Beamer Overlay Techniques Demo

Interactive Examples

LaTeX Overlay Guide

July 9, 2025

Table of Contents

Basic Pause Command

- First point

Basic Pause Command

- First point
- Second point appears after pause

Basic Pause Command

- First point
- Second point appears after pause
- Third point appears last

Using \only

This text appears only on slide 1

Using \only

This text replaces it on slide 2

Using \only

And this appears on slide 3

Using \only

This stays from slide 4 onwards

Using \uncover

Space is Reserved

Always visible

Using \uncover

Space is Reserved

Always visible

Appears on slide 2 (space was reserved)

Using \uncover

Space is Reserved

Always visible

Appears on slide 2 (space was reserved)

Appears on slide 3

Using \uncover

Space is Reserved

Always visible

Appears on slide 2 (space was reserved)

Appears on slide 3

Only on slide 4

Using \visible and \invisible

Visible Demo:

- Always visible

Invisible Demo:

- Hidden on slide 2
- Hidden on slide 3

Using \visible and \invisible

Visible Demo:

- Always visible
- From slide 2

Invisible Demo:

- Hidden on slide 3
- Hidden on slide 1

Using \visible and \invisible

Visible Demo:

- Always visible
- From slide 2
- Only on slide 3

Invisible Demo:

- Hidden on slide 2
- Hidden on slide 1

Using \alt

Welcome to Slide 1

Blue on other slides

Using \alt

Now on Slide 2

Red on slide 2

Dynamic Highlighting with \alert

Theorem

The sum $a + b$ equals c when $a = c - b$.

Dynamic Highlighting with \alert

Theorem

The sum $a + b$ equals c when $a = c - b$.

- We start with the sum

Dynamic Highlighting with \alert

Theorem

The sum $a + b$ equals c when $a = c - b$.

- We start with the sum
- We know the result

Dynamic Highlighting with \alert

Theorem

The sum $a + b$ equals c when $a = c - b$.

- We start with the sum
- We know the result
- We can derive the relationship

Automatic Incremental Lists

Manual

- First

Automatic

- First

Automatic Incremental Lists

Manual

- First
- Second

Automatic

- First
- Second

Automatic Incremental Lists

Manual

- First
- Second
- Third

Automatic

- First
- Second
- Third

Alert on Appearance

1. First item (highlighted when appearing)

Alert on Appearance

1. First item (highlighted when appearing)
2. Second item (highlighted when appearing)

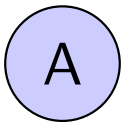
Alert on Appearance

1. First item (highlighted when appearing)
2. Second item (highlighted when appearing)
3. Third item (highlighted when appearing)

Alert on Appearance

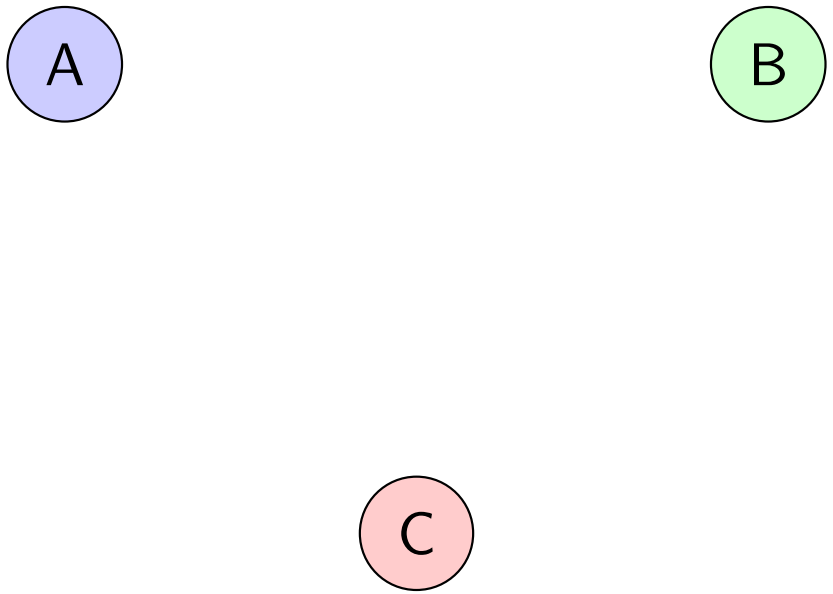
1. First item (highlighted when appearing)
2. Second item (highlighted when appearing)
3. Third item (highlighted when appearing)
4. Fourth item (highlighted when appearing)

