

§2. System składu tekstu L^AT_EX

Robert Janczewski

Gdańsk, 2025

Cechy systemu

\LaTeX to język opisu dokumentów tekstowych, umożliwiający tworzenie wysokiej jakości dokumentów zawierających tekst, formuły matematyczne i grafikę.

Opisując dokument przy pomocy \LaTeX -a określamy jedynie jego zawartość i logiczną strukturę, a ustalenie szczegółów wyglądu zostawiamy \LaTeX -owi.

\LaTeX jest de facto zbiorem makr dla systemu \TeX , który zajmuje się właściwym składem dokumentu.

\TeX realizuje te elementy składu, które można zautomatyzować: podział tekstu na strony, układanie tekstu i grafik na stronie, numerowanie i odwołania do numerów, tworzenie spisów treści, bibliografii, indeksu itd.

Aby skorzystać z \LaTeX -a, niezbędny jest edytor tekstu i jedna z dystrybucji \LaTeX -a.

Lista edytorów wspierających pracę z dokumentami \LaTeX -owymi jest długa, podobnie jak lista dystrybucji \LaTeX -a.

Dla przykładu, w systemie Windows działa MikTeX (<https://miktex.org/>) i TeXnicCenter (<http://www.texniccenter.org/>), a pod Linuksem TeXLive (<https://www.tug.org/texlive/>) i kwrite (<https://www.kde.org/applications/utilities/kwrite/>).

Istnieją także rozwiązania nie wymagające instalacji, ale wymagające dostępu do internetu, np. serwis <https://www.overleaf.com/>.

Praca z L^AT_EX-em

Pierwszym etapem pracy z L^AT_EX-em jest przygotowanie pliku źródłowego, po nim następuje kompilacja (jedno- lub wielokrotna), dająca w wyniku docelowy dokument i rozmaite pliki pomocnicze.

L^AT_EX korzysta m.in. z plików o następujących rozszerzeniach:

- 1 tex; plik źródłowy;
- 2 cls; pliki źródłowe klas dokumentów;
- 3 sty; pliki źródłowe pakietów makr;
- 4 dvi/pdf; docelowe dokumenty;
- 5 toc/lof/lot; spis treści, rysunków i tabel;
- 6 aux; plik wykorzystywany przy wielokrotnej kompilacji;
- 7 log; zapis przebiegu kompilacji.

Efektom kompilacji jest także wyświetlany na ekranie raport, zawierający informacje o błędach składni, ostrzeżeniach i błędach składu.

Ostrzeżenia pojawiają się wtedy, gdy pewne czcionki są niedostępne, gdy korzystamy z przestarzałych pakietów, gdy dokonano pewnych (niekoniecznie pożądanych przez autora) korekt w dokumencie itd.

Błędy składu pojawiają się wtedy, gdy algorytmom stosowanym przez T_EX-a nie udało się tak ułożyć tekstu, by dobrze wyglądał.

Dostajemy wówczas najlepiej według T_EX-a ułożony tekst, ale musimy go zmodyfikować, jeśli chcemy by był dobrze złożony.

Dobrym zwyczajem jest takie przygotowanie finalnej wersji dokumentu, żeby przy jego kompilacji nie było ostrzeżeń ani błędów składu.

Schemat pliku źródłowego

Typowy plik *.tex składa się z następujących fragmentów:

- 1 preambuły; preambuła musi określać klasę dokumentu i używane przez dokument pakiety, można tam także umieścić rozmaite polecenia, np. definiujące nowe makra i zmieniające wygląd dokumentu;
- 2 dokumentu właściwego; zawiera on treść dokumentu docelowego oraz polecenia określające logiczną strukturę dokumentu.

To, co znajduje się w preambule wpływa na wygląd dokumentu, ale nie generuje żadnej treści—ta musi znajdować się w dokumencie właściwym.

Wszystko, co znajduje się po dokumencie właściwym jest ignorowane.

Przykład

Poniżej znajduje się fragment preambuły i prawie pusta właściwa część dokumentu.

```
\documentclass[polish,10pt]{beamer}
  \usepackage[T1]{fontenc}
  \usepackage{polski}
  \usepackage{babel}
  \newcommand{\highlight}[1]{\alert{\emph{#1}}}
\begin{document}
  To jest \highlight{T E S T}.
\end{document}
```

Istnieje wiele gotowych klas dokumentów przygotowanych z myślą o różnych typach dokumentów, np. book i amsbook dla książek, article i amsart dla artykułów oraz beamer dla prezentacji.

