

**Józef Roehrich**

# **L<sup>A</sup>T<sub>E</sub>X z Józkiem**

*45 stron o tym, jak zacząć składać perfekcyjne dokumenty w L<sup>A</sup>T<sub>E</sub>X-u*

**WWW.ODWROCONALOGIKA.PL**

2026

# *Spis treści*

Wstęp	4
Zacznij pisać	8
Podstawy formatowania tekstu	13
Listy i wyliczenia	17
Wzory w $\LaTeX$ -u	21
Rysunki	27
Odwołania do etykiet	29
Tabele	32
Bibliografia	40
Epilog	45

# Zacznij pisać

*Hello World!*

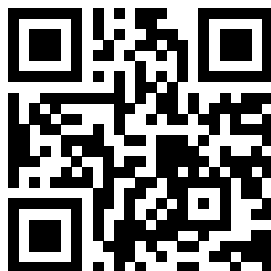
Każdy programista, niezależnie, w jakim języku programuje, zaczyna swoją przygodę z nowym językiem od programu typu „Hello World!”. W tym przypadku będzie tak samo — zaczniemy od razu pisać w  $\LaTeX$ -u. Ale chwilunia!<sup>6</sup> Przecież nie mam środowiska, w którym będę pisał. Masz dwie opcje: pierwsza to instalacja  $\LaTeX$ -a na własnym komputerze, druga — którą ja bardziej Ci polecam, to Overleaf.<sup>7</sup> Na początek proponuję Ci darmowy plan Overleaf — ma on pewne ograniczenia, natomiast ma niewątpliwie kilka zalet:

- jeśli założysz tam darmowe konto, możesz zacząć pisać w  $\LaTeX$ -u już za 10 minut;
- w Overleaf masz ogarniętą instalację  $\LaTeX$ -a — wszystkie niezbędne pakiety są już tam zainstalowane, więc nie będziesz miał żadnych problemów;
- zaletą Overleaf jest również to, że działa w chmurze: jest niezależny od systemu, na którym pracujesz, dzięki czemu jeżeli coś działa u mnie, to zadziała też u Ciebie.

Super! Zatem załóż konto na Overleaf, a za 10 minut spotkamy się na następnej stronie i stworzymy pierwszy dokument. To będzie tylko taki „pierwszy dokumencik”, a potem już będzie — jak to mówią — z góry.

<sup>6</sup> Tak mówił Królik Bugs!

<sup>7</sup> Ja właściwie łączę te dwie opcje: mobilność Overleafa oraz brak ograniczeń na własnym komputerze.



## Mój pierwszy dokument

Na pierwszy ogień idzie najprostszy kod. Gdy założysz swój pierwszy projekt w Overleafie twym oczom ukaże się startowy plik `main.tex` o następującej treści (Kod 2).

```

1 \documentclass{article}
2 \usepackage{graphicx} % Required for inserting images
3
4 \title{Mój pierwszy dokument}
5 \author{Józef Roehrich}
6 \date{November 2025}
7
8 \begin{document}
9
10 \maketitle
11
12 \section{Introduction}
13
14 \end{document}

```

Kod 2: Pierwszy kod

Po kompilacji możesz zobaczyć swój pierwszy  $\text{\LaTeX}$ -owy PDF, choć właściwie jeszcze nic nie zrobiłeś. Zmienimy od razu kilka rzeczy (linia 1 w kodzie 2).

```

1 \documentclass[a4paper, onside]{mwbk}

```

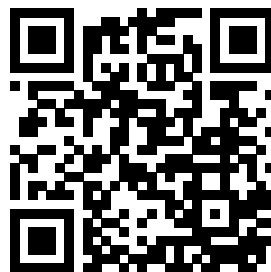
Dzięki tym zmianom zastosujemy klasę dokumentu typu książka: klasa `mwbk` to polski odpowiednik klasy `book`. Klasa `mwbk` zawiera domyślne ustawienia typograficzne, zgodne z polskimi normami; jest więc bardziej odpowiednia do zastosowań w polskich książkach oraz pracach dyplomowych. Napiszmy zatem coś, jakiś fragment tekstu (zaczynj pisać w linii 13 Listingu 2).

```

1 \section{Introduction}
2 Józek miał dziś udany dzień, zaczął pisać swojego e-booka.

```

Jak widzisz, nie wygląda to zbyt dobrze, na dodatek wywaliło nam pierwszy błąd:



Przykład wideo: rozdziały i podrozdziały

LaTeX Error: Command \k unavailable in encoding OT1.

