Politechnika Warszawska
Wydział Matematyki i Nauk Informacyjnych

Beamer,
czyli prezentacje w LaTeX-u

Marek Gągolewski
M.Gagolewski@mini.pw.edu.pl

Warszawa, 1 października 2016 r.

Beamer

Cóż to takiego?

Beamer (niem. *der Beamer*, *-s*) to najpopularniejszy LaTeX-owy pakiet do tworzenia pięknych prezentacji.

Dowiedz się więcej

www.gagolewski.com/teaching/tutorials/beamer/

Beamer

Cóż to takiego?

Beamer (niem. *der Beamer*, *-s*) to najpopularniejszy LaTeX-owy pakiet do tworzenia pięknych prezentacji.

Dowiedz się więcej

`www.gagolewski.com/teaching/tutorials/beamer/`

Plan prezentacji

1. Jak zacząć?
   - Ustawienia Beamera
   - Ramki
   - Bloki
   - Pauzy
2. Ściągawka z LaTeX-a
   - Wzory
   - Twierdzenia i definicje
   - Listy
   - Tabele
   - Rysunki
   - Kody źródłowe

Jak zacząć?

Prezentacje — na co zwracamy uwagę:

1. temat,
2. struktura, forma, treść,
3. wygląd,
4. sposób przedstawienia.

Uwaga

Dzisiaj zajmiemy się tylko przygotowaniem prezentacji **pod względem wizualnym**.

Inne aspekty są nie mniej ważne!

Jak zacząć?

Prezentacje — na co zwracamy uwagę:

1. temat,
2. struktura, forma, treść,
3. wygląd,
4. sposób przedstawienia.

Uwaga

Dzisiaj zajmiemy się tylko przygotowaniem prezentacji **pod względem wizualnym**.

Inne aspekty są nie mniej ważne!

Jak zacząć?

Prezentacje — na co zwracamy uwagę:

1. temat,
2. struktura, forma, treść,
3. wygląd,
4. sposób przedstawienia.

Uwaga

Dzisiaj zajmiemy się tylko przygotowaniem prezentacji **pod względem wizualnym**.

Inne aspekty są nie mniej ważne!

Jak zacząć?

Prezentacje — na co zwracamy uwagę:

1. temat,
2. struktura, forma, treść,
3. wygląd,
4. sposób przedstawienia.

Uwaga

Dzisiaj zajmiemy się tylko przygotowaniem prezentacji **pod względem wizualnym**.

Inne aspekty są nie mniej ważne!

Jak zacząć?

Prezentacje — na co zwracamy uwagę:

1. temat,
2. struktura, forma, treść,
3. wygląd,
4. sposób przedstawienia.

Uwaga

Dzisiaj zajmiemy się tylko przygotowaniem prezentacji **pod względem wizualnym**.

Inne aspekty są nie mniej ważne!

Jak zacząć?

Prezentacje — na co zwracamy uwagę:

1. temat,
2. struktura, forma, treść,
3. wygląd,
4. sposób przedstawienia.

Uwaga

Dzisiaj zajmiemy się tylko przygotowaniem prezentacji **pod względem wizualnym**.

Inne aspekty są nie mniej ważne!

Instalacja

Uwaga

Beamer powinien być zainstalowany wraz z LaTeX-em.
Jeśli tak nie jest, pobieramy go ze strony:
`https://bitbucket.org/rivanvx/beamer/downloads`

Przyda nam się także dobry edytor plików LaTeX-a, np.

1. *Kile* pod Linuxem,
2. *TeXnicCenter*,
3. *RStudio*,
4. *TeXmaker*

Najważniejsze ułatwienia:

1. automatyczne przeładowanie pliku PDF po kompilacji (~~*Adobe Reader*~~),
2. sprawdzanie pisowni,
3. kompilacja pliku za pomocą skrótu klawiszowego.

Instalacja

Uwaga

Beamer powinien być zainstalowany wraz z LaTeX-em.
Jeśli tak nie jest, pobieramy go ze strony:
`https://bitbucket.org/rivanvx/beamer/downloads`

Przyda nam się także dobry edytor plików LaTeX-a, np.

1. *Kile* pod Linuxem,
2. *TeXnicCenter*,
3. *RStudio*,
4. *TeXmaker*

Najważniejsze ułatwienia:

1. automatyczne przeładowanie pliku PDF po kompilacji (~~*Adobe Reader*~~),
2. sprawdzanie pisowni,
3. kompilacja pliku za pomocą skrótu klawiszowego.

Instalacja

Uwaga

Beamer powinien być zainstalowany wraz z LaTeX-em.
Jeśli tak nie jest, pobieramy go ze strony:
`https://bitbucket.org/rivanvx/beamer/downloads`

Przyda nam się także dobry edytor plików LaTeX-a, np.

1. *Kile* pod Linuxem,
2. *TeXnicCenter*,
3. *RStudio*,
4. *TeXmaker*

Najważniejsze ułatwienia:

1. automatyczne przeładowanie pliku PDF po kompilacji (~~*Adobe Reader*~~),
2. sprawdzanie pisowni,
3. kompilacja pliku za pomocą skrótu klawiszowego.

