

Beamer Feature Showcase

Comprehensive Reference for Automated slides

System Reference

January 11, 2026

This reference covers structure, content, visuals, and animation.

Table of Contents

- 1 Structure & Layout
 - 2 Rich Content
 - 3 Visuals & Navigation
 - 4 Animation

Structure & Layout

Incremental Reveals

To keep the video engaging, reveal content step-by-step using <+->.

- **Step 1:** Introduce the concept.

Incremental Reveals

To keep the video engaging, reveal content step-by-step using <+->.

- **Step 1:** Introduce the concept.
- **Step 2:** Expand on details.

Incremental Reveals

To keep the video engaging, reveal content step-by-step using <+->.

- **Step 1:** Introduce the concept.
 - **Step 2:** Expand on details.
 - **Step 3:** Conclude the point.

Comparing Approaches (Columns)

Use columns to compare side-by-side.

Approach A

```
1 def fib(n):
2     if n <= 1: return n
3     return fib(n-1) + fib(n-2)
4
```

Approach B

```
1 def fib(n):
2     a, b = 0, 1
3     for _ in range(n):
4         a, b = b, a + b
5     return a
6
```

Beamer Blocks

Organize information using standard Beamer environment blocks.

Standard Block

This is a normal block for general information.

Alert Block

Use this for warnings or critical points.

Examples

This is an example block, useful for case studies.

Rich Content

Math & Equations

Beamer handles complex math gracefully.

- Inline Math: $e^{i\pi} + 1 = 0$
- Block Equations with overlays:

Math & Equations

Beamer handles complex math gracefully.

- Inline Math: $e^{i\pi} + 1 = 0$
- Block Equations with overlays:

$$f(x) = x^2 + 2x + 1$$

Math & Equations

Beamer handles complex math gracefully.

- Inline Math: $e^{i\pi} + 1 = 0$
- Block Equations with overlays:

$$\begin{aligned}f(x) &= x^2 + 2x + 1 \\&= (x + 1)^2\end{aligned}$$

Code Evolution

We can simulate "Magic Move" by showing code changes across overlays.

```
1 def calculate_area(radius):  
2     pi = 3.14  
3     return pi * radius * radius  
4
```

Initial State

Code Evolution

We can simulate "Magic Move" by showing code changes across overlays.

```
1 import math  
2  
3 def calculate_area(radius):  
4     return math.pi * radius * radius  
5
```

Refactored (Import Math)

Styled Components

Use tcolorbox to simulate modern UI components.

Info Card

This box mimics a standardized component style.

Alert Component

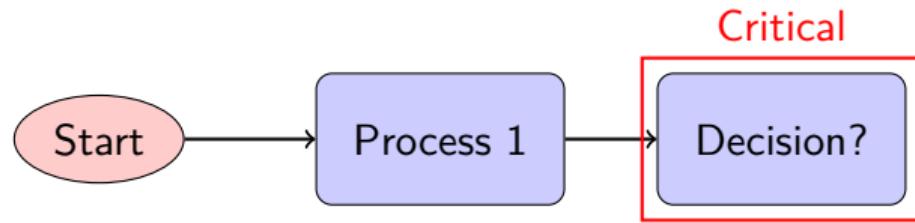
Warning: This is an important alert.

Visuals & Navigation

Digrams (TikZ)



Digrams (TikZ)



Navigation Buttons

Beamer supports interactive buttons.

[Jump to Transitions](#)

[▶ Go to Transitions](#)

Animation

Transitions

This slide used a dissolve transition to appear.

Other standard transitions:

- `transblindshorizontal`
- `transboxin`
- `transglitter`