

Technologie informacyjne

Wprowadzenie do L^AT_EXa

Rafał Kabaciński

11 października 2019

Spis treści

1	Czym jest L^AT_EX	1
2	Rozpoczęcie pracy w L^AT_EXu	2
2.1	Struktura dokumentu	2
2.2	Komendy w L ^A T _E Xu	4
2.3	Pisanie dokumentów w języku polskim	4
3	Podstawy pisania dokumentów w L^AT_EXu	5
3.1	Tytuł i autor	5
3.2	Spis treści	6
3.3	Otoczenia	7
3.4	Formatowanie tekstu	7
4	Formuły matematyczne	8
5	Tabele i rysunki	10
6	Kod źródłowy w dokumentach	12

1 Czym jest L^AT_EX

L^AT_EX jest zestawem makr dla języka i kompilatorów T_EX. Z punktu widzenia użytkownika jest to język znaczników, jak HTML, czy XML, czyli język pozwalający zapisać nie tylko treść dokumentu ale również jego wygląd. W porównaniu do edytorów WYSIWYG (ang. what you see is what you get) pozwala na uzyskanie większej kontroli nad wyglądem dokumentu, oraz wymusza nadanie jednolitej struktury. Dzięki temu łatwe jest pisanie spójnych stylistycznie dokumentów oraz zautomatyzowane tworzenie różnego typu spisów, lub odnośników. Wadą takiego podejścia do pisania dokumentów jest często duży nakład pracy potrzebny do głębokiego zmodyfikowania, lub przygotowania nowego stylu.

Więcej informacji na temat L^AT_EXa można znaleźć w następujących źródłach:

- bezpłatny podręcznik, w języku angielskim: [3],
- polskie tłumaczenie powyższego podręcznika: [4],

- polski podręcznik na portalu WikiBooks: [2],
- dokumentacja portalu Overleaf: [1].

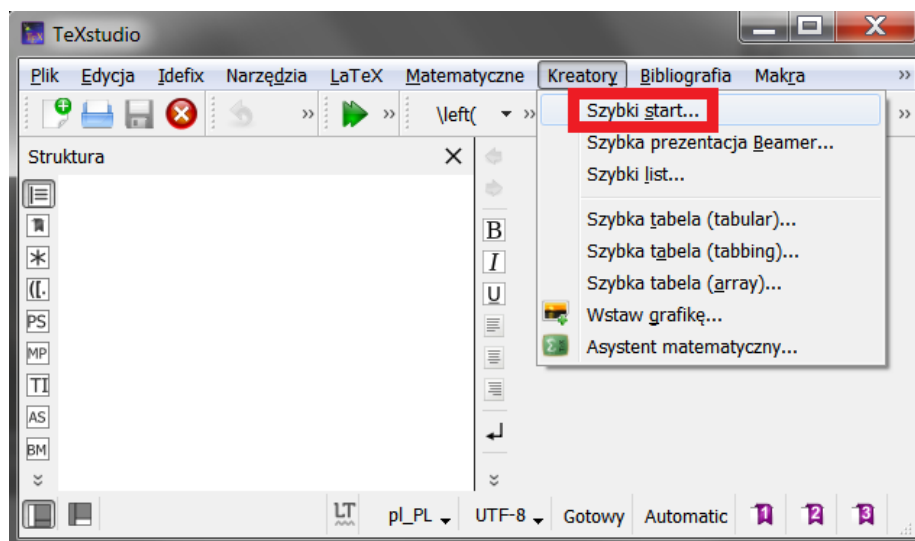
2 Rozpoczęcie pracy w \LaTeX u

Dokumenty w \LaTeX u pisane są w osobnym pliku, nazywanym plikiem źródłowym, względem dokumentu wynikowego. Aby uzyskać plik wynikowy w żądanym formacie, jak na przykład *pdf*, konieczna jest kompilacja pliku źródłowego.

Zadanie: W instrukcji pojawiać się będą bloki takie jak ten, w których będą umieszczone zadania do wykonania. Prawidłowe wykonanie wszystkich poleceń pozwoli wygenerować dokument taki sam jak załączony do poniższej instrukcji dokument przykładowy.