Instalacja

Uwaga

Beamer powinien być zainstalowany wraz z LaTeX-em.
Jeśli tak nie jest, pobieramy go ze strony:
`https://bitbucket.org/rivanvx/beamer/downloads`

Przyda nam się także dobry edytor plików LaTeX-a, np.

1. *Kile* pod Linuxem,
2. *TeXnicCenter*,
3. *RStudio*,
4. *TeXmaker*

Najważniejsze ułatwienia:

1. automatyczne przeładowanie pliku PDF po kompilacji (~~*Adobe Reader*~~),
2. sprawdzanie pisowni,
3. kompilacja pliku za pomocą skrótu klawiszowego.

Instalacja

Uwaga

Beamer powinien być zainstalowany wraz z LaTeX-em.
Jeśli tak nie jest, pobieramy go ze strony:
`https://bitbucket.org/rivanvx/beamer/downloads`

Przyda nam się także dobry edytor plików LaTeX-a, np.

1. *Kile* pod Linuxem,
2. *TeXnicCenter*,
3. *RStudio*,
4. *TeXmaker*

Najważniejsze ułatwienia:

1. automatyczne przeładowanie pliku PDF po kompilacji (~~*Adobe Reader*~~),
2. sprawdzanie pisowni,
3. kompilacja pliku za pomocą skrótu klawiszowego.

Instalacja

Uwaga

Beamer powinien być zainstalowany wraz z LaTeX-em.
Jeśli tak nie jest, pobieramy go ze strony:
`https://bitbucket.org/rivanvx/beamer/downloads`

Przyda nam się także dobry edytor plików LaTeX-a, np.

1. *Kile* pod Linuxem,
2. *TeXnicCenter*,
3. *RStudio*,
4. *TeXmaker*

Najważniejsze ułatwienia:

1. automatyczne przeładowanie pliku PDF po kompilacji (~~*Adobe Reader*~~),
2. sprawdzanie pisowni,
3. kompilacja pliku za pomocą skrótu klawiszowego.

Instalacja

Uwaga

Beamer powinien być zainstalowany wraz z LaTeX-em.
Jeśli tak nie jest, pobieramy go ze strony:
`https://bitbucket.org/rivanvx/beamer/downloads`

Przyda nam się także dobry edytor plików LaTeX-a, np.

1. *Kile* pod Linuxem,
2. *TeXnicCenter*,
3. *RStudio*,
4. *TeXmaker*

Najważniejsze ułatwienia:

1. automatyczne przeładowanie pliku PDF po kompilacji (~~*Adobe Reader*~~),
2. sprawdzanie pisowni,
3. kompilacja pliku za pomocą skrótu klawiszowego.

Plik źródłowy .tex

```
1  \documentclass[12pt,pdftex]{beamer}
2
3  \usepackage[T1]{polski}
4  \usepackage[polish]{babel}
5  \usepackage[utf8]{inputenc}
6  \usepackage[T1]{fontenc}
7
8  % LADOWANIE PAKIETOW DODATKOWYCH...
9  % USTAWIENIA WYGLADU...
10
11 \begin{document}
12
13     % "RAMKI"...
14
15 \end{document}
```

Kompilacja

Pliki źródłowe polecam kompilować programem `pdflatex` (zob. dalej).

Plik źródłowy .tex

```
1  \documentclass[12pt,pdftex]{beamer}
2
3  \usepackage[T1]{polski}
4  \usepackage[polish]{babel}
5  \usepackage[utf8]{inputenc}
6  \usepackage[T1]{fontenc}
7
8  % LADOWANIE PAKIETOW DODATKOWYCH...
9  % USTAWIENIA WYGLADU...
10
11 \begin{document}
12
13     % "RAMKI"...
14
15 \end{document}
```

Kompilacja

Pliki źródłowe polecam kompilować programem `pdflatex` (zob. dalej).

Wystroje (*themes*)

Dostosowywanie wyglądu

Beamer udostępnia wiele gotowych szablonów odpowiadających za wygląd prezentacji.

Wśród nich wyróżniamy m.in. schematy dla:

- układu slajdów — \usetheme[*opcje*]{*nazwa*},
- kolorów — \usecolortheme[*opcje*]{*nazwa*},
- czcionek — \usefonttheme[*opcje*]{*nazwa*}.

Wystroje (*themes*)

Dostosowywanie wyglądu

Beamer udostępnia wiele gotowych szablonów odpowiadających za wygląd prezentacji.

Wśród nich wyróżniamy m.in. schematy dla:

- układu slajdów — **usetheme**[*opcje*]{*nazwa*},
- kolorów — **usecolortheme**[*opcje*]{*nazwa*},
- czcionek — **usefonttheme**[*opcje*]{*nazwa*}.

