



Beamer in $\text{\LaTeX}$

Prof. Pranay Kumar Saha

April 2, 2025





What is Beamer?

Presentations in $\text{\LaTeX}$

- Beamer is a $\text{\LaTeX}$ document class (`\documentclass{beamer}`) specifically designed for creating presentation slides.
- It leverages $\text{\LaTeX}$'s strengths:
 - Excellent typesetting quality, especially for mathematics.
 - Structured source code using familiar $\text{\LaTeX}$ commands.
 - Consistent look and feel through themes.
 - Powerful features like overlays (step-by-step reveals).
 - Generates PDF output directly.
 - Integrates seamlessly with bibliography (BibLaTeX), figures, tables, etc.
- The core content unit in Beamer is the `frame` environment, which corresponds to a single slide (potentially with multiple overlay steps).



Minimal Beamer Document

The Core Structure

Here's the absolute minimum needed for a one-slide presentation:

```
\documentclass{beamer}
\begin{document}
\begin{frame}
  \frametitle{My First Slide} % Optional title for the frame
  Hello, Beamer World!
\end{frame} %%%
\end{document}
```



Minimal Beamer Document

The Core Structure

Key Points

- `\documentclass{beamer}`: Specifies the Beamer class.
- `\begin{document}... \end{document}`: Standard $\text{\LaTeX}$ document structure.
- `\begin{frame}... \end{frame}`: Defines a single slide.
- `\frametitle{...}`: Sets the title displayed on the slide (often in the header). Optional: `\framesubtitle{...}`.

Compilation

Compile using `pdflatex` (or your preferred $\text{\LaTeX}$ engine) just like any other $\text{\LaTeX}$ document. Usually needs only one run unless you use features like TOC, labels/refs, or bibliography.



Structuring: The Title Page

Define title information in the preamble:

```
% In Preamble:  
\title{My Awesome Presentation}  
\subtitle{An Introduction to Beamer}  
\author{A. N. Author}  
\institute{University of \LaTeX}  
\date{\today} % Or a specific date
```

Then, use the `\titlepage` command within a frame:

```
% In Document Body:  
\begin{frame}  
  \titlepage  
\end{frame} %%%
```

The appearance is controlled by the chosen theme.



Structuring: Sections & Subsections

Use standard $\text{\LaTeX}$ sectioning commands to structure your presentation:

```
\section{Introduction}
```

```
% Frames for intro...
```

```
\section{Methods}
```

```
\subsection{Data Collection}
```

```
% Frames for data collection...
```

```
\subsection{Analysis}
```

```
% Frames for analysis...
```

```
\section{Conclusion}
```

```
% Frames for conclusion...
```



Structuring: Sections & Subsections

Purpose

- Provides logical structure to your source code.
- Used to generate the Table of Contents.
- Often displayed automatically by themes in navigation elements (header, sidebar, etc.) to orient the audience.



Structuring: Table of Contents

Generate an outline slide using `\tableofcontents`. Place it usually near the beginning.

```
\begin{frame}{Outline} % Set frame title to "Outline"  
  \tableofcontents  
\end{frame} %%
```



Structuring: Table of Contents

Options

Control highlighting or section visibility:

- `\tableofcontents[currentsection]`: Highlights the current section.
- `\tableofcontents[currentsubsection]`: Highlights the current subsection.
- Combine options:

`\tableofcontents[currentsection, subsectionstyle=show/shaded/hide]`

Requires at least two compilation runs to stabilize.



Themes: Overall Look & Feel

Themes control the visual design (layout, fonts, colors). Load them in the preamble.

```
\usetheme{ThemeName}
```

Sets the main presentation theme (layout, structure elements).

- Common examples: Madrid, Warsaw, Singapore, Boadilla, AnnArbor, Berlin, Darmstadt, default.

```
\usetheme{Madrid} % Use the Madrid theme
```



Themes: Overall Look & Feel

Finding Themes

Explore options online! Search for "Beamer Theme Matrix".

The figure illustrates the impact of Beamer themes on the presentation's appearance. It compares a default theme with the 'Madrid theme + beaver' theme.

