

## 2. Crash course $\LaTeX$ a dla piszących prace licencjackie z matematyki

Michał Strzelecki · [michalst@mimuw.edu.pl](mailto:michalst@mimuw.edu.pl) · 30 września 2024

☞ Ten plik (drugi z trzech powiązanych) to lektura obowiązkowa dla wszystkich (stopki można pominąć). BARDZO proszę o zweryfikowanie, czy nie ma tu jakichś nieznanymi Państwu informacji *przed* przystąpieniem do spisywania czegokolwiek w  $\TeX$ u.

W tekście mogą znajdować się nieścisłości – chętnie przyjmę wszelkie uwagi.

### Po co powstał ten tekst?

Ogarnięcie  $\TeX$ a/ $\LaTeX$ a<sup>1</sup> na minimalnym poziomie, pozwalającym stworzyć krótki plik z jakimiś wzorami matematycznymi, powinno być stosunkowo proste. W sieci dostępne są różne pomoce.<sup>2</sup> Chciałbym jednak:

- dodatkowo ułatwić start osobom, które zetkną się z  $\TeX$ em po raz pierwszy,
- upewnić się, że wszyscy wiedzą o trochę bardziej zaawansowanych funkcjach przydatnych konkretnie przy spisywaniu *dłuższego* tekstu *matematycznego*,
- zredukować liczbę złych, nieświadomie powielanych nawyków,
- wskazać rozsądne ustawienia.

### Ogólna filozofia

*Nie* zajmujemy się bezpośrednio wyglądem i formatowaniem tekstu (pogrubione tytuły, kursywa w treściach twierdzeń, itp., itd.) ani numeracją rozdziałów, twierdzeń, wzorów. Większej ręcznej ingerencji będzie wymagało głównie odpowiednie złamanie bardziej skomplikowanych, wielolinijkowych wzorów.

Skupiamy się na treści (sformułowania twierdzeń, lematów, dowody) i zwracamy uwagę na strukturę dokumentu (rozdziały, podrozdziały, twierdzenia, lematy, dowody).

### Struktura pliku źródłowego, preambuła

Plik źródłowy (rozszerzenie `*.tex`) w środku będzie wyglądał tak:

```
\documentclass[licencjacka]{pracamgr}
\usepackage{amsmath,amsthm,amssymb}
...
\begin{document}
...
\end{document}
```

1. Wszystko przed komendą `\begin{document}` to preambuła. Tam znajdują się m.in. p.w. ustawienia związane z formatowaniem tekstu, w tym te załadowane przez klasę (plik `pracamgr.cls`, który umieszczamy w folderze z plikiem źródłowym).

---

<sup>1</sup>Patrząc: [What are TeX and LaTeX?](https://tex.stackexchange.com); [tex.stackexchange.com](https://tex.stackexchange.com) [dostęp: 29.06.2024]

<sup>2</sup>M.in. „Nie za krótkie wprowadzenie do systemu  $\LaTeX 2\epsilon$ ” czy tutorial na [www.overleaf.com](https://www.overleaf.com).

2. Komenda `\usepackage{...}` ładuje dodatkowe pakiety. P.n. przydadzą się wszystkie trzy podane wyżej.<sup>3</sup>
3. Właściwy tekst pracy licencjackiej wpisujemy po `\begin{document}`.
  - a) Powinni móc Państwo normalnie wprowadzać polskie litery (bo klasa pracy licencjackiej ładuje pakiety `inputenc`,<sup>4</sup> `fontenc`<sup>5</sup>).
4. Pusty szablon pracy i przykładowy plik wypełniony treścią znajda Państwo na stronie wydziału.<sup>6</sup>

## Rozdziały, podrozdziały, akapity

Ze względu na swoją długość praca licencjacka musi mieć przejrzystą strukturę.