Wystroje (*themes*)

Moje ulubione ustawienia:

```
\usetheme{Warsaw}
\usecolortheme{whale}
\useoutertheme{infolines}
\useinnertheme{circles}
\usefonttheme{professionalfonts}

\setbeamertemplate{navigation symbols}{}
\setbeamercovered{transparent}

\setbeamertemplate{theorems}[numbered]
```

Wykaz dostępnych wystrojów

Zobacz: `http://www.hartwork.org/beamer-theme-matrix/`.

Ciekawostka

Większość osób wykorzysta (niestety) udostępniony przeze mnie szablon.

Wystroje (*themes*)

Moje ulubione ustawienia:

```
\usetheme{Warsaw}
\usecolortheme{whale}
\useoutertheme{infolines}
\useinnertheme{circles}
\usefonttheme{professionalfonts}

\setbeamertemplate{navigation symbols}{}
\setbeamercovered{transparent}

\setbeamertemplate{theorems}[numbered]
```

Wykaz dostępnych wystrojów

Zobacz: `http://www.hartwork.org/beamer-theme-matrix/`.

Ciekawostka

Większość osób wykorzysta (niestety) udostępniony przeze mnie szablon.

Wystroje (*themes*)

Moje ulubione ustawienia:

```
\usetheme{Warsaw}
\usecolortheme{whale}
\useoutertheme{infolines}
\useinnertheme{circles}
\usefonttheme{professionalfonts}

\setbeamertemplate{navigation symbols}{}
\setbeamercovered{transparent}

\setbeamertemplate{theorems}[numbered]
```

Wykaz dostępnych wystrojów

Zobacz: `http://www.hartwork.org/beamer-theme-matrix/`.

Ciekawostka

Większość osób wykorzysta (niestety) udostępniony przeze mnie szablon.

Ramki (*frames*)

Podstawową jednostką tworzonej przez nas prezentacji są tzw. **ramki**.

Definicje wszystkich ramek powinny być zawarte w środowisku *document*.

Uwaga

Jedna ramka może odpowiadać za stworzenie więcej niż jednego slajdu.

```
\begin{frame}
  \frametitle{Tytul ramki} % MOZNA POMINAC

  % ... TRESC ...

\end{frame}
```

Ramki (*frames*)

Podstawową jednostką tworzonej przez nas prezentacji są tzw. **ramki**.

Definicje wszystkich ramek powinny być zawarte w środowisku *document*.

Uwaga

Jedna ramka może odpowiadać za stworzenie więcej niż jednego slajdu.

```
\begin{frame}
  \frametitle{Tytul ramki} % MOZNA POMINAC

  % ... TRESC ...

\end{frame}
```

Ramki (*frames*)

Podstawową jednostką tworzonej przez nas prezentacji są tzw. **ramki**.

Definicje wszystkich ramek powinny być zawarte w środowisku *document*.

Uwaga

Jedna ramka może odpowiadać za stworzenie więcej niż jednego slajdu.

```
1 \begin{frame}
2   \frametitle{Tytul ramki} % MOZNA POMINAC
3
4   % ... TRESC ...
5
6 \end{frame}
```

Ramka tytułowa

Oto przykład kodu tworzącego ramkę tytułową.

```
\thispagestyle{empty}
\begin{frame}
  \title[Dlaczego istnieje\dots]%
  {Dlaczego istnieje raczej cos niz nic?}
  \author[G. R. Brzeczyszczykiewicz]%
  {Grzegorz Romuald Brzeczyszczykiewicz}
  \institute[IPP]%
  {Instytut Problemow Pryncypialnych}
  \date[\today]%
  {Seminarium \textit{XYZ},\\
     Warszawa, \today}

  \titlepage
\end{frame}
```

Dlaczego jest raczej coś niż nic?

Grzegorz Romuald Brzęczyszczykiewicz

Instytut Problemów Pryncypialnych

Seminarium *TakCiekaweŻeAżStrach*,
Warszawa, 1 października 2016

Spis treści

Organizacja prezentacji

Prezentacje w Beamerze możemy (i powinniśmy) organizować tak, jak każdy inny dokument w LaTeX-u.

W tym celu posługujemy się instrukcjami \section{...} i \subsection{...}.

Dzięki temu możemy m.in. wygenerować spis treści.

```
1 \section*{Plan prezentacji}
2 \begin{frame}{Plan prezentacji}
3
4   \tableofcontents
5
6 \end{frame}
```

Spis treści

Organizacja prezentacji

Prezentacje w Beamerze możemy (i powinniśmy) organizować tak, jak każdy inny dokument w LaTeX-u.

W tym celu posługujemy się instrukcjami \section{...} i \subsection{...}.

Dzięki temu możemy m.in. wygenerować spis treści.

```
1  \section*{Plan prezentacji}
2  \begin{frame}{Plan prezentacji}
3
4    \tableofcontents
5
6  \end{frame}
```

Plan prezentacji

1. Jak zacząć?
   - Ustawienia Beamera
   - Ramki
   - Bloki
   - Pauzy
2. Ściągawka z LaTeX-a
   - Wzory
   - Twierdzenia i definicje
   - Listy
   - Tabele
   - Rysunki
   - Kody źródłowe

Bloki (*blocks*)

Treść

Ramki wypełniać można dowolną treścią (zwykły kod w LaTeX-u).