2.1 Struktura dokumentu

Pliki źródłowe \LaTeX a muszą składać się z *preambuły* i *ciała* dokumentu. We współczesnych edytorach można znaleźć kreatory pozwalające wstawić gotową strukturę bez konieczności samodzielnego wpisywania wszystkich elementów. Na ilustracji 1 zaprezentowano gdzie można znaleźć taki kreator w programie TeXstudio.

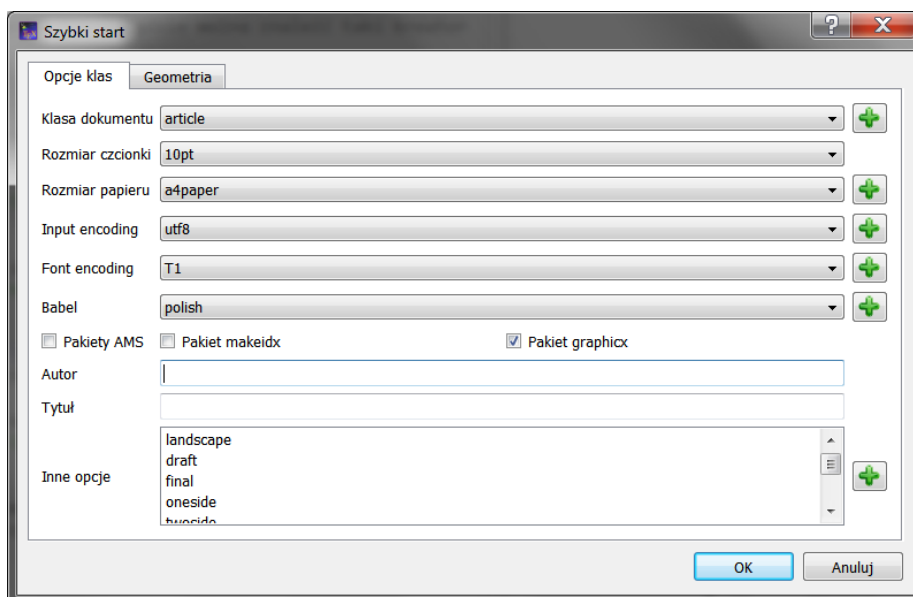


Ilustracja 1: Umiejscowienie kreatora dokumentów w programie TeXstudio

Po uruchomieniu kreatora pojawia się okno zaprezentowane na ilustracji 2

Zadanie: Utwórz nowy dokument za pomocą kreatora w programie TeXstudio korzystając z ustawień jak na ilustracji 2

Preambuła to wszystko co rozpoczyna się od komendy `\documentclass` i kończy na komendzie `\begin{document}`. Ciało dokumentu zawiera się wewnątrz otoczenia `document`, czyli znajduje się pomiędzy komendami `\begin{document}` a `\end{document}`. W preambule definiuje się styl dokumentu i dodaje pakiety



Ilustracja 2: Okno kreatora dokumentów w programie TeXstudio

rozszerzeń. Dobrą praktyką jest definiowanie nowych komend tylko w preambule, ale nie jest to wymagane. Ciało dokumentu zawiera treść wraz ze znacznikami formatowania. Nie można wprowadzać tekstu poza ciałem dokumentu, przeważnie uniemożliwi to kompilację.