Napisałem celowo zdanie w języku polskim, aby użyć polskich znaków — nie ma ich w zestawie znaków OT1. Musimy zastosować inny zestaw znaków. Naprawimy to, dodatkowo naprawimy dwie inne rzeczy:

```
1 \usepackage{graphicx} % Required for inserting images
2 \usepackage[utf8]{inputenc}
3 \usepackage[T1]{fontenc}
4 %\usepackage[polish]{babel}
```

W drugiej linii powyższego kodu (kod 3) informujemy  $\LaTeX$ -a, że kod źródłowy używa kodowania utf8. W trzeciej linii wymuszamy użycie zestawu znaków T1, a w czwartej linii konfigurujemy pakiet babel do obsługi języka polskiego (pozostaw tę linię zakomentowaną, komentujemy kod poprzez wstawienie znaku %<sup>8</sup>). Teraz, po kompilacji, wszystko powinno działać poprawnie — nie powinien być żaden błąd kompilacji.

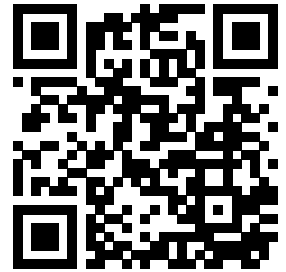
### Rozdziały i podrozdziały

Ten dokument wygląda dość „tyso”, musimy stworzyć rozdziały i podrozdziały. Aby stworzyć rozdział posłużymy się komendą `\chapter{}`, natomiast komendę `\section{}` w naszym dokumencie już mamy, `\section{}` to podrozdział.

```
1 \chapter{Wstęp} %rozdział
2 \section{Introduction} %podrozdział
3 Józek miał dziś ciężki dzień!
```

Kod 3: UTF-8 i polskie czcionki

<sup>8</sup> Ciekawe, jak zatem wstawił tutaj znak %?



Przykład wideo: rozdziały i podrozdziały

Skompilowałeś plik: teraz wyjściowy plik PDF wygląda już dużo lepiej. Rozdział zaczyna się słowami *Chapter 1* (tę warstwę składu tekstu ogarnia  $\LaTeX$  nie musisz tego pisać samemu). W kolejnej linii jest tytuł rozdziału *Wstęp*, a w następnej linii pojawił się podrozdział *1.1 Introduction*. Jest tylko jeden mały problem: w dokumencie pisanym w języku polskim rozdział powinien być rozdziałem, a nie *chapterem*. Wystarczy, że odkomentujesz tę linię `\usepackage[polish]{babel}` i wszystko będzie działało tak, jak trzeba. Tytuł podrozdziału się nie zmienił, wciąż jest *Introduction*: to wpisaliśmy „z ręki”, nazwij ten podrozdział *Wprowadzenie* i będzie gotowe.

```
1 \section{Wprowadzenie} %podrozdział
```

## Spis treści

Spis treści jest bardzo ważnym elementem każdej pracy. Zasadą jest, że powinien być generowany automatycznie — jako autor dokumentu nie powinieneś zaprzętać sobie tym głowy. Komputery zostały stworzone właśnie po to, aby wykonywać za nas tego typu powtarzalną pracę. W  $\LaTeX$ -u spis treści tworzony jest na podstawie rozdziałów i podrozdziałów, które zdefiniowaliśmy w poprzednim podrozdziale. Aby go wygenerować, wystarczy umieścić komendę `\tableofcontents` zaraz po komendzie `\begin{document}`, w tej części pliku, w której wywołujemy także stronę tytułową poleceniem `\maketitle`.

```
1 \begin{document}
2
3 \maketitle
4 \tableofcontents
```



Przykład wideo: spis treści

Jeżeli jesteś już tutaj, oznacza to, że pracujesz w  $\LaTeX$ -u już ok. 15 minut: zobacz, jak dużo udało się nam już zrobić. Twój dokument nabiera dopiero rumieńców.

## Pisanie tekstów w $\LaTeX$ -u

Kontrola tekstu w  $\LaTeX$ -u opiera się na logicznym opisaniu struktury zamiast ręcznego ustawiania wyglądu.<sup>9</sup> Nowy akapit tworzymy, pozostawiając pustą linię w kodzie źródłowym — to sygnał dla kompilatora, że rozpoczynamy nową myśl. Jeśli zależy nam jedynie na przejściu do nowej linii bez rozpoczynania akapitu, używamy polecenia `\` lub `\newline`. Warto pamiętać, że  $\LaTeX$  ignoruje wielokrotne spacje i złamania linii wpisane w kodzie źródłowym, ponieważ samodzielnie dba o poprawne łamanie tekstu, zachowując profesjonalny wygląd dokumentu bez ręcznej ingerencji.

Najłatwiej zobaczysz różnicę na tych trzech krótkich przykładach — w tym celu załóż trzy nowe podrozdziały `\section{Coś tam!}`, wklej poniższy kod i skompiluj plik.<sup>10</sup>

```

1 \chapter{Trzy krótkie przykłady} %rozdział
2 \section{Jeden enter nic nie robi} %podrozdział
3 To jest tekst pierwszej linii.
4 A to nadal jest ten sam akapit, mimo że nacisnąłem Enter.
5
6 \section{Przejście do nowej linii} %podrozdział
7 To jest tekst pierwszej linii.\
8 A to nowa linia, wymuszona poleceniem.
9
10 \section{Nowy akapit} %podrozdział
11 Akapit pierwszy. %po tym akapicie zostaw pustą linię
12
13 Akapit drugi. Tutaj możesz jeszcze coś dopisać.
```

Mam nadzieję, że poczułeś już bluesa! Zobacz, jak szybko nam idzie — jesteśmy dopiero na 12 stronie, a już mamy tyle zrobione. Co więcej, dotychczas pojawił się tylko jeden błąd kompilacji, więc rezultaty są naprawdę świetne.