Dla zwiększenia estetyki warto czasem stworzyć **blok**.

```
\begin{block}{Tresc}
    Ramki wypelniac mozna dowolna trescia...

    \bigskip
    Dla zwiekszenia estetyki warto...
\end{block}
```

Płachta na byka

A tutaj zamiast środowiska *block* używamy *alertblock*.

Bloki (*blocks*)

Treść

Ramki wypełniać można dowolną treścią (zwykły kod w LaTeX-u).

Dla zwiększenia estetyki warto czasem stworzyć **blok**.

```
\begin{block}{Tresc}
    Ramki wypelniac mozna dowolna trescia ...

    \bigskip
    Dla zwiekszenia estetyki warto ...
\end{block}
```

Płachta na byka

A tutaj zamiast środowiska *block* używamy *alertblock*.

Bloki (*blocks*)

Treść

Ramki wypełniać można dowolną treścią (zwykły kod w LaTeX-u).

Dla zwiększenia estetyki warto czasem stworzyć **blok**.

```
1 \begin{block}{Tresc}
2     Ramki wypelniac mozna dowolna trescia ...
3
4     \bigskip
5     Dla zwiekszenia estetyki warto ...
6 \end{block}
```

Płachta na byka

A tutaj zamiast środowiska *block* używamy *alertblock*.

Pauzy

Podział ramki na wiele slajdów

Polecenie \pause służy do podziału ramki na więcej niż jeden slajd.

A po co?

Dzięki temu możemy „odkrywać” kolejne fragmenty przed słuchaczami.

```
1 \begin{block}{Podzial ramki na wiele slajdow}
2     Polecenie \pause...
3 \end{block}
4
5 \pause\bigskip
6 \begin{block}{A po co?}
7     Dzieki temu...
8 \end{block}
```

Pauzy

Podział ramki na wiele slajdów

Polecenie \pause służy do podziału ramki na więcej niż jeden slajd.

A po co?

Dzięki temu możemy „odkrywać” kolejne fragmenty przed słuchaczami.

```
1 \begin{block}{Podzial ramki na wiele slajdow}
2     Polecenie \pause...
3 \end{block}
4
5 \pause\bigskip
6 \begin{block}{A po co?}
7     Dzieki temu...
8 \end{block}
```

Pauzy

Podział ramki na wiele slajdów

Polecenie \pause służy do podziału ramki na więcej niż jeden slajd.

A po co?

Dzięki temu możemy „odkrywać" kolejne fragmenty przed słuchaczami.

```
1 \begin{block}{Podzial ramki na wiele slajdow}
2     Polecenie \pause ...
3 \end{block}
4
5 \pause\bigskip
6 \begin{block}{A po co?}
7     Dzieki temu ...
8 \end{block}
```

Ściągawka z LaTeX-a

1 Jak zacząć?
- Ustawienia Beamera
- Ramki
- Bloki
- Pauzy

2 Ściągawka z LaTeX-a
- Wzory
- Twierdzenia i definicje
- Listy
- Tabele
- Rysunki
- Kody źródłowe

Wzory

Równania numerowane

Środowisko *equation* tworzy równania numerowane, np.

$$f(m) = \min_{i=1,\dots,n} g_i(m). \tag{1}$$

```
1 \begin{equation}
2     f(m) = \min_{i=1,\dots,n}   g_i(m).
3 \end{equation}
```

Uwaga: powyżej mamy $\min$, a nie min... Zawsze można napisać `$\mathrm{min}$`...

Wzory

Równania nienumerowane

Środowisko *equation** tworzy równania nienumerowane, np.

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(- \frac{(x-\mu)^2}{2\sigma^2} \right).$$

```
1 \begin{equation*}
2     f(x) = \tfrac{1}{\sqrt{2\pi\sigma^2}}\;
3         \exp\Big(-\tfrac{(x-\mu)^2}{2\sigma^2}\Big).
4 \end{equation*}
```

Uwaga: powyżej mamy $\exp$, a nie exp...

Wzory inline

Oczywiście pamiętamy, że wzory w treści tekstu, np. $\widehat{\zeta}_0^*(x)$, generujemy przy użyciu \$...\$.

Wzory

Równania nienumerowane

Środowisko *equation** tworzy równania nienumerowane, np.

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\Big(- \frac{(x-\mu)^2}{2\sigma^2} \Big).$$

```
\begin{equation*}
    f(x) = \tfrac{1}{\sqrt{2\pi\sigma^2}}\;
        \exp\Big(-\tfrac{(x-\mu)^2}{2\sigma^2}\Big).
\end{equation*}
```

Uwaga: powyżej mamy $\exp$, a nie exp...

Wzory `inline`

Oczywiście pamiętamy, że wzory w treści tekstu, np. $\widehat{\zeta}_0^*(x)$, generujemy przy użyciu \$...\$.