TikZ with Overlays

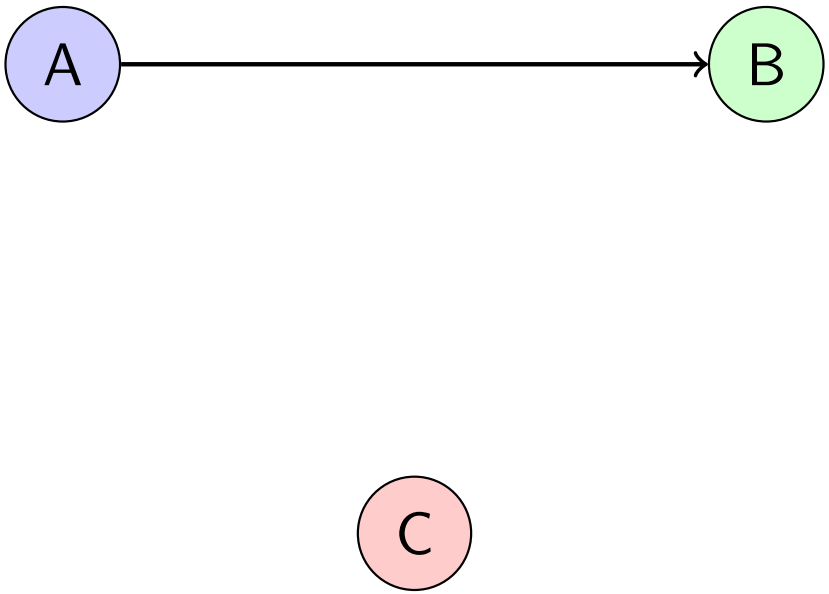


TikZ with Overlays

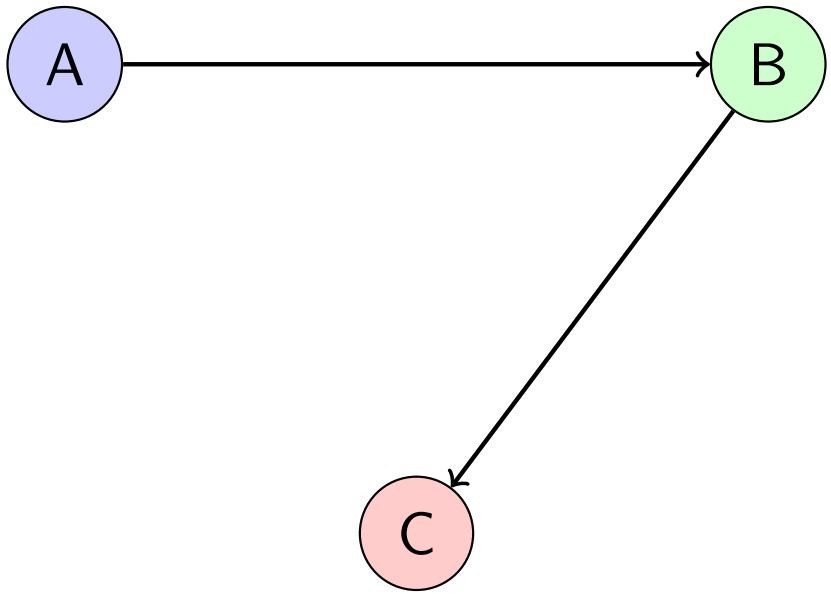
TikZ with Overlays



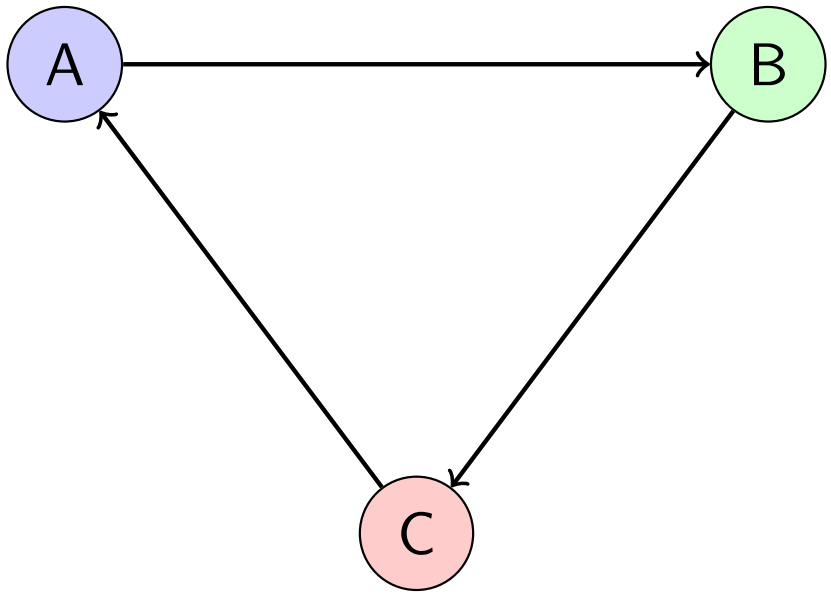
TikZ with Overlays



TikZ with Overlays

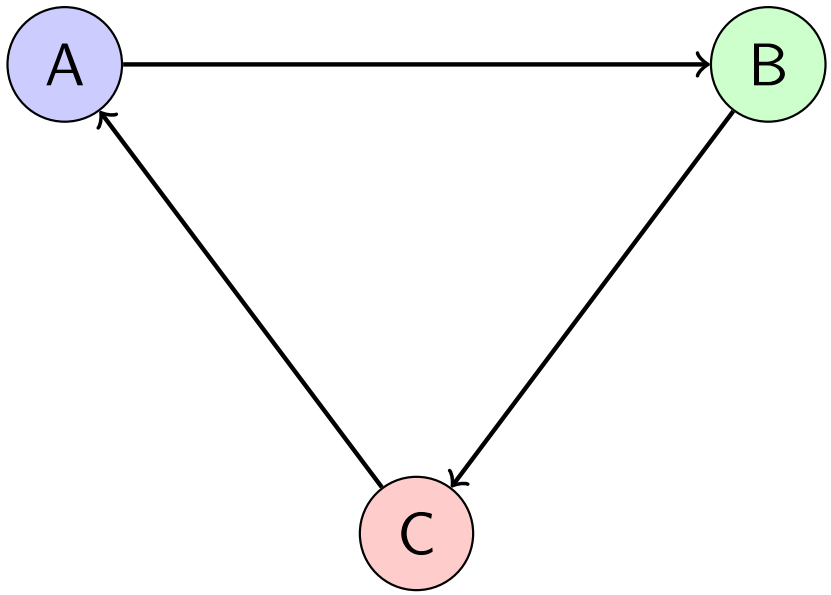


TikZ with Overlays



