronę, na której znajduje się dany element. W tym celu stosujemy polecenie `\pageref{nazwa_obiektu}`. Stosując indywidualne nazwy obiektów, nie musimy martwić się o ewentualne zmiany numeracji wynikające z pracy nad tekstem. LaTeX zajmie się koniecznymi modyfikacjami.

Przykładowo, aby odwołać się do rozdziału piszemy:

```
\section{Rodzaje sznurówek}\label{sznu}
W rozdziale \ref{sznu} opiszemy problem doboru sznurówek.
```

W celu odwołania się do tabeli:

```
\begin{table}[h]
  \begin{tabular}{c|c|c}
    X & 0 & 0 \\ \hline
    0 & X & X \\ \hline
    X & 0 & X \\
  \end{tabular}
  \caption{Kółko i krzyżyk.}
  \label{kik}
\end{table}
```

Tabela `\ref{kik}` zawiera przykładową partię w kółko i krzyżyk. Znajduje się ona na stronie `\pageref{kik}`.

Ćwiczenia

Stwórz odwołania do pozostałych obiektów użytych w trakcie ćwiczeń (równań i rysunków).