Twierdzenia i definicje

Bloki

Twierdzenia i definicje możemy podawać w blokach tworzonych ręcznie.

Własne środowiska

Wygodniej i estetyczniej jest jednak samodzielnie zdefiniować środowiska do obsługi tego typu obiektów.

```
\newtheorem{twierdzenie}{Twierdzenie}
\renewcommand{\proofname}{Dowod}
\newtheorem{lemat}[twierdzenie]{Lemat}
\newtheorem{wniosek}[twierdzenie]{Wniosek}
\newtheorem{stwierdzenie}[twierdzenie]{Stwierdzenie}

\theoremstyle{definition}
\newtheorem*{definicja}{Definicja}
\newtheorem*{oznaczenie}{Oznaczenie}
```

Twierdzenia i definicje

Bloki

Twierdzenia i definicje możemy podawać w blokach tworzonych ręcznie.

Własne środowiska

Wygodniej i estetyczniej jest jednak samodzielnie zdefiniować środowiska do obsługi tego typu obiektów.

```
\newtheorem{twierdzenie}{Twierdzenie}
\renewcommand{\proofname}{Dowod}
\newtheorem{lemat}[twierdzenie]{Lemat}
\newtheorem{wniosek}[twierdzenie]{Wniosek}
\newtheorem{stwierdzenie}[twierdzenie]{Stwierdzenie}

\theoremstyle{definition}
\newtheorem*{definicja}{Definicja}
\newtheorem*{oznaczenie}{Oznaczenie}
```

Twierdzenia i definicje

Bloki

Twierdzenia i definicje możemy podawać w blokach tworzonych ręcznie.

Własne środowiska

Wygodniej i estetyczniej jest jednak samodzielnie zdefiniować środowiska do obsługi tego typu obiektów.

```
1 \newtheorem{twierdzenie}{Twierdzenie}
2 \renewcommand{\proofname}{Dowod}
3 \newtheorem{lemat}[twierdzenie]{Lemat}
4 \newtheorem{wniosek}[twierdzenie]{Wniosek}
5 \newtheorem{stwierdzenie}[twierdzenie]{Stwierdzenie}
6
7 \theoremstyle{definition}
8 \newtheorem*{definicja}{Definicja}
9 \newtheorem*{oznaczenie}{Oznaczenie}
```

Twierdzenia i definicje
Przykład

Definicja

Funkcję $f : \mathbb{R} \to \mathbb{R}$ nazywamy **straszną**, jesli $\lim_{x\to\infty} f(x) = \infty$.

Twierdzenie 1

Dla każdej strasznej funkcji f zachodzi $(\forall y)\ f(y) = 77$ lub $f(y) \neq 77$.

Dowód.
Przecież to widać. A poza tym raczej nie umieszczamy dowodów na slajdach, chyba że chcemy uśpić słuchaczy. Albo się popisać. Co należało pokazać. □

Twierdzenia i definicje
Przykład

Definicja

Funkcję $f : \mathbb{R} \to \mathbb{R}$ nazywamy **straszną**, jesli $\lim_{x\to\infty} f(x) = \infty$.

Twierdzenie 1

Dla każdej strasznej funkcji f zachodzi $(\forall y)\ f(y) = 77$ lub $f(y) \neq 77$.

Dowód.
Przecież to widać. A poza tym raczej nie umieszczamy dowodów na slajdach, chyba że chcemy uśpić słuchaczy. Albo się popisać. Co należało pokazać. □

Twierdzenia i definicje

Przykład

Definicja

Funkcję $f : \mathbb{R} \to \mathbb{R}$ nazywamy **straszną**, jesli $\lim_{x\to\infty} f(x) = \infty$.

Twierdzenie 1

Dla każdej strasznej funkcji f zachodzi $(\forall y)\ f(y) = 77$ lub $f(y) \neq 77$.

Dowód.
Przecież to widać. A poza tym raczej nie umieszczamy dowodów na slajdach, chyba że chcemy uśpić słuchaczy. Albo się popisać. Co należało pokazać. $\square$

Twierdzenia i definicje

Przykład

Kod źródłowy:

```
\begin{definicja}
    Funkcje $f:\mathbb{R}\to\mathbb{R}$ nazywamy
    \emph{straszna}, je li $\lim_{x\to\infty} f(x)=\infty$.
\end{definicja}

\bigskip\pause
\begin{twierdzenie}
    Dla kazdej strasznej funkcji $f$ zachodzi
    $(\forall y)$ $f(y)=77 \text{ lub } f(y)\neq 77$.
\end{twierdzenie}

\bigskip\pause
\textbf{Dow d}.

Cos tam cos tam\dots
\hfill$\boxdot$
```

Listy

- Listę **wypunktowaną**
- tworzymy przy użyciu
- środowiska *itemize*.

```
\begin{itemize}
        \item Liste \emph{wypunktowana}
  \pause\item tworzymy przy uzyciu
  \pause\item srodowiska itemize.
\end{itemize}
```

Uwaga

W dokumentacji Beamera można znaleźć inny, być może wygodniejszy sposób, „pauzowania” elementów listy.