Tryby pracy L^AT_EX-a

W trakcie pracy nad dokumentem L^AT_EX może znajdować się w jednym z poniższych trybów:

- 1 matematycznym; w tym trybie przetwarzane są formuły matematyczne, obowiązują wówczas zupełnie inne reguły składu: spacje są nieznaczące, każdy znak ma przypisaną interpretację (zmienna, operator, relacja itd.), dostępne są rozmaite polecenia specjalne (greckie litery, specjalne symbole itd.);
- 2 tekstowym pionowym; w tym trybie składane są akapity, które następnie są dzielone na strony;
- 3 tekstowym poziomym; w tym trybie składany jest pojedynczy akapit, który jest dzielony na zdania, zdania na wyrazy a wyrazy na litery i ligatury.

Tryb matematyczny trzeba włączyć ręcznie, pozostałe tryby uruchamiane są automatycznie.

Tryb tekstowy

Akapity rozdzielamy pustą linią lub poleceniem `\par`. Podział na linie i strony następuje automatycznie, ale można go wymusić—służą do tego polecenia `\newline` i `\newpage`.

Dokument to zwykły tekst, w którym znajdują się polecenia (makra). Istnieją dwa typy poleceń:

- 1 otoczenia postaci `\begin{nazwa} ... \end{nazwa}`; otoczenie oddziałuje wyłącznie na znajdujący się wewnątrz niego tekst;
- 2 zwykłe polecenia postaci `\nazwa`; wywołany przez polecenie efekt obowiązuje od tego momentu do końca tekstu.

Istnieją także jednoznakowe polecenia specjalne, ich omówienie znajduje się dalej.

Tryb matematyczny

Formuły otaczamy znakiem `$`, powodując włączenie trybu matematycznego. Ten tryb włączany jest także przez rozmaite polecenia takie jak:

- 1 `$$... $$` lub `\begin{displaymath} ... \end{displaymath}` lub `\begin{equation*} ... \end{equation*}`; formuła zostanie złożona w postaci wyśrodkowanej, w osobnym akapicie;
- 2 `\begin{equation} ... \end{equation}`; formule zostanie nadany numer i zostanie ona złożona w postaci wyśrodkowanej, w osobnym akapicie;
- 3 `\begin{eqnarray} ... \end{eqnarray}`; powstanie wielowierszowa formuła numerowana (z gwiazdką na końcu—nienumerowana).

Tych poleceń nie można zagnieżdżać; `\ensuremath` wymusza użycie trybu matematycznego.

Przykłady

Źródło	Dokument docelowy
$\{x \leq n: 2 x\}$	$\{x \leq n: 2 x\}$
$f: X \rightarrow Y$	$f: X \rightarrow Y$
$\mathbb{R}_+ \ni x \mapsto f(x) = \lg x \in \mathbb{R}$	$\mathbb{R}_+ \ni x \mapsto f(x) = \lg x \in \mathbb{R}$
$G=(V, E)$ jest grafem	$G=(V, E)$ jest grafem
$\Delta = \Theta(n)$	$\Delta = \Theta(n)$
$A=[a_{ij}^2]_{i,j=1, \dots, n}$	$A=[a_{ij}^2]_{i,j=1, \dots, n}$
$\frac{1}{2}\sqrt{5}$	$\frac{1}{2}\sqrt{5}$
a', a''	a', a''

Przykłady

Źródło	Dokument docelowy
<pre>\$a=\begin{cases}</pre> <pre>1\text{,}&\text{gdym}b=0\\</pre> <pre>2\text{,}&\text{gdym}b\neq 0</pre> <pre>\end{cases}\$</pre>	$a = \begin{cases} 1, & \text{gdym} b = 0 \\ 2, & \text{gdym} b \neq 0 \end{cases}$
<pre> \$\displaystyle\left(</pre> <pre> \frac{\sum_{i=1}^na_i}{n}</pre> <pre> \right)^n\geqslant</pre> <pre> \prod_{i=1}^na_i\$</pre>	$\left(\frac{\sum_{i=1}^na_i}{n}\right)^n \geqslant \prod_{i=1}^na_i$

Znaki specjalne

Niektóre znaki mają specjalne znaczenie:

- 1 `#`; używane w makrach do oznaczania argumentów, np. `#1` to pierwszy argument makra;
- 2 `%`; komentarz, wszystko od tego znaku do końca linii jest ignorowane;
- 3 `$`; polecenie przejścia do trybu matematycznego;
- 4 `&`; separator pól w tabelach;
- 5 `~`; niełamliwa spacja;
- 6 `_` i `^`; indeks dolny i górny (działa wyłącznie w trybie matematycznym);
- 7 `\`; znak rozpoczynający polecenia;
- 8 `{` i `}`; znaki otaczające parametry makr lub fragmenty tekstu;
- 9 `[` i `]`; znaki otaczające opcjonalne parametry makr.