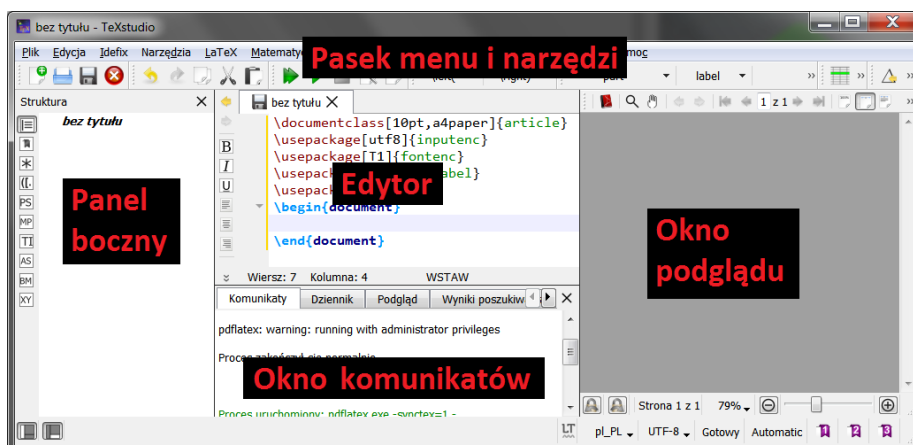
Główne okno TeXstudio zaprezentowano na ilustracji 3. W jego skład wchodzi pasek menu i narzędzi (domyślnie u góry), panel boczny (domyślnie po lewej), okno edytora (domyślnie po środku), okno komunikatów (domyślnie na dole) i okno podglądu (domyślnie po prawej). Aby skompilować dokument należy kliknąć przyciski oznaczone pojedynczą bądź podwójnym zielonym trójkątem (patrz ilustracja 4). Ikona z pojedynczym trójkątem oznacza samą kompilację, natomiast z podwójnym kompilację i podgląd.

Uwaga: aby pojawił się podgląd dokument musi zostać skompilowany z powodzeniem. Jeżeli ciało dokumentu jest puste dokument nie zostanie skompilowany.

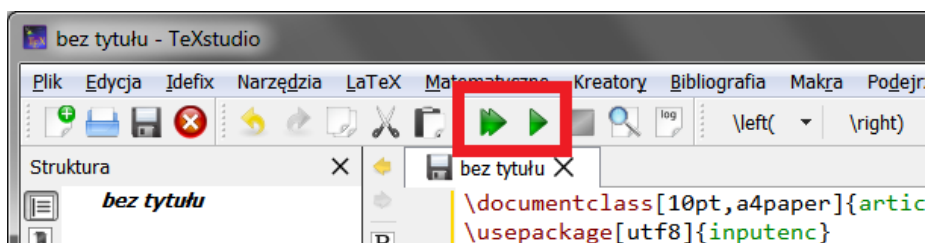
Uwaga: Okno podglądu pokazuje wynik ostatniej udanej kompilacji. Informację czy kompilacja zakończyła się powodzeniem można znaleźć w oknie komunikatów!

Preambuła wstawiona przez edytor zawiera od razu kilka pakietów. Pierwszym z nich jest pakiet `\usepackage[utf8]{inputenc}` informujący kompilator o kodowaniu pliku źródłowego. Po utworzeniu pliku nie należy zmieniać oznaczenia kodowania, gdyż uniemożliwi to kompilację dokumentu!

Zadanie: Skompiluj z powodzeniem swój pierwszy dokument w \LaTeX u.



Ilustracja 3: Okno programu TeXstudio



Ilustracja 4: Przyciski kompilacji w programie TeXstudio

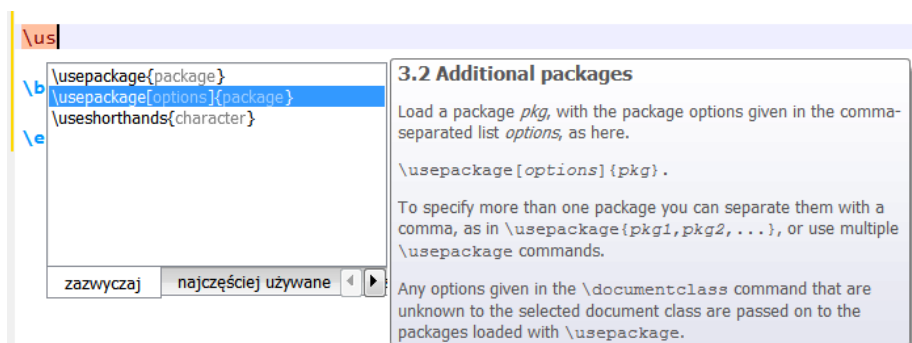
Uwaga: \LaTeX jest zbiorem wielu osobnych programów, wymieniających ze sobą informacje za pomocą plików tymczasowych pojawiających się w tym samym folderze co plik kompilowany. Z tego względu dobrą praktyką jest umieszczanie tych dokumentów w osobnych folderach.