Listy

- Listę **wypunktowaną**
- tworzymy przy użyciu
- środowiska *itemize*.

```
\begin{itemize}
        \item Liste \emph{wypunktowana}
  \pause\item tworzymy przy uzyciu
  \pause\item srodowiska itemize.
\end{itemize}
```

Uwaga

W dokumentacji Beamera można znaleźć inny, być może wygodniejszy sposób, „pauzowania” elementów listy.

Listy

- Listę **wypunktowaną**
- tworzymy przy użyciu
- środowiska *itemize*.

```
\begin{itemize}
        \item Liste \emph{wypunktowana}
  \pause\item tworzymy przy uzyciu
  \pause\item srodowiska itemize.
\end{itemize}
```

Uwaga

W dokumentacji Beamera można znaleźć inny, być może wygodniejszy sposób, „pauzowania” elementów listy.

Listy

- Listę **wypunktowaną**
- tworzymy przy użyciu
- środowiska *itemize*.

```
1 \begin{itemize}
2         \item Liste \emph{wypunktowana}
3   \pause\item tworzymy przy uzyciu
4   \pause\item srodowiska itemize.
5 \end{itemize}
```

Uwaga

W dokumentacji Beamera można znaleźć inny, być może wygodniejszy sposób, „pauzowania” elementów listy.

Listy

- Listę **wypunktowaną**
- tworzymy przy użyciu
- środowiska *itemize*.

```
1 \begin{itemize}
2         \item Liste \emph{wypunktowana}
3   \pause\item tworzymy przy uzyciu
4   \pause\item srodowiska itemize.
5 \end{itemize}
```

Uwaga

W dokumentacji Beamera można znaleźć inny, być może wygodniejszy sposób, „pauzowania” elementów listy.

Listy

1. Listę **numerowaną**
2. tworzymy przy użyciu
3. środowiska *enumerate*.

```
\begin{enumerate}
        \item Liste \emph{numerowana}
  \pause\item tworzymy przy uzyciu
  \pause\item srodowiska enumerate.
\end{enumerate}
```

Zagłębianie

Nic nie stoi na przeszkodzie, by stworzyć listę w liście.

Listy

1. Listę **numerowaną**
2. tworzymy przy użyciu
3. środowiska *enumerate*.

```
\begin{enumerate}
        \item Liste \emph{numerowana}
  \pause\item tworzymy przy uzyciu
  \pause\item srodowiska enumerate.
\end{enumerate}
```

Zagłębianie

Nic nie stoi na przeszkodzie, by stworzyć listę w liście.

Listy

1. Listę **numerowaną**
2. tworzymy przy użyciu
3. środowiska *enumerate*.

```
\begin{enumerate}
        \item Liste \emph{numerowana}
  \pause\item tworzymy przy uzyciu
  \pause\item srodowiska enumerate.
\end{enumerate}
```

Zagłębianie

Nic nie stoi na przeszkodzie, by stworzyć listę w liście.

Listy

1. Listę **numerowaną**
2. tworzymy przy użyciu
3. środowiska *enumerate*.

```
1 \begin{enumerate}
2         \item Liste \emph{numerowana}
3   \pause\item tworzymy przy uzyciu
4   \pause\item srodowiska enumerate.
5 \end{enumerate}
```

Zagłębianie

Nic nie stoi na przeszkodzie, by stworzyć listę w liście.

Listy

1. Listę **numerowaną**
2. tworzymy przy użyciu
3. środowiska *enumerate*.

```
1 \begin{enumerate}
2         \item Liste \emph{numerowana}
3   \pause\item tworzymy przy uzyciu
4   \pause\item srodowiska enumerate.
5 \end{enumerate}
```

Zagłębianie

Nic nie stoi na przeszkodzie, by stworzyć listę w liście.

Tabele

Tablica: Wartości krytyczne

A	B	C	D	E
X	1	2	3	4
Y	3	4	5	6

```
\begin{table}\centering
    \caption{Wartosci krytyczne}
    \begin{tabular}{|c||cccc|}
    \hline
    A & B & C & D & E \\
    \hline\hline
    X & 1 & 2 & 3 & 4 \\
    Y & 3 & 4 & 5 & 6 \\
    \hline
    \end{tabular}
\end{table}
```

Tabele

Tablica: Wartości krytyczne

A	B	C	D	E
X	1	2	3	4
Y	3	4	5	6

```
\begin{table}\centering
    \caption{Wartosci krytyczne}
    \begin{tabular}{|c||cccc|}
    \hline
    A & B & C & D & E \\
    \hline\hline
    X & 1 & 2 & 3 & 4 \\
    Y & 3 & 4 & 5 & 6 \\
    \hline
    \end{tabular}
\end{table}
```

Tabele

Dane

W programie R nietrudno jest napisać funkcję, która służy do wypisywania na ekran kodu tabeli LaTeX-owej. Dzięki temu można w swej pracy ładnie formatować np. ramki danych (*data frames*).