Znaki specjalne

Sposoby uzyskania w tekście znaków specjalnych przedstawia poniższa tabela.

Źródło	Dokument docelowy
<code>\#</code>	<code>#</code>
<code>\%</code>	<code>%</code>
<code>\\$</code>	<code>\$</code>
<code>\&</code>	<code>&</code>
<code>\$\sim\$</code>	<code>~</code>
<code>_</code>	<code>_</code>
<code>\verb.^.</code>	<code>^</code>
<code>\textbackslash</code>	<code>\</code>
<code>\$\{\$</code>	<code>{</code>
<code>\$\}\$</code>	<code>}</code>

Do tworzenia list używane są następujące otoczenia:

- ❶ `\begin{itemize} ... \end{itemize}`; tworzy listę wypunktowaną;
- ❷ `\begin{enumerate} ... \end{enumerate}`; tworzy listę numerowaną;
- ❸ `\begin{description} ... \end{description}`; tworzy listę opisową;

Element listy zaczyna się od polecenia `\item` (`\item[nazwa elementu]` w przypadku listy opisowej).

Listy można zagnieżdżać, można także zmienić znaki używane w wypunktowaniu i styl numerowania. Pakiet `enumitem` daje duże możliwości dopasowania stylu list do potrzeb.

Tabele

Tabele tworzymy przy pomocy otoczenia `\begin{tabular}{opis kolumn} ... \end{tabular}`.

Opis kolumn może zawierać litery `l`, `r`, `c` i pionowe kreski. Litery określają sposób wyrównywania zawartości kolumn, a kreski informują, czy kolumny mają być separowane pionowymi liniami.

Wiersze tabeli kończymy poleceniem `\\` lub `\\ \hline`, jeśli po wierszu ma być linia pozioma. Kolumny separujemy znakiem `&`.

Tabele są z zasady jednostronicowe. Jeśli chcemy korzystać z tabel automatycznie dzielonych między strony, należy skorzystać z odpowiedniego pakietu, np. `longtable`.

Do zmiany rozmiaru czcionki można użyć wbudowanych poleceń, takich jak `\tiny`, `\normalfont`, `\large`, `\Large`, `\LARGE`, `\huge` i `\Huge`.

Zmianę stylu czcionki uzyskujemy następującymi poleceniami:

- 1 `\textbf{}` lub `\bfseries`; pogrubienie;
- 2 `\textmd{}` lub `\mdseries`; pismo normalnej grubości;
- 3 `\textrm{}` lub `\upshape`; pismo proste;
- 4 `\textit{}` lub `\itshape`; italik;
- 5 `\textsl{}` lub `\slshape`; pismo pochylę;
- 6 `\textsc{}` lub `\scshape`; kapitaliki.

Grafikę można dołączać z zewnętrznych plików, służy do tego polecenie `\includegraphics`. Można ją także tworzyć przy pomocy pakietu `tikz`.

Tworzony obrazek umieszczamy wewnątrz otoczenia `\begin{tikzpicture} ... \end{tikzpicture}`, a budujemy przy pomocy poleceń generujących proste kształty geometryczne: `\filldraw`, `\draw` itd.

Możliwe jest przy tym nadawanie punktom etykiet i stosowanie konstrukcji programistycznych typu pętla, instrukcje warunkowe i zmienne.

Pakiet `algorithm2e` ułatwia zapis algorytmów w pseudokodzie. Algorytmy należy umieszczać wewnątrz otoczenia `\begin{algorithm} ... \end{algorithm}`.

Instrukcje występujące w algorytmach są generowane odpowiednimi poleceniami:

- 1 `\Input{opis danych};` opis wejścia;
- 2 `\Output{opis wyniku};` opis wyjścia;
- 3 `\uIf{warunek}{instrukcje}` i `\Else{instrukcje};` instrukcje warunkowe;
- 4 `\For{zakres pętli}{instrukcje};` pętla.