5. Komendy `\chapter{...}`, `\section{...}` tworzą nowy rozdział, podrozdział (sekcję) o podanym tytule; zostanie im *automatycznie* nadany odpowiedni numer (i będzie się potem można łatwo do nich odwołać – patrz sekcja o komendach `\label`, `\ref` poniżej).
6. Nowy akapit tworzymy po prostu zostawiając w edytorze *pustą linię* (tj. wstawiamy *dwa* enter).
  - a) Nie należy ręcznie łamać akapitów za pomocą komend `\\` czy `\newline`, ani ręcznie modyfikować wcięć za pomocą komend `\hspace{...}` czy `\noindent`.

## Wyrażenia matematyczne

Krótkie wyrażenia matematyczne występujące w tekście (ang. *inline math*) wpisujemy w pojedynczych dolarach: `$...$` (długie i skomplikowane wzory – patrz następna sekcja); można też użyć `\(...\)`.<sup>7</sup>

7. Do wpisania specjalnych symboli (takich jak np. ułamki, greckie litery, funkcje trygonometryczne i inne operatory, sumy, całki, itp., itd.) istnieją specjalne komendy, np.:
  - a) `\frac{3}{4}` (ang. *fraction*),
  - b) `\alpha`, `\omega`, `\Omega` (małe i wielkie litery),
  - c) `\sin`, `\cos`, `\ln`, `\lim`, `\liminf` (koniecznie z ukośnikiem!),
  - d) `\sum_{i=1}^n` (ang. *sum*), `\int_0^{\infty}` (ang. *integral*).
8. W wyrażeniach matematycznych występujących w tekście unikamy wysokich lub piętrowych napisów.
  - a) Lepiej napisać  $\exp(-x^2/2)$  czy  $(a^2 + b^2)^{-1/2}$  niż zapisać te same wyrażenia za pomocą `e^{-\frac{x^2}{2}}` oraz `\frac{1}{\sqrt{a^2+b^2}}`.
  - b) Nieodpowiedni zapis da mniej czytelne napisy (małe litery) i spowoduje powstawanie brzydkich pustych prześwitów między wierszami (rozjeżdżanie się linijek).

<sup>3</sup>Patrz: What does each AMS package do?; [tex.stackexchange.com](http://tex.stackexchange.com) [dostęp: 29.06.2024]

<sup>4</sup>Patrz: fontenc vs inputenc; [tex.stackexchange.com](http://tex.stackexchange.com) [dostęp: 29.06.2024]

<sup>5</sup>Patrz: Why should I use \usepackage[T1]{fontenc}?; [tex.stackexchange.com](http://tex.stackexchange.com) [dostęp: 29.06.2024]

<sup>6</sup>We wrześniu 2024 dostępny w zakładce Studia/Student, w dziale Informator dla studentów, na podstronie „Prace i egzaminy dyplomowe”.

<sup>7</sup>Patrz: Are \ ( and \) preferable to dollar signs for math mode?; [tex.stackexchange.com](http://tex.stackexchange.com) [dostęp: 27.09.2024]

## Środowiska `equation` oraz `align`

Jeśli wzór jest długi (powiedzmy dłuższy niż pół szerokości wiersza), skomplikowany (zagnieżdżone nawiasy, piętrowe ułamki), szczególnie istotny lub chcemy, żeby można się było do niego później odwołać, to wpisujemy go w osobnym wierszu lub wierszach (ang. *display math*). Piszemy

```
\begin{srodowisko}
...
\end{srodowisko}
```

(w miejscu `srodowisko` na ogół znajdzie się `equation*` lub `align*` albo ich wersja bez gwiazdki).