2.2 Komendy w \LaTeX u

Komenda w \LaTeX u składa się z ukośnika wstecznego: „\” i ciągu literowo-cyfrowego, każdy inny znak w tym spacje przerywa komendę. Jeżeli komenda przyjmuje argumenty obowiązkowe znajdują się one pomiędzy nawiasami nawiasów klamrowych: „{ }”. Argumenty opcjonalne umieszczane są w nawiasach kwadratowych: „[]”. Kolejność występowania nawiasów zależy od komendy, dlatego warto korzystać z podpowiedzi oferowanych przez współczesne edytory. Przykład takich podpowiedzi zaprezentowano na ilustracji 5. Wyboru wariantu dokonuje się strzałkami na klawiaturze i potwierdza klawiszem *enter*.

2.3 Pisanie dokumentów w języku polskim

Drugim pakietem wstawionym przez kreator jest `\usepackage[T1]{fontenc}` informujący kompilator o kodowaniu czcionek w dokumencie wynikowym. Dla języka polskiego prawidłowym zestawem jest *T1*. Brak tego pakietu lub ustawienie niewłaściwej opcji spowoduje błędy w kompilacji lub wstawienie niewłaściwych



Ilustracja 5: Podpowiedzi w edytorze TeXstudio

liter w miejsce polskich znaków diakrytyzowanych. Trzecim pakietem jest pakiet `\usepackage[polish]{babel}`. Pakiet ten pozwala na dostosowanie między innymi podziału słów między wierszami według reguł języka polskiego. Oprócz powyższych warto dodać jeszcze trzeci pakiet: `\usepackage{polski}`, dostosowujący pozostałe aspekty L^AT_EXa.

Zadanie: Dodaj, korzystając z podpowiedzi edytora, pakiet polski i skompiluj dokument.

Uwaga: Pakiety w L^AT_EXu są pisane przez niezależnych autorów i może między nimi dochodzić do konfliktów. Czasem pakiety są w stanie koegzystować bez konieczności ingerencji w ich makra, o ile są dodane w określonej kolejności. Przykładem jest pakiet *babel* z opcją *polish* i pakiet *amssymb*. Jeżeli pakiet AMS jest dodany przed pakietem *babel* dokument można skompilować, w odwrotnej kolejności prowadzi to do błędu. Najbezpieczniejszą strategią jest dodawanie nieznanymi pakietów pojedynczo.

3 Podstawy pisania dokumentów w L^AT_EXu

3.1 Tytuł i autor

Elementy należące do struktury dokumentu, takie jak na przykład jego tytuł nie powinny być formatowane wprost lecz należy skorzystać z odpowiednich funkcji do ich wywołania. Aby zdefiniować zawartość tytułu należy użyć poleceń:

- `\title{}` – definiuje tytuł dokumentu,
- `\author{}` – definiuje autora/autorów,
- `\date{}` – zmienia datę pod tytułem.

Zadanie: zdefiniuj tytuł i autora swojego dokumentu i go skompiluj. Czy tytuł się pojawił?

Powyższe komendy definiują zawartość powyższych pól w dokumencie ale nie wstawiają samego tytułu. Poleceniem, które umieści tytuł jest komenda:

```
\maketitle
```

Tytuł pojawi się w dokumencie w miejscu wywołania tego polecenia. Komenda `\date{}` różni się od pozostałych, gdyż nawet przy braku zdefiniowanej daty zostanie wstawiona data kompilacji. Polecenia tego używa się gdy konieczna jest zmiana daty.