Default Theme (Left):

- Title: About the Beamer class in presentation making
- Subtitle: A short story
- Authors: A. B. Arthur¹ J. Doe²
- Footnote: ¹Faculty of Physics Very Famous University
- Footnote: ²Faculty of Chemistry Very Famous University
- Date: Very Large Conference, April 2021
- Footer: Arthur, Doe (VFU) About Beamer VLC 2021 1 / 8

Madrid theme + beaver (Right):

- Title: Demonstrating large fonts
- Subtitle: Demonstrating larger fonts
- Authors: A. B. Arthur¹ J. Doe²
- Footnote: ¹Faculty of Physics Very Famous University
- Footnote: ²Faculty of Chemistry Very Famous University
- Date: Very Large Conference, April 2021
- Footer: Arthur, Doe (VFU) Madrid Beamer VLC 2021 1 / 8

The Madrid theme + beaver theme features a more modern, clean design with a light gray background, larger fonts, and a prominent blue header bar. It also includes a sidebar with a 'Remark' section.

Figure: Themes drastically change appearance (Illustrative).



Color and Font Themes

Fine-tune appearance within a presentation theme.

`\usecolortheme{ThemeName}`

Modifies the color palette used for elements.

- Examples: default, whale, orchid, beaver, seahorse.

```
\usecolortheme{whale} % Use the 'whale' color scheme
```

`\usefonttheme{ThemeName}`

Changes font styles (e.g., serif vs sans-serif, bold structural elements).

- Examples: default, serif, structurebold, structuresmallcapsserif.

```
\usefonttheme{serif} % Use serif fonts where possible
```



Color and Font Themes

Inner/Outer Themes

For advanced control, you can set themes for inner elements (like itemize bullets, blocks) using `\useinnertheme` and outer elements (header, footer, sidebar) using `\useoutertheme`. Often unnecessary for beginners.



Basic Content Elements

Most standard $\text{\LaTeX}$ content works as expected:

Text Formatting:

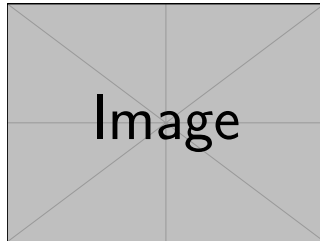
- Paragraphs (separated by blank lines).
- `\emph{emphasis}` (*emphasis*)
- `\textbf{bold}` (**bold**)
- `\textit{italics}` (*italics*)

Lists:

- Item 1
 - Item 2
1. Number 1
 2. Number 2

Images: (graphicx needed)

`\includegraphics[width=...]{file}`



Size relative to `\textwidth` or `\paperwidth`!

Tables & Math: Standard `tabular`, `booktabs`,



Displaying Code

Remember the `[fragile]` Option!

When a frame contains verbatim-like content, you **MUST** use the `[fragile]` option for the frame environment.

Options for Code

1. **verbatim environment:** Simple, no syntax highlighting. Displays text exactly as typed, including spacing. Requires `[fragile]`.
2. **listings package:** (Not used here) Good syntax highlighting, configurable.
3. **minted package:** (Not used here) Excellent highlighting via Pygments (Python).



Displaying Code

Remember the `[fragile]` Option!

`verbatim` Example

```
% Python code example
def greet(name):
    """Greets the person."""
    print(f"Hello, {name}!")

greet("Beamer User")
```

`\begin{frame}[fragile]`

Crucial for frames with `verbatim` or sometimes complex `\verb`. Beamer writes frame content to temporary files, and fragile content needs special handling.



Beamer Blocks

Structuring Content Within a Frame

Blocks visually group related content. Their appearance is heavily theme-dependent.

Standard Block Title

This is a standard block environment. Good for general grouping. Use `\begin{block}{Title}...\end{block}`.

Alert Block Title

This is an `alertblock`. Often used for warnings, important points. Typically uses distinct colors (e.g., red).

Example Block Title

This is an `exampleblock`. Often used for code examples, illustrations. Typically uses different colors (e.g., green).



Overlays: Step-by-Step Reveals

The Core Dynamic Feature

Overlays allow you to reveal parts of a single slide incrementally without duplicating the entire frame.

Basic Syntax: `<overlay specification>`

Many commands (`\item`, `\textbf`, `\includegraphics`, etc.) accept an overlay specification in angle brackets `< >` immediately following the command.

