

Лабораторная работа №4

Including Graphics (LaTeX)

Тема: подключение изображений, floats, ссылки `\label/\ref`,
сравнение `\textwidth` и `\ linewidth`

Цель работы

- Научиться подключать внешние изображения в LaTeX через `graphicx`
- Освоить параметры изменения вида рисунков: размер, поворот, обрезка
- Понять, как работают *floats* и спецификаторы размещения
- Разобраться с перекрёстными ссылками и количеством прогонов компиляции
- Сравнить `\textwidth` и `\ linewidth` (в т.ч. в режиме `twocolumn`)

Задание

1. Вставить собственное изображение из подпапки `figs/`
2. Исследовать `height`, `width`, `scale`, `angle`, `trim`, `clip`
3. Добавить «длинный» текст через `lipsum`, протестировать `float'ы`
`[h, t, b, p]`
4. Сравнить `\textwidth` и `\ linewidth` (обычный режим и `twocolumn`)
5. Проверить работу `\label/\ref`, включая намеренно неправильные случаи

Структура проекта

- `main4_ru.tex` / `main4.tex` --- исходники (RU/EN)
- `figs/image.png` --- своё изображение
- `main4_ru.pdf` / `main4.pdf` --- результат компиляции
- `report.md` → `report.pdf`, `report.docx`
- `presentation.md` → `presentation.pdf`

Подключение графики

Для внешних изображений нужен пакет `graphicx`:

```
\usepackage{graphicx}
```

Удобно хранить картинки в подпапке и добавить путь:

```
\graphicspath{{figs/}}
```

Вставка изображения

Минимальный пример:

```
\begin{center}  
    \includegraphics[height=2cm]{image.png}  
\end{center}
```



LaTeX сам подбирает расширение, но на практике лучше указывать явно:

{image.png} или {image.pdf}

Управление размером

Часто задают размер относительно текста страницы:

```
\includegraphics[height=0.35\textheight]{image.png}  
\includegraphics[width=0.55\textwidth]{image.png}
```

- **width** --- ширина
- **height** --- высота Пропорции сохраняются автоматически

Масштаб и поворот

```
\includegraphics[scale=0.6]{image.png}  
\includegraphics[angle=15, scale=0.6]{image.png}
```

- **scale** --- общий масштаб
- **angle** --- поворот (в градусах)

Обрезка: trim + clip

trim задаётся как **left bottom right top**:

```
\includegraphics[  
    clip,  
    trim=20 10 80 40,  
    width=0.6\textwidth  
]{image.png}
```

- **clip** обязателен, иначе обрезка не применится
- удобно «подрезать» поля/лишнее

Зачем нужен `lipsum`

Чтобы LaTeX мог «двигать» float'ы, нужен объём текста:

```
\usepackage{lipsum}  
...  
\lipsum[1-4]
```

Иначе не видно, как размещение реально работает

Floats: рисунки как плавающие объекты

Пример float'a:

```
\begin{figure}[ht]
\centering
\includegraphics[width=0.6\textwidth]{image.png}
\caption{Пример float с [ht].}
\label{fig:ht}
\end{figure}
```

Внутри float лучше использовать `\centering`, а не `center`

Спецификаторы размещения

- [h] --- here (если получится)
- [t] --- top (вверх страницы)
- [b] --- bottom (вниз страницы)
- [p] --- отдельная страница для float'ов

LaTeX выбирает оптимально, если места «здесь» нет

Принудительно «строго здесь»: [H]

Пакет `float` добавляет `[H]`:

```
\usepackage{float}
...
\begin{figure}[H]
  \centering
  \includegraphics[width=0.55\textwidth]{image.png}
  \caption{Строгое размещение [H].}
  \label{fig:H}
\end{figure}
```

⚠️ Может создавать большие пустые области --- использовать
осторожно

Перекрёстные ссылки

Ставим метку:

```
\label{fig:ht}
```

Ссылаемся:

```
см. рисунок~\ref{fig:ht}
```

Тильда \sim не даёт переносить строку между словом и номером

Почему нужны 2 прогона компиляции

`\label / \ref` используют `.aux` файл:

1. 1-й прогон: LaTeX «собирает» номера и записывает их в `.aux`
2. 2-й прогон: подставляет номера в текст

Если видишь `??` --- просто собери ещё раз
Обычно нужно
минимум (2)
прогона

Где правильно ставить `\label` {=tex} в `figure`

✓ Правильно: после (или внутри) `\caption`

```
\caption{...}  
\label{fig:ok}
```

✗ Неправильно: до `\caption`

```
\label{fig:wrong}  
\caption{...}
```

Потому что `\label` ссылается на последний пронумерованный

`\label`

`{=tex}` в `equation`: правильно

и неправильно



Правильно (внутри окружения):

```
\begin{equation}
e^{i\pi}+1=0
\label{eq:inside}
\end{equation}
```



Неправильно (после окружения):

```
\begin{equation}
a^2+b^2=c^2
\end{equation}
```

`\textwidth`

`{=tex}` vs

`\ linewidth`

`{=tex}`

- `\textwidth` --- ширина текстового блока страницы
- `\ linewidth` --- текущая ширина строки (может отличаться локально)

В `twocolumn` разница заметна: `\ linewidth` \approx ширина колонки

Эксперимент с `twocolumn`

Компилируем два варианта:

```
\documentclass[a4paper,12pt]{article}  
% и затем  
\documentclass[a4paper,12pt,twocolumn]{article}
```

И сравниваем:

```
\includegraphics[width=0.8\textwidth]{image.png}  
\includegraphics[width=0.8\linewidth]{image.png}
```

Компиляция

Обычно достаточно `pdflatex` (и два прогона):

```
pdflatex main4_ru.tex  
pdflatex main4_ru.tex
```

Для EN-версии аналогично:

```
pdflatex main4.tex  
pdflatex main4.tex
```

Итоги

Выполнено:

- Подключена собственная графика из `figs/`
- Проверены `height/width/scale/angle/trim/clip`
- Исследованы `float`'ы и спецификаторы `[h,t,b,p]` + `[H]`
- Изучены перекрёстные ссылки и необходимость (2) прогонов
- Показана разница `\textwidth` и `\ linewidth` (в т.ч. `twocolumn`)
- Подготовлены отчёт и презентация, материалы опубликованы