Przykładowy kod:

```
latexprintdf <- function(dane, form) {
    stopifnot(ncol(dane) == length(form))
    for (i in 1:nrow(dane)) {
        for (j in 1:ncol(dane)) {
            if (!is.na(dane[i,j]))
                cat(sprintf(form[j], dane[i,j]))
            if (j < ncol(dane)) cat(" & ") else cat(" \\\\\n")
        }
    }
}
```

Tabele

Dane

W programie R nietrudno jest napisać funkcję, która służy do wypisywania na ekran kodu tabeli LaTeX-owej. Dzięki temu można w swej pracy ładnie formatować np. ramki danych (*data frames*).

Przykładowy kod:

```
1  latexprintdf <- function(dane, form) {
2      stopifnot(ncol(dane) == length(form))
3      for (i in 1:nrow(dane)) {
4          for (j in 1:ncol(dane)) {
5              if (!is.na(dane[i,j]))
6                  cat(sprintf(form[j], dane[i,j]))
7              if (j < ncol(dane)) cat(" & ") else cat(" \\\\\n")
8          }
9      }
10 }
```

Tabele

Przykładowe użycie:

```
> dane <- data.frame(x=1:5, y=pi/(1:5))
> form <- c("%g", "%.3f")
> latexprintdf(dane, form)
1 & 3.142 \\
2 & 1.571 \\
3 & 1.047 \\
4 & 0.785 \\
5 & 0.628 \\
```

zob. też funkcję `knitr::kable()`.

Rysunki
Formaty plików

`pdflatex`

`pdflatex` obsługuje pliki graficzne m.in. w formatach PDF i JPEG.

Format pliku

Nie zapisujmy wykresów w formacie rastrowym (np. JPEG), gdyż będą **brzydko** wyglądać.

`latex`

Zwykły kompilator `latex` obsługuje pliki graficzne typu PostScript.

Rysunki
Formaty plików

`pdflatex`

`pdflatex` obsługuje pliki graficzne m.in. w formatach PDF i JPEG.

Format pliku

Nie zapisujmy wykresów w formacie rastrowym (np. JPEG), gdyż będą **brzydko** wyglądać.

`latex`

Zwykły kompilator `latex` obsługuje pliki graficzne typu PostScript.

Rysunki
Formaty plików

`pdflatex`

`pdflatex` obsługuje pliki graficzne m.in. w formatach PDF i JPEG.

Format pliku

Nie zapisujmy wykresów w formacie rastrowym (np. JPEG), gdyż będą **brzydko** wyglądać.

`latex`

Zwykły kompilator `latex` obsługuje pliki graficzne typu PostScript.

Rysunki

Przykład: generowanie wykresu w R

Fragment kodu w R:

```
pdf("nazwapliku.pdf", height=8, width=5)
plot(...)
dev.off()
```

Dodawanie ramki z rysunkiem:

```
\begin{frame}
  \begin{center}
    \begin{figure}
      \includegraphics[width=6cm, angle=270]
        {Theoretical_h-index_Pareto.pdf}

      \caption{Przykladowy wykres wygenerowany
        w srodowisku \texttt{R}}
    \end{figure}
  \end{center}
\end{frame}
```