Zadanie: Skompiluj dokument z tytułem

3.2 Spis treści

Zadanie: Od tego momentu nie kompiluj dokumentu dopóki nie pojawi się takie polecenie w zadaniu

Jedną z głównych zalet \LaTeX a jest zautomatyzowane tworzenie list takich jak spis treści. Dostępne są następujące poziomy podziału dokumentu:

- `\part{tytuł}` – część,
- `\chapter{tytuł}` – rozdział, dostępny tylko w dokumentach typu książka (book) i raport (report),
- `\section{tytuł}` – sekcja,
- `\subsection{tytuł}` – podsekcja,
- `\subsubsection{tytuł}` – podpodsekcja,
- `\paragraph{tytuł}` – paragraf,
- `\subparagraph{tytuł}` – podparagraf.

W dokumentach typu list (letter) nie są dostępne żadne stopnie podziału.

Komendą jaka wstawia spis treści do dokumentu jest:

`\tableofcontents`

Polecenie to wstawi spis treści w miejscu w którym zostanie umieszczone w ciele dokumentu.

Zadanie: Umieść w swoim dokumencie polecenie *tableofcontents* i przepisz pierwsze dwie sekcje z dokumentu przykładowego. Dopiero wtedy skompiluj dokument. Do tytułu jednej z sekcji dopisz losową literę i skompiluj dokument jeszcze raz. Gdzie pojawiła się losowa litera dopisana do tytułu? Czy kolejna kompilacja rozwiązuje problem?

W tytułach sekcji w dokumencie przykładowym wykorzystano logo \LaTeX a. Poleceniem wstawienia takiego loga jest `\LaTeX`.

Uwaga: Nazwy poleceń w \LaTeX u są wrażliwe na wielkość liter! polecenie *LaTeX* to nie to samo co *latex*!

Zadanie: Wstaw loga \LaTeX do swoich tytułów sekcji. Czy wyglądają tak jak w dokumencie przykładowym?

\LaTeX ignoruje białe znaki po komendach. Jeżeli potrzebna jest spacja po komendzie należy użyć polecenia wstawienia spacji: „`\`”.

3.3 Otoczenia

Ważnym elementem \LaTeX a są tak zwane otoczenia. Przeważnie korzystają one z konstrukcji:

```
\begin{Nazwa_otoczenia}  
    zawartość ...  
\end{Nazwa_otoczenia}
```

Istotną cechą otoczeń jest to, że muszą się nawzajem otaczać, nie mogą się przecinać! W związku z tym niedozwolona jest na przykład taka konstrukcja:

```
\begin{Otoczenie1}  
    \begin{Otoczenie2}  
  
    \end{Otoczenie1}  
    \end{Otoczenie2}
```

Przykładem otoczenia jest środowisko do pisania list wypunktowanych:

```
\begin{itemize}  
    \item Treść punktu 1  
    \item Treść punktu 2  
\end{itemize}
```

Siostrzanym środowiskiem jest otoczenie dla list numerowanych:

```
\begin{enumerate}  
    \item Treść punktu 1  
    \item Treść punktu 2  
\end{enumerate}
```

Zadanie: Odtwórz w swoim dokumencie listę wypunktowaną z dokumentu przykładowego

3.4 Formatowanie tekstu

W dokumentach pisanych w \LaTeX u nie jest dobrą praktyką aby zmieniać lokalnie formatowanie lub rozmiar czcionek, gdyż prowadzi to do powstania niespójności stylistycznych. Zmian takich należy dokonywać na poziomie stylu całego dokumentu, na przykład poprzez zmianę rozmiaru czcionki w argumencie opcjonalnym polecenia `\documentclass[arg. opcjonalne]{typ dokumentu}`.