Powyższa lista nie jest kompletna, dostępnych jest dużo innych poleceń.

Organizacja dokumentu

Dostępne jednostki podziału dokumentu zależą od jego klasy.

Książki oferują podział na część wstępną (`\frontmatter`), główną (`\mainmatter`) i tylną (`\backmatter`). Oprócz tego dostępne są części (`\part`), rozdziały (`\chapter`), podrozdziały (`\section`) i mniejsze jednostki: `\subsection`, `\subsubsection` i `\paragraph`.

Mniejsze dokumenty oferują to samo, tyle, że nie ma w nich rozdziałów i części, a całość jest traktowana jak część główna.

Etykiety i automatyczna numeracja

Z każdym miejscem w tekście związany jest licznik, do którego można się dostać nadając mu etykietę.

Etykietę tworzymy poleceniem `\label{nazwa}`, a odczytujemy `\ref{nazwa}` (wartość) lub `\pageref{nazwa}` (numer strony, na którą trafiła).

To, który licznik zostanie wykorzystany, jest ustalane automatycznie przez klasę dokumentu—wewnątrz rozdziału będzie to inny licznik niż wewnątrz twierdzenia czy rysunku.

Twierdzenia, dowody i definicje

Do tworzenia otoczeń typu twierdzenie służy pakiet `amsthm`.

Nowy typ twierdzenia tworzymy poleceniem `\newtheorem{nazwa}{napis}[licznik]`. Nazwa określa nazwę tworzonego otoczenia, napis to napis pojawiający się na jego początku.

Twierdzenie, do którego nie podajemy dowodu kończymy zawsze znakiem `\qed` (w dowodach pojawia się on automatycznie na końcu).

Można zmienić styl twierdzenia przy pomocy polecenia `\theoremstyle`. Dowody umieszczamy wewnątrz otoczenia `\begin{proof} ... \end{proof}`.

Tworzenie makr

Nowe polecenia tworzymy pomocy polecenia
`\newcommand{nazwa}[liczba parametrów]{definicja}`.

Nowe otoczenia tworzymy pomocy polecenia
`\newenvironment{nazwa}[liczba parametrów]{początek}{koniec}`.

„Początek” jest wykonywany, kiedy w tekście pojawia się `\begin{nazwa}`, a „koniec”, gdy pojawia się `\end{nazwa}`.

Można zmieniać istniejące polecenia, wystarczy zmienić `new` na `renew` w powyższych konstrukcjach.

Jeśli nowe polecenie/otoczenie ma mieć parametry, to w części definiującej i -ty parametr widoczny jest jako `#i`. Parametry otoczenia są widoczne wyłącznie w jego części początkowej.

Prezentacje





Prezentacje tworzymy przy pomocy klasy beamer. Opis klasy beamer można znaleźć pod adresem <https://tug.ctan.org/macros/latex/contrib/beamer/doc/beameruserguide.pdf>.

Prezentacja składa ze slajdów, przeznaczonych dla widowni oraz notatek, widocznych jedynie dla prelegenta. Notatki tworzymy poleceniem `\note`.

Slajdy mogą zawierać automatycznie tworzone panele nawigacyjne, informację o autorze i tytule, a także informację o tym, do którego rozdziału i podrozdziału należą.

Treść slajdów umieszczamy wewnątrz otoczenia `\begin{frame} ... \end{frame}`. O stylu slajdu decyduje wybrany wcześniej, zazwyczaj w preambule, styl dokumentu.

Książki o L^AT_EX-u

-  Lamport L.: „System opracowywania dokumentów L^AT_EX. Podręcznik i przewodnik użytkownika”, WNT, Warszawa 2004.
-  Oetiker T., Partl H., Hyna I., Schlegl E., Przechlewski T., Kubiak R., Gołdasz J., Serwin M.: „Nie za krótkie wprowadzenie do systemu L^AT_EX 2_ε albo L^AT_EX 2_ε w 156 minut”, <https://ctan.gust.org.pl/tex-archive/info/lshort/polish/lshort-pl.pdf>
-  Goossens M., Mittelbach F., Samarin A.: „The L^AT_EX Companion”, Addison-Wesley 1994.
-  Goossens M., Rahtz S., Mittelbach F.: „The L^AT_EX Graphics Companion. Illustrating documents with T_EX and PostScript”, Addison-Wesley 1997.