- `<+>`: Incremental reveal (appears on next step). Often used with `\item<+>`.
- `<N>`: Appears only on overlay step N.
- `<N->`: Appears on step N and all subsequent steps.
- `<-N>`: Appears on steps 1 through N.
- `<N-M>`: Appears on steps N through M.
- `<N,M,P>`: Appears on steps N, M, and P.



Overlays: Step-by-Step Reveals

The Core Dynamic Feature

Example

```
\item<1-> First point.  
\item<2-> Second point.  
\item<3-> Third point. \textbf<4->{This part} gets bold later.
```

- First point appears.



Overlays: Step-by-Step Reveals

The Core Dynamic Feature

Example

```
\item<1-> First point.  
\item<2-> Second point.  
\item<3-> Third point. \textbf<4->{This part} gets bold later.
```

- First point appears.
- Second point appears now.



Overlays: Step-by-Step Reveals

The Core Dynamic Feature

Example

```
\item<1-> First point.  
\item<2-> Second point.  
\item<3-> Third point. \textbf<4->{This part} gets bold later.
```

- First point appears.
- Second point appears now.
- Third point. This part gets bold later.



Overlays: Step-by-Step Reveals

The Core Dynamic Feature

Example

```
\item<1-> First point.  
\item<2-> Second point.  
\item<3-> Third point. \textbf<4->{This part} gets bold later.
```

- First point appears.
- Second point appears now.
- Third point. **This part** gets bold later.



Overlays: The `\pause` Command

A Simpler Way to Reveal Sequentially

The `\pause` command is a simple way to create overlay steps between paragraphs or list items. Each `\pause` increments the overlay counter.

Code

Here is the first piece of text.

```
\pause % End of step 1
```

Now the second piece appears.

```
\pause % End of step 2
```

```
\begin{itemize}
```

```
  \item First item. \pause % End of step 3
```

```
  \item Second item.
```

```
\end{itemize}
```

```
% End of step 4
```



Overlays: The `\pause` Command

A Simpler Way to Reveal Sequentially

Live Demo

Here is the first piece of text.



Overlays: The `\pause` Command

A Simpler Way to Reveal Sequentially

Live Demo

Here is the first piece of text.
Now the second piece appears.



Overlays: The `\pause` Command

A Simpler Way to Reveal Sequentially

Live Demo

Here is the first piece of text.

Now the second piece appears.

- First item.



Overlays: The `\pause` Command

A Simpler Way to Reveal Sequentially

Live Demo

Here is the first piece of text.

Now the second piece appears.

- First item.
- Second item.

Less flexible than `< >` specifications, but often sufficient.



Overlays: More Commands (Briefly)

Finer Control Over Visibility

Besides the basic `< >` specification, there are commands for more control:

- `\only<spec>{content}`: Includes content *only* on specified steps. **Does not reserve space** on other steps. Can cause layout jumps.
- `\uncover<spec>{content}`: Makes content visible only on specified steps. **Reserves space** (content is transparent/invisible) on other steps. Preferred for stable layout.
- `\visible<spec>{content}`: Like `\uncover`, reserves space.
- `\invisible<spec>{content}`: Hides content on specified steps, reserves space.
- `\alt<spec>{content shown}{content hidden}`: Shows different content depending on whether the step matches the specification. Reserves space for the larger of the two contents.



Overlays: More Commands (Briefly)

Finer Control Over Visibility

Example using `uncover`

This text is always here.

Often, simple `< >` on items or `\pause` is enough for common use cases.



Overlays: More Commands (Briefly)

Finer Control Over Visibility

Example using `uncover`

This text is always here. This text appears on step 2+ but space was reserved.

Often, simple `< >` on items or `\pause` is enough for common use cases.



Overlays: More Commands (Briefly)

Finer Control Over Visibility

Example using `uncover`

This text is always here. This text appears on step 2+ but space was reserved. And this on step 3+.

Often, simple `< >` on items or `\pause` is enough for common use cases.



Columns: Side-by-Side Content

Use the `columns` and `column` environments to arrange content horizontally.