Czasem konieczne są jednak pewne zmiany, aby wyróżnić pewne fragmenty tekstu i \LaTeX udostępnia takie narzędzia:

- `\emph{tekst}` – wyróżnia tekst, sposób wyróżnienia zależy od stylu dokumentu,
- `\textbf{tekst}` – pogrubia czcionkę,
- `\textit{tekst}` – pochyla czcionkę,
- `\underline{tekst}` – podkreśla tekst,

Możliwa jest też lokalne zdefiniowanie rozmiaru tekstu:

- `\Huge` – tekst,
- `\huge` – tekst,
- `\LARGE` – tekst,
- `\Large` – tekst,
- `\large` – tekst,
- `\normalsize` – tekst,
- `\small` – tekst,
- `\footnotesize` – tekst,
- `\scriptsize` – tekst,
- `\tiny` – tekst,

Uwaga: Powyższe polecenia zmieniają rozmiar w całym otoczeniu w którym zostaną użyte! Aby ograniczyć ich zasięg należy umieszczać je wraz z tekstem w nawiasach klamrowych.

4 Formuły matematyczne

Język \LaTeX znany jest z bardzo rozbudowanego systemu do składu formuł matematycznych, który wykorzystywany jest w innych środowiskach, jak na przykład Matlab czy Python. Same formuły matematyczne w \LaTeX u muszą być umieszczone w odpowiednich otoczeniach matematycznych których są trzy rodzaje:

- `\sin x` – pojedyncze dolary umieszczają formułę w linii wraz z tekstem:

$\sin x$

- `$$\sin x$$` – podwójne dolary umieszczają formułę w osobnej linii:

$$\sin x$$

- `\begin{equation}\sin x\end{equation}` – otoczenie *equation* umieszcza formułę w osobnej linii i nadaje jej numer:

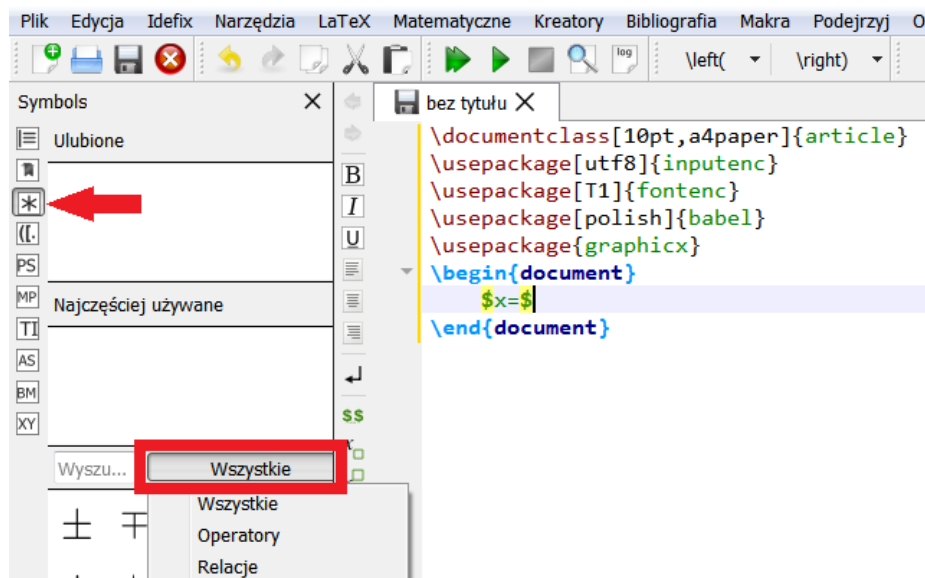
$$\sin x \tag{1}$$

Tylko otoczenia *equation* pozwalają na wykorzystanie polecenia *label* i odwołanie się do danej formuły.

Uwaga: Wewnątrz otoczeń matematycznych nie wolno zostawiać pustych linii. Może prowadzić to do nieoczekiwanych błędów!

Przy pisaniu formuł matematycznych przydatny jest przybornik z symbolami matematycznymi dostępny w programie TeXstudio. Dostęp do niego można uzyskać klikając jego ikonę na panelu bocznym, co zaprezentowano na ilustracji 6. Zakres wyświetlanych symboli można ograniczyć przyciskiem z prawej strony panelu.