<sup>9</sup> Tak jest w Wordzie i innych edytorach WYSIWYG (ang. *What You See Is What You Get*)

<sup>10</sup> Przy okazji zobaczysz również, jak rozrasta się spis treści.

Kod 4: Łamanie linii w  $\LaTeX$ -u

# Rysunki

## *Obraz jest jak tysiąc słów*

Jeden obraz potrafi przekazać więcej treści niż tysiąc słów. Rysunki są integralną częścią większości prac dyplomowych i bardzo często pochodzą z książek. W tym e-booku zajmę się wyłącznie  $\LaTeX$ -ową stroną tematu, czyli sposobem wstawiania ilustracji do tekstu. Nie będę natomiast omawiał samego procesu tworzenia rysunków — to materiał na osobną publikację. Mogę jednak podsunąć Ci film na YouTube, w którym pokazuję, w jaki sposób skanować rysunki z książek. Rysunek musi mieć odpowiednią jakość, zatem gdy skanujesz ilustracje do pracy dyplomowej, postaraj się, aby jego rozdzielczość wynosiła 600 dpi. Taka wartość pozwoli wydrukować Twoją pracę na każdej drukarce laserowej, a jednocześnie stanowi doskonały kompromis — większa rozdzielczość powoduje, że plik PDF generowany przez  $\LaTeX$ -a staje się bardzo duży.

## *Podpisy i odwołania do obrazków*

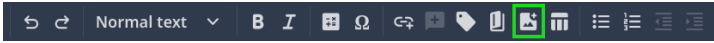
Za chwilę wstawimy do tekstu pierwszy rysunek. Na początek zaproponuję Ci jakiegoś gotowca. Na mojej stronie znajdziesz gnuplota pracującego online, dzięki niemu wygenerujesz plik potrzebny do ćwiczeń (zapisz go jako plik w formacie PNG). Możesz pozmienić sobie dane wejściowe.



*Przykład wideo: Skanowanie rysunków*



*Gnuplot online  
gnuplot.odwroconalogika.pl*



Rys. 3: Wstaw rysunek

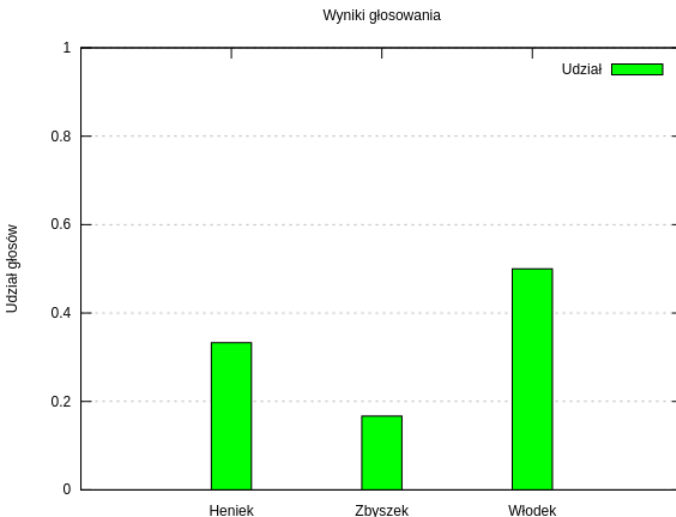
Aby wstawić rysunek, kliknij ikonę Wstaw rysunek (Rys. 3) albo wklej kod 14. Overleaf świetnie ułatwia pracę: gdy zaczniesz edytować ścieżkę do pliku PNG w linii 3, Overleaf automatycznie podpowie dostępne pliki znajdujące się w katalogu rysunki/. Opis rysunku wpiszesz w linii 4, wewnątrz polecenia `\caption{...}`, a etykietę (czyli identyfikator potrzebny do późniejszych odwołań) możesz wpisać w linii 5, w komendzie `\label{fig:...}`.

```

1 \begin{figure}
2   \centering
3   \includegraphics[width=0.5\linewidth]{rysunki/wykres.
   png}
4   \caption{Enter Caption}
5   \label{fig:placeholder}
6 \end{figure}

```

Kod 14: Wstaw rysunek



Rys. 4: Przykładowy rysunek

## **To jest próbka.**

Pełny e-book (PDF, 45 stron) dostajesz po zakupie:

**29 zł**

[Kup pełną wersję w sklepie](#)

W pełnej wersji m.in.: tabele, rysunki, wzory, odwołania, bibliografia, gotowe przykłady.