Syntax

```
\begin{columns}[options] % e.g., [T] align tops, [c] center vertically
```

```
\begin{column}{<width>} % e.g., 0.5\textwidth
```

```
% Content for first column...
```

```
\end{column}
```

```
\begin{column}{<width>} % e.g., 0.45\textwidth
```

```
% Content for second column...
```

```
\end{column}
```

```
% Add more columns if needed...
```

```
\end{columns}
```



Columns: Side-by-Side Content

Column 1

- Point A
- Point B

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.

Column 2

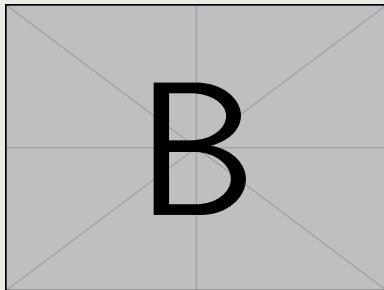


Figure: Image in column 2



Speaker Notes

Reminders for the Presenter

Add notes visible only to the speaker using the `\note{}` command.

Code

```
\begin{frame}{Important Concept}
  This slide explains the Core Idea.
  \note{ % Start speaker note
    Remember to emphasize point X.
    Maybe mention the historical context briefly.
    Don't spend more than 2 minutes here.
  } % End speaker note
  Here is the main content...
\end{frame} %%
```



Speaker Notes

Reminders for the Presenter

Viewing Notes

Notes are not shown in the standard PDF output. To see them:

- Add `\setbeameroption{show notes}` to your preamble (creates pages with notes).
- Use presentation viewers that support Beamer notes (like `pdfpc`, some dual-screen modes in PDF viewers).



Handouts

Creating Printable Versions

To create a version optimized for printing (usually showing only the final state of each overlay slide):

Use a Class Option

Change the document class line in your preamble:

```
% Original:
```

```
% \documentclass{beamer}
```

```
% For Handouts:
```

```
\documentclass[handout]{beamer}
```



Customization (Overview)

Going Beyond Themes (Advanced)

Beamer offers deep customization, but it can be complex:

`\setbeamercolor`

Change the foreground (`fg`) or background (`bg`) color of specific Beamer elements (like block titles, itemize items, frametitle).

% Example: Make alert block titles blue

```
\setbeamercolor{alerted text}{fg=blue}
```

```
\setbeamercolor{block title alerted}{bg=blue!20!white, fg=black}
```

`\setbeamertemplate`

Redefine the template used to draw elements (like navigation symbols, footnotes, block appearance). This is very powerful but requires understanding Beamer's template system.



Customization (Overview)

Going Beyond Themes (Advanced)

```
% Example: Remove navigation symbols  
\setbeamertemplate{navigation symbols}{{}}
```

Recommendation

Stick with existing themes initially. Dive into deep customization only when necessary, and consult the Beamer user guide (`texdoc beamer`).



Workflow & Best Practices

- **Start Simple:** Begin with basic frames and structure, add complexity (themes, overlays) gradually.
- **Structure First:** Use `\section` and `\subsection` early to organize your thoughts and code.
- **Choose Theme:** Select a theme that broadly fits your needs early on. Avoid excessive tweaking initially.
- **Purposeful Overlays:** Use overlays (`\pause`, `<` `>`) to guide the audience, reveal information logically, or build complex diagrams. Don't overuse them just for visual effects.
- **Keep Slides Clean:** Avoid cluttering slides with too much text or too many elements. Use speaker notes (`\note`) for details.
- **Remember `[fragile]`:** Essential for frames with code listings or verbatim text. Add it immediately when needed.
- **Compile Often:** Check your output frequently to catch errors early. Remember TOC/refs need multiple runs.
- **Read the Docs:** The Beamer User Guide (`texdoc beamer`) is comprehensive and the ultimate reference.



Summary

You've learned the basics of Beamer:

- Core structure: `\documentclass{beamer}`, `frame`, `frametitle`.
- Presentation structure: `\titlepage`, `\section`, `\tableofcontents`.
- Appearance: Themes (`\usetheme`, etc.).
- Content: Standard $\text{\LaTeX}$, Blocks, Code (`[fragile]`!).
- Dynamics: Overlays (`<` `>`, `\pause`), Columns.
- Interaction: Hyperlinks, Buttons.
- Output: Notes (`\note`), Handouts (`[handout]`).

Happy Beamer Presenting!



Beamer in $\text{\LaTeX}$

Thank You for Listening!