9. Środowisko `equation*` służy do zapisania wzoru mieszczącego się w jednej linii.
  - a) Alternatywnie można użyć `\[...]` (robi dokładnie to samo).
  - b) Lepiej nie używać `$$...$$`.<sup>8</sup>
10. Środowisko `align*` służy do składania wzorów czy ciągów (nie-)równości, które są zbyt długie by zmieścić się w jednej linijce; `\\` wskazuje miejsca złamania kolejnych linijek, zaś `&` – miejsca ich wyrównania (na ogół znaki (nie-)równości). Piszemy więc mniej więcej tak:

```
\begin{align*}
... &= ..... \\
&= ..... \\
&= ...
\end{align*}
```

- a) Proszę nie używać środowiska `eqnarray`.<sup>9</sup>
  - b) Przypadek wzorów z bardzo długimi wyrażeniami – patrz trzeci plik.
11. Środowiska `equation`, `align` (bez gwiazdek) działają j.w., ale dodatkowo ponumerują wiersze (żeby można się było potem do nich odwołać – patrz sekcja o komendach `\label`, `\eqref` poniżej).
12. Sugerowałbym dodać do preambuły

```
\numberwithin{equation}{chapter}
```

(wtedy drugi wzór w Rozdziale 1 dostanie numerek (1.2); dzięki temu Czytelnik, który natknie się na odwołanie do wzoru (1.2) w Rozdziale 3, będzie wiedział, gdzie szukać tego wzoru).

13. Wyśrodkowany wzór jest nadal częścią zdania, a zatem na jego końcu czasami powinien występować znak interpunkcyjny (kropka – jeśli kończy zdanie, przecinek – jeśli potem występuje np. spójnik „gdzie”).

## Twierdzenia i dowody

Do zapisywania twierdzeń itp. używamy odpowiednich środowisk. Trzeba je najpierw zdefiniować – sugerowałbym umieszczenie w preambule następujących definicji:

<sup>8</sup>Patrz: Why is `\[ ... ]` preferable to `$$ ... $$`?; [tex.stackexchange.com](https://tex.stackexchange.com) [dostęp: 27.06.2024]

<sup>9</sup>Patrz: `eqnarray` vs `align`; [tex.stackexchange.com](https://tex.stackexchange.com) [dostęp: 20.06.2024]

```

\newtheorem{theorem}{Twierdzenie}[chapter]
\newtheorem{lemma}[theorem]{Lemat}
\newtheorem{corollary}[theorem]{Wniosek}
\newtheorem{proposition}[theorem]{Stwierdzenie}

```

```

\theoremstyle{definition}
\newtheorem{definition}[theorem]{Definicja}
\newtheorem{example}[theorem]{Przykład}

```

```

\theoremstyle{remark}
\newtheorem{remark}[theorem]{Uwaga}

```

(w stopkach podaję: ogólne wyjaśnienie tej składni,<sup>10</sup> uzasadnienie wyboru numeracji,<sup>11</sup> uzasadnienie wyboru formatowania<sup>12</sup>). Potem piszemy np.

```

\begin{theorem}
...
\end{theorem}

```

(w miejsce trzech kropek wpisujemy oczywiście treść twierdzenia).

14. Twierdzenia itp. zostaną *automatycznie* ponumerowane (i będzie się potem można łatwo do nich odwołać – patrz sekcja o komendach `\label`, `\ref` poniżej) oraz zostanie zastosowane odpowiednie formatowanie tekstu.
15. Dowody wpisujemy za pomocą

```

\begin{proof}
...
\end{proof}

```

(nie trzeba definiować tego środowiska).

---

<sup>10</sup>W pierwszym nawiasie klamrowym po `\newtheorem` jest nazwa środowiska, której będziemy używać w edytorze (można zmienić wg upodobań); w drugim nawiasie klamrowym jest nazwa nagłówka (po polsku), którą zobaczymy w pliku wyjściowym. Dodatkowe argumenty `[theorem]`, `[chapter]` ustawiają numerację, zaś `\theoremstyle` ustawia formatowanie tekstu – patrz następne stopki. Oczywiście można podobnie zdefiniować więcej środowisk (Fakt, Hipoteza, Pytanie itd.).