Podstawowe polecenia przydatne przy pisaniu formuł matematycznych:



Ilustracja 6: Przybornik z symbolami matematycznymi w TeXstudio

- $x^{2b} = x^{2b},$
- $x_{2b} = x_{2b},$
- $\frac{a}{b} = \frac{a}{b},$
- $\int i(t) dt = \int i(t) dt,$
- $\int_{-T}^T i(t) dt = \int_{-T}^T i(t) dt.$

Zadanie: Odwzoruj pierwszy wzór matematyczny z dokumentu do odtworzenia. W jakim otoczeniu matematycznym został od zapisany?

W \LaTeX u nawiasy okrągłe i kwadratowe w formułach matematycznych można stosować wprost: $()$, $[\]$. W przypadku nawiasów klamrowych należy poprzedzić je ukośnikami wstecznymi. Na przykład $\{\}$, daje: $\{ \}$. Nawiasy takie jednak nie dostosują swoich rozmiarów do zawartości. Aby uzyskać taki efekt należy użyć poleceń \left i \right :

$$\left(\frac{a}{b} \right) = \left(\frac{a}{b} \right).$$

Polecenia te tworzą otoczenie nawiasowe, w związku z czym nie mogą się przecinać z innymi otoczeniami. Znaki nawiasów nie muszą sobie odpowiadać, można otworzyć otoczenie nawiasem okrągłym i zamknąć kwadratowym: $\left(\frac{a}{b} \right) = \left(\frac{a}{b} \right)$. Jeżeli nie chcemy aby z którejś strony pojawił się znak nawiasu należy użyć kropki, samo otoczenie musi jednak zostać zamknięte: $\left \{ \frac{a}{b} \right. = \left \{ \frac{a}{b} \right.$.

Aby odtworzyć drugi wzór z dokumentu przykładowego konieczne będzie otoczenie *array*. Służy ono do definiowania tabel w trybie matematycznym. Przyjmuje ono jeden argument obowiązkowy w którym definiuje się liczbę i wyjustowanie kolumn. W tym celu dostępne są trzy litery: *l* – kolumna wyjustowana do lewej, *c* – kolumna wyjustowana do środka, *r* – kolumna wyjustowana

do prawej. Do separacji komórek wewnątrz tabeli służy znak ampersand: „&”. Przejście do nowego wiersza jest oznaczone podwójnym ukośnikiem wstecznym: „\”. Przykładowo układ 2 na 2 elementy, wyjustowane do środka, będzie zdefiniowany następująco:

```
\begin{array}{cc}
a & b \\
c & d \\
\end{array}
```

Zadanie: Spróbuj odtworzyć wzór na macierz rotacji z dokumentu do odtworzenia.

5 Tabele i rysunki

Tabele w trybie tekstowym definiuje się za pomocą otoczenia *tabular*, które działa tak samo jak otoczenie *array* w trybie matematycznym. Linie oddzielające kolumny uzyskać można wstawiając znak kreski pionowej: „|” między literami definiującymi kolumny. Z kolei linie poziome uzyskuje się za pomocą komendy `\hline` pomiędzy wierszami:

```
\begin{tabular}{|c|c|}
\hline
a & b \\
c & d \\
\hline
\end{tabular}
```

a	b
c	d

Tabela zdefiniowana za pomocą otoczenia *tabular* jest traktowana tak jak tekst i może być wstawiona nawet w środku linii:

a	b
c	d

. Aby wyróżnić tabelę z tekstu należy umieścić ją w otoczeniu pływającym *table*:

```
\begin{table}
\begin{tabular}{|c|c|}
\hline
a & b \\
c & d \\
\hline
\end{tabular}
\end{table}
```

Zadanie: Zdefiniuj własną tabelę w otoczeniu pływającym. Gdzie się pojawiła w dokumencie po kompilacji?

Otoczenia pływające przyjmują argument opcjonalny informujący o pożądanym położeniu. Dostępne są następujące litery:

- h – w miejscu w którym jest w kodzie,
- t – u góry strony,
- b – na dole strony,
- p – na osobnej stronie z tabelami,

Powyższe argumenty są jednak jedynie sugestiami, a tabela może zostać umieszczona dowolnie. Znak wykrzyknika za parametrem położenia pozwala dodatkowo poluzować ustawienia L^AT_EXa, ale nadal nie jest bezwzględnym nakazem. Przykład:

```
\begin{table}[h!]
  \begin{tabular}{|c|c|}
    a & b \\ \hline
    c & d \\ \hline
  \end{tabular}
\end{table}
```

Zadanie: Sprawdź czy dodanie parametru położenia przeniesie Twoją tabelę w prawidłowe miejsce.