Rysunki

Przykład: generowanie wykresu w R

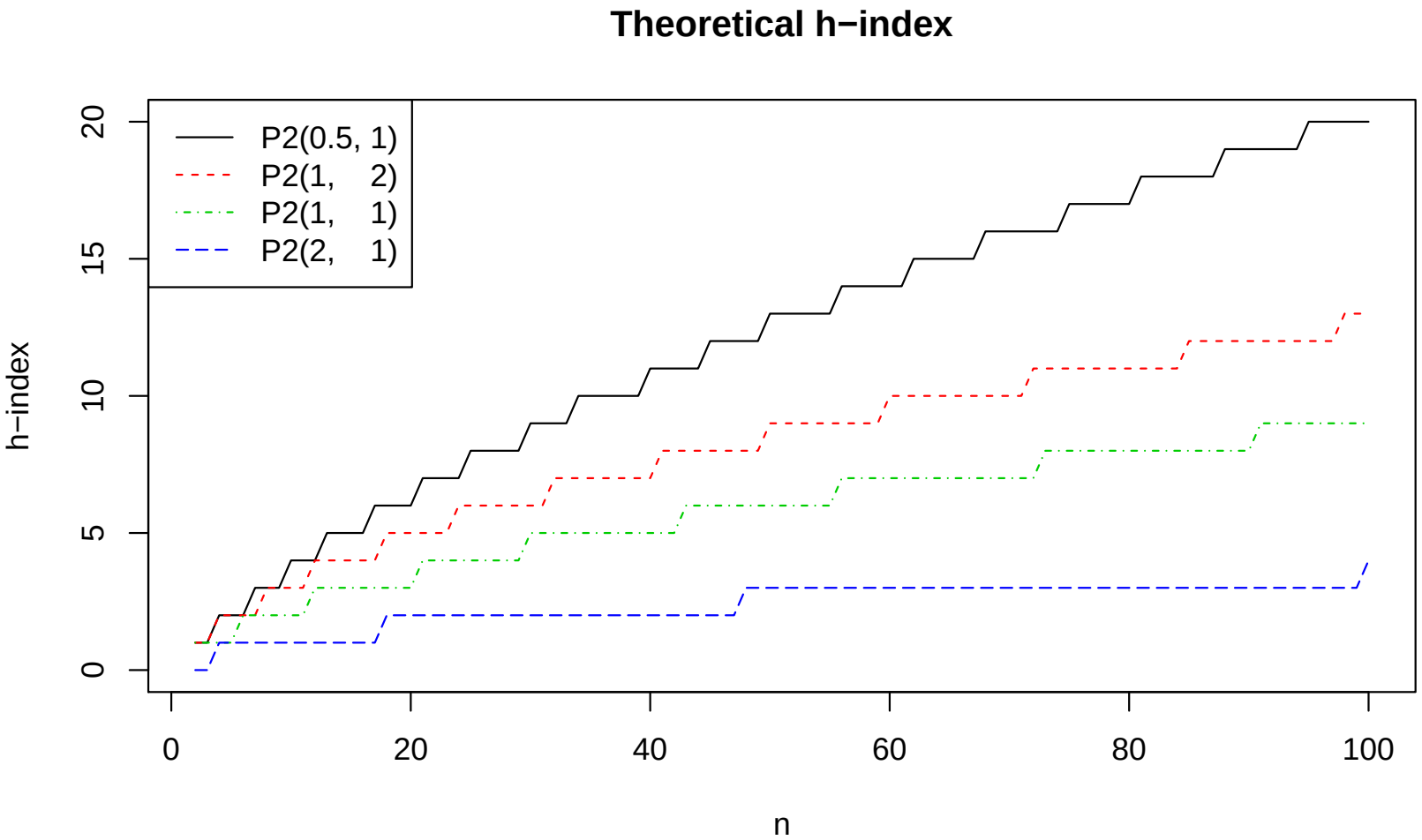
Fragment kodu w R:

```
pdf("nazwapliku.pdf", height=8, width=5)
plot(...)
dev.off()
```

Dodawanie ramki z rysunkiem:

```
\begin{frame}
  \begin{center}
    \begin{figure}
      \includegraphics[width=6cm, angle=270]
        {Theoretical_h-index_Pareto.pdf}

      \caption{Przykladowy wykres wygenerowany
        w srodowisku \texttt{R}}
    \end{figure}
  \end{center}
\end{frame}
```



Rysunek: Przykładowy wykres wygenerowany w środowisku R

Kody źródłowe

Pakiet `listings`

Jeśli zachodzi potrzeba dodania do prezentacji kodów źródłowych, możemy skorzystać z pakietu `listings` (zob. np. plik .tex dla niniejszej prezentacji).

`listings` obsługuje kolorowanie składni dla różnych języków programowania (w tym `R`, `C++` i LaTeX).

Środowisko `verbatim`

W ostateczności zawsze można skorzystać np. ze środowiska `verbatim` z pakietu *verbatim*.

Kody źródłowe

Pakiet `listings`

Jeśli zachodzi potrzeba dodania do prezentacji kodów źródłowych, możemy skorzystać z pakietu `listings` (zob. np. plik .tex dla niniejszej prezentacji).

`listings` obsługuje kolorowanie składni dla różnych języków programowania (w tym `R`, `C++` i LaTeX).

Środowisko `verbatim`

W ostateczności zawsze można skorzystać np. ze środowiska `verbatim` z pakietu *verbatim*.

Podsumowanie

Co się stało?

Omówiliśmy podstawowe zagadnienia związane z tworzeniem prezentacji w LaTeX-u z użyciem pakietu Beamer.

Co nas czeka?

Jak widać, nie ominie nas konieczność uzyskania pewnej biegłości w posługiwaniu się LaTeX-em.

Rada

Pamiętajmy na koniec, że prezentacja to tylko narzędzie wspomagające dobrą komunikację.

Podsumowanie

Co się stało?

Omówiliśmy podstawowe zagadnienia związane z tworzeniem prezentacji w LaTeX-u z użyciem pakietu Beamer.

Co nas czeka?

Jak widać, nie ominie nas konieczność uzyskania pewnej biegłości w posługiwaniu się LaTeX-em.

Rada

Pamiętajmy na koniec, że prezentacja to tylko narzędzie wspomagające dobrą komunikację.

Podsumowanie

Co się stało?

Omówiliśmy podstawowe zagadnienia związane z tworzeniem prezentacji w LaTeX-u z użyciem pakietu Beamer.

Co nas czeka?

Jak widać, nie ominie nas konieczność uzyskania pewnej biegłości w posługiwaniu się LaTeX-em.

Rada

Pamiętajmy na koniec, że prezentacja to tylko narzędzie wspomagające dobrą komunikację.

Bibliografia

Zobacz: `www.gagolewski.com/teaching/tutorials/beamer/`

Powodzenia!