<sup>11</sup>Sugerowałbym numerowanie kolejnych wyników wg wzoru: Twierdzenie 1.1, Lemat 1.2, Twierdzenie 1.3, Wniosek 1.4, Lemat 1.5, tj. łącznie wszystkie typy stwierżeń (po to w definicjach jest wszędzie `[theorem]`), pierwsza liczba to numer rozdziału (po to jest `[chapter]`). To dużo czytelniejsze niż: Twierdzenie 1, Lemat 1, Twierdzenie 2, Wniosek 1, Lemat 2. Ponadto dzięki temu Czytelnik, który szuka Lematu 2.8, wie, że musi szukać w Rozdziale 2 (a jeśli natrafi tam na Definicję 2.12, to wie ponadto, że jest za daleko).

<sup>12</sup>Treści twierżeń, stwierżeń, lematów, wniosków na ogół składa się kursywą (bo są ważne i chcemy je wyróżnić), nagłówki często są pogrubione (żeby je aktywnie wyróżnić) – tak robi domyślnie `\newtheorem`. Jeśli pojawiają się definicje, to ich treści składałbym zwykłym krojem (bo są mniej ważne niż twierdzenia) – po to włączamy `\theoremstyle{definition}`. Uwagi są najmniej ważne, więc w nich nie wyróżniałbym nagłówka pogrubieniem, a jedynie kursywą (tak jak w przypadku dowodów) – po to włączamy `\theoremstyle{remark}`. Jeśli ktoś wolałby np. składać nagłówki uwag **boldem** albo **kapitałkami**, to może poczytać, jak definiować inne style.

## Komendy `\label`, `\ref`, `\eqref`

L<sup>A</sup>T<sub>E</sub>X sam ponumeruje rozdziały, sekcje, twierdzenia, lematy, wzory, ilustracje, itp. występujące w pracy. Komenda `\label{...}` umieszczona wewnątrz np. rozdziału, twierdzenia czy wyśrodkowanego wzoru nada mu etykietę (w miejsce trzech kropek wpisujemy wybraną przez siebie nazwę). Żeby odwołać się w innym miejscu do odpowiedniego fragmentu tekstu używamy komendy `\ref{...}` (w przypadku rozdziałów, twierdzeń itp.) lub `\eqref{...}` (w przypadku wzorów).

16. Pisałbym: „na mocy Lematu 1.2”, „w Rozdziale 3” (a nie: „na mocy lematu 1.2”, „w rozdziale 3”) oraz „ze wzoru (4.5)”, „kończy to dowód twierdzenia”, „w następnym rozdziale” (tu małą literą).
17. Przed odwołaniami warto dać twardą spację (tj. napisać `Lemat~\ref{...}`), żeby numerki twierdzenia czy wzoru nie uciekł do następnej linii.

## Makra

Żeby uzyskać literę  $\mathbb{P}$  trzeba użyć komendy `\mathbb{P}`.<sup>13</sup> Może być wygodniej i czytelniej zdefiniować w preambule makro:

```
\newcommand*\PP{\mathbb{P}} % ozn. na prawdopodobieństwo
```

i móc potem pisać `\PP(X \geq 0)` (zamiast `\mathbb{P}(X \geq 0)`).

18. Zastosowana powyżej wersja komendy `\newcommand*` (z gwiazdką) wystarczy do większości prostych zastowań i będzie najbezpieczniejsza.<sup>14</sup>
  - a) Raczej nie będzie powodu, by używać komendy `\def`.<sup>15</sup>
  - b) Lepiej nie używać jednoliterowych nazw makr.<sup>16</sup>
  - c) Można też tworzyć bardziej skomplikowane makra z argumentami.

19. Nie warto tworzyć makr typu:

```
\newcommand*\begthm{\begin{theorem}} % NIE!
```

– sprawią, że plik źródłowy będzie mniej czytelny.

20. Warto tworzyć semantyczne makra typu:

```
\newcommand*\closure{\overline} % domknięcie zbioru
```

– sprawią, że plik źródłowy będzie bardziej czytelny. Będzie też łatwiej w razie czego zmienić notację.

