

Edycja tekstu w programie L^AT_EX— definiowanie poleceń i otoczeń

JOLANTA MARZEC

13 listopada 2017

1 Definiowanie poleceń

Jeśli jakaś komenda pojawia się często w tekście, warto zdefiniować jej krótszy odpowiednik. Taką definicję wstawiamy w *preamble*. Do skorzystania z poniższych komend będzie potrzebny pakiet `amssymb`.

1. Zdefiniuj polecenie `\R` komendą `\newcommand{\R}{\mathbb{R}}`. Sprawdź działanie zdefiniowanego polecenia wpisując `\R` (w trybie matematycznym, np. w $\$$).
2. Dodaj polecenie `\newcommand{\Rm}{\mathbb{R}}` i sprawdź, że tym razem `\Rm` działa tylko w trybie tekstowym.
3. Aby korzystać z jakiegoś polecenia zarówno w trybie matematycznym jak i tekstowym, w definicji należy użyć `\ensuremath`. W *preamble* dodaj polecenie `\newcommand{\Q}{\ensuremath{\mathbb{Q}}}` i sprawdź, że oba kody `\Q` i `\Q` dają ten sam wynik.
4. Dodaj polecenie `\newcommand{\ciag}[1]{\langle \#1 \rangle}` i sprawdź jego działanie wpisując `\ciag{x}`.
5. Dodaj polecenie `\newcommand{\mat}[4]{\left(\begin{matrix} \#1 & \#2 \\ \#3 & \#4 \end{matrix}\right)}` i sprawdź jego działanie wpisując `\mat{a}{b}{c}{d}` (w trybie matematycznym).
6. Sprawdź działanie komend `\Re x` i `\Im x`. Aby uzyskać powszechnie stosowany zapis można zmienić działanie tych komend poprzez podanie nowej definicji: `\renewcommand{\Re}{\mathrm{Re}}` i `\renewcommand{\Im}{\mathrm{Im}}`. Sprawdź ponownie działanie komend `\Re x` i `\Im x`.

2 Definiowanie otoczeń

Podobnie jak definiujemy nowe polecenia, można również definiować nowe otoczenia. Do tego służy komenda

```
\newenvironment{nazwa otoczenia}[ilość argumentów]{kod początkowy}{kod końcowy}
```

1. Zdefiniuj nowe otoczenie poprzez `\newenvironment{cw}{\noindent\textbf{Ćwiczenie}}{\begin{itshape}}{\end{itshape}}`

Sprawdź jego działanie wpisując kod

```
\begin{cw}
```

Przykładowy tekst pisany kursywą.

```
\end{cw}
```

2. Powyższy przykład można zmodyfikować poprzez dodanie dodatkowego argumentu, np. nazwy ćwiczenia:

```
\newenvironment{cw}[1]{\noindent\textbf{Ćwiczenie (#1)}}{\begin{itshape}}{\end{itshape}}
```

Aby przywołać ten argument, wpisujemy go w nawiasy klamrowe, np.:

```
\begin{cw}{Kursywa}
```

Przykładowy tekst pisany kursywą.

```
\end{cw}
```

3. Numeracja nowych otoczeń. Najpierw (w preambule) musimy zdefiniować licznik otoczenia, najlepiej przez użycie tej samej nazwy jak nazwa otoczenia, np. `\newcounter{cw}`. Następnie modyfikujemy definicję otoczenia:

```
\newenvironment{cw}[1]{
\noindent\refstepcounter{cw}{\textbf{Ćwiczenie \thecw\, (#1)}}{\begin{itshape}}
{\end{itshape}}
```

Użycie komendy `\refstepcounter{}` pozwala na późniejsze odwoływanie się do tego numeru poprzez `\label{}`. Komenda `\thecw` odpowiada za przywołanie odpowiedniego numeru. Jeśli chcemy użyć innej numeracji, np. pochodzącej z otoczenia *theorem* powinniśmy napisać odpowiednio `\refstepcounter{theorem}` i `\thetheorem`.

4. Definiowanie nowych list. Zdefiniuj licznik `\newcounter{lista}` i sprawdź działanie definicji

```
\newenvironment{lista}[1]{
\begin{list}{#1{lista}}
{\usecounter{lista}\setlength{\rightmargin}{\leftmargin}}}
{\end{list}\setcounter{lista}{0}}
```

wpisując kod

```
\begin{lista}{\roman}
\item Ten składnik jest numerowany w systemie rzymskim.
\item Ten też.
\end{lista}
```

Zamiast komendy `\roman` można też użyć `\arabic`, `\Roman` oraz `\alph`, lub je zmodyfikować do swoich potrzeb, np. poprzez `\bfseries\large\alph`.