Wewnątrz otoczeń pływających, do opisu zawartości, służy polecenie `\caption{}`:

```
\begin{table}[h!]
\caption{Przykład}
  \begin{tabular}{|c|c|}
    a & b \\ \hline
    c & d \\ \hline
  \end{tabular}
\end{table}
```

Otoczeniem pływającym dla ilustracji jest otoczenie *figure*. Same ilustracje można wstawić za pomocą polecenia:

```
\includegraphics[argumenty opcjonalne]{ścieżka do pliku}|
```

Ścieżki do pliku mogą być względne lub bezwzględne. Zalecane jest umieszczanie plików z rysunkami w tym samym folderze co kompilowany plik, lub podfolderze i podanie ścieżki względnej:

```
\includegraphics{Rys/pp-putlogopelne.png}
```

Foldery oddziela się znakiem ukośnika: „/”. W celu wyjustowania rysunku do środka można użyć polecenia `\centering`, które wyśrodkuje wszystko wewnątrz otoczenia w którym zostało użyte. Aby zmienić rozmiar rysunku, należy wpisać polecenie: `width=0.5\textwidth`, które dostosuje szerokość rysunku do połowy szerokości tekstu:

```
\begin{figure}[h!]
  \centering
  \includegraphics[width=0.5\textwidth]
    {Rys/pp-putlogopelne.png}
  \caption{Przykładowy rysunek}
\end{figure}
```

Aby odwołać się w dokumencie do danego rysunku bądź tabeli pod poleceniem `\caption{}` należy umieścić polecenie `\label{etykieta}`. Następnie w miejscu w którym ma pojawić się odwołanie należy umieścić polecenie `\ref{etykieta}`.

Zadanie: Odtwórz rysunek i tabelę z dokumentu przykładowego wraz z odwołaniami.

6 Kod źródłowy w dokumentach

Aby wpisać kod źródłowy do dokumentu można skorzystać z podstawowego polecenia `\verb|kod|`. Pozwala ono na wpisanie kodu, który nie będzie interpretowany przez \LaTeX i będzie wyróżniony z tekstu. Wypisywany jest on za pomocą czcionki o stałej szerokości. Komenda ta pozwala jednak na wstawienie tylko pojedynczej linii kodu. W przypadku bardziej rozbudowanego źródła, lub potrzeby kolorowania składni, dodać należy pakiet `\usepackage{listings}`. Język dla którego ma być kolorowana składnia można ustawić za pomocą polecenia: `\lstset{language=C++}` lub w argumentach opcjonalnych otoczenia z wstawianym kodem:

```
\begin{lstlisting}[language=C++]
for(int i=0;i<10;i++){
\end{lstlisting}
```

Możliwe jest też wstawienie do dokumentu całego pliku z kodem źródłowym:

```
\lstinputlisting[language=C++] {plik.cpp}
```

Zadanie: Przepisz kod źródłowy z przykładowego dokumentu, wraz z właściwym kolorowaniem, składni.

Literatura

- [1] Dokumentacja portalu Overleaf. https://www.overleaf.com/learn/latex/Main_Page.
- [2] *LaTeX w Wikibooks*. <https://pl.wikibooks.org/wiki/LaTeX>.
- [3] Tobias Oetiker et al. *A (Not So) Short Introduction to $\text{\LaTeX} 2_{\epsilon}$* , 2018. <https://ctan.org/pkg/lshort-english>.
- [4] Tobias Oetiker i inni. *Nie za krótkie wprowadzenie do systemu $\text{\LaTeX} 2_{\epsilon}$* , 2007. <https://ctan.org/tex-archive/info/lshort/polish/>.