## Bibliografia

Za pomocą `\cite{...}` cytujemy pozycje wpisane w bibliografii. Po więcej szczegółów i wytycznych odsyłam do trzeciego pliku.

<sup>13</sup>Do tego jest potrzeby pakiet `amsfonts`, który jest ładowany przez pakiet `amssymb`.

<sup>14</sup>Patrz: What's the difference between `\newcommand` and `\newcommand*`?; [tex.stackexchange.com](https://tex.stackexchange.com) [dostęp: 26.09.2024]

<sup>15</sup>Patrz: What is the difference between `\def` and `\newcommand`?; [tex.stackexchange.com](https://tex.stackexchange.com) [dostęp: 24.09.2024]

<sup>16</sup>Patrz: Single letter macros; [tex.stackexchange.com](https://tex.stackexchange.com) [dostęp: 26.09.2024]

## Co jeszcze dodać do preambuły?

Nie warto ładować na wszelki wypadek tony pakietów, jeśli nie są nam potrzebne lub nie wiemy, co robią.

21. Na pewno przyda się `\usepackage{amsmath,amsthm,amssymb}` (i przytoczone wyżej definicje twierzeń).<sup>17</sup>
22. Chyba warto dodać `\usepackage{lmodern}` (użyje trochę lepszych czcionek).<sup>18</sup>
23. Chyba warto dodać `\usepackage{microtype}` (trochę poprawi ogólny wygląd tekstu).
24. Raczej wygodnie jest dodać `\usepackage[hidelinks]{hyperref}` (sprawi, że wszystkie odnośniki będą klikalne; podana opcja usunie brzydkie kolorowe ramki wokół odnośników).
  - a) To raczej powinno zostać załadowane jako ostatni pakiet.<sup>19</sup>

## Inne wskazówki

25. Rozsądny edytor będzie kolorował składnię i podpowiadał lub sam uzupełniał wprowadzane komendy.
26. Warto dodawać komentarze do kodu – mogą wyglądać j.w. w przykładowych definicjach makr lub np. tak:

```
%-----  
% Dowód gł. twierdzenia:  
%-----
```

27. Pojedyncze entery wstawione w edytorze nie rozpoczynają nowego akapitu i są traktowane jak zwykłe spacje, więc warto ich używać, żeby zwiększyć czytelność plik źródłowego.
  - a) Jeśli linie będą krótsze, to będzie łatwiej zlokalizować błędy.
  - b) Wszystkie `\begin{...}`, `\end{...}` warto umieszczać w osobnej linii.
28. Może się przydać forum [tex.stackexchange.com](https://tex.stackexchange.com) oraz strona [ctan.org](https://ctan.org).
  - a) W szczególności informacji o bardziej zaawansowanych funkcjach można szukać w krótkim<sup>20</sup> i dłuższym<sup>21</sup> przewodniku do pakietu `amsmath`.

---

<sup>17</sup>Patrz: [What does each AMS package do?](https://tex.stackexchange.com); [tex.stackexchange.com](https://tex.stackexchange.com) [dostęp: 29.06.2024]

<sup>18</sup>Patrz: [Is it still useful to load the lmodern package?](https://tex.stackexchange.com); [tex.stackexchange.com](https://tex.stackexchange.com) [dostęp: 26.09.2024]

<sup>19</sup>Patrz: [Which packages should be loaded after hyperref instead of before?](https://tex.stackexchange.com); [tex.stackexchange.com](https://tex.stackexchange.com) [dostęp: 26.09.2024]

<sup>20</sup>Patrz: [short-math-guide – Guide to using amsmath and related packages to typeset mathematical notation with L<sup>A</sup>T<sub>E</sub>X](https://ctan.org); [ctan.org](https://ctan.org) [dostęp: 30.09.2024]

<sup>21</sup>Patrz: [amsmath – AMS mathematical facilities for L<sup>A</sup>T<sub>E</sub>X](https://ctan.org); [ctan.org](https://ctan.org) [dostęp: 30.09.2024]