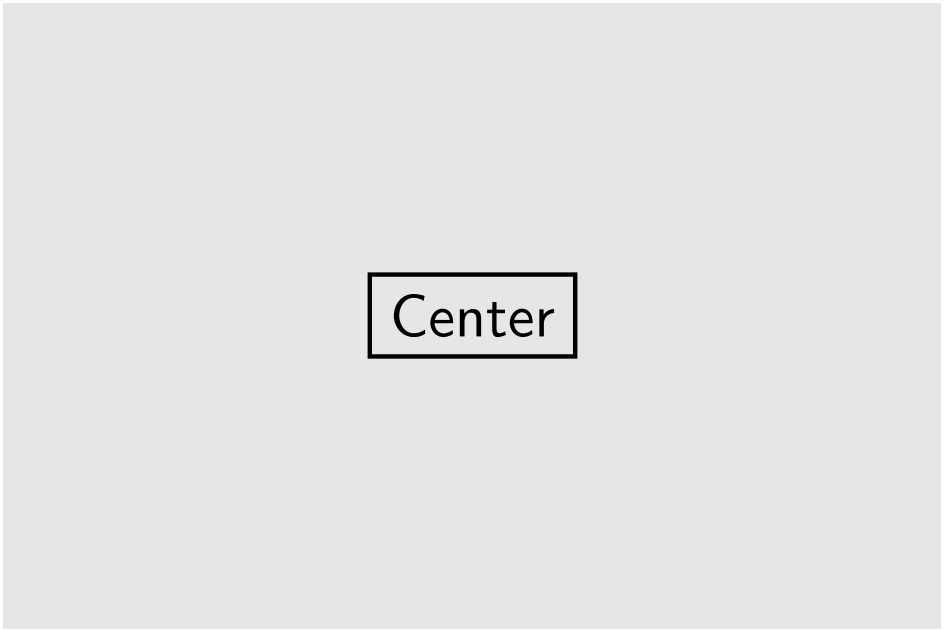
TikZ with Overlays

Triangle Graph

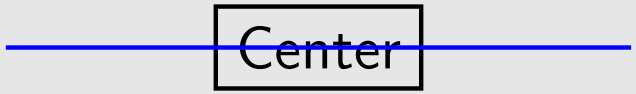


TikZ Visible On Style

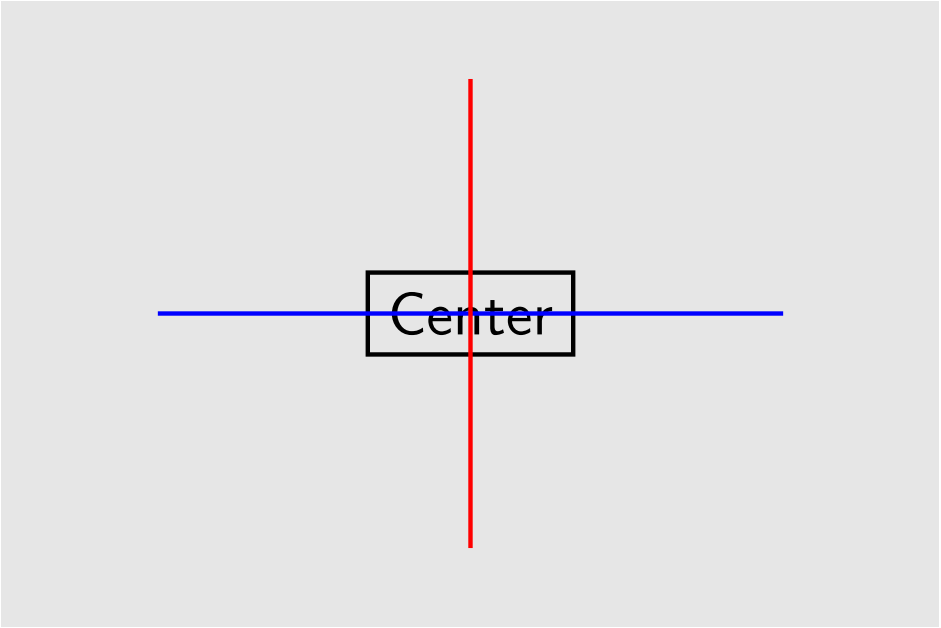
TikZ Visible On Style



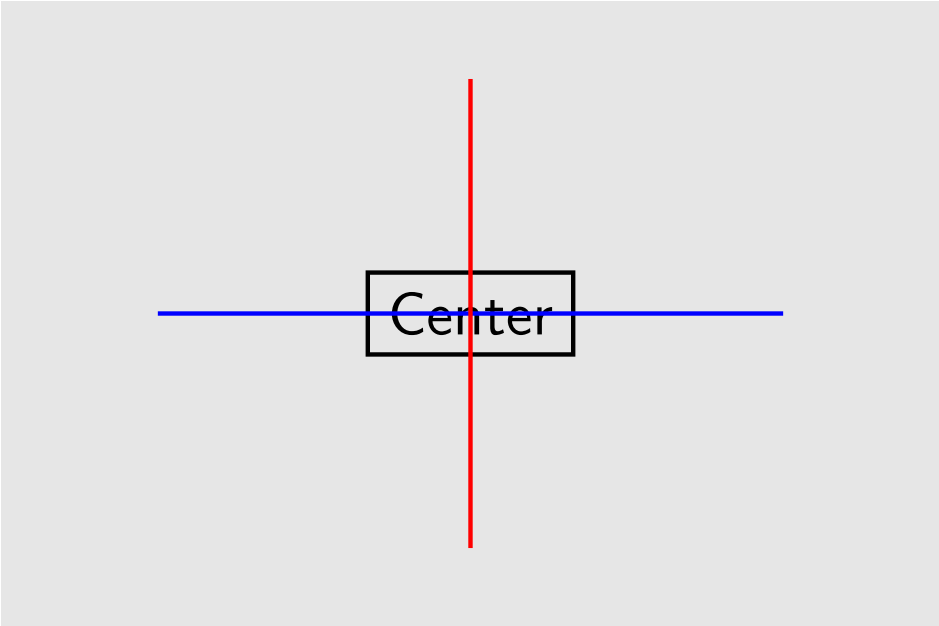
TikZ Visible On Style



TikZ Visible On Style



TikZ Visible On Style



Coordinate System!

Using \temporal

Before activation

- Watch the text above

Using \temporal

ACTIVE!

- Watch the text above
- It's now active

Using \temporal

ACTIVE!

- Watch the text above
- It's now active

Using \temporal

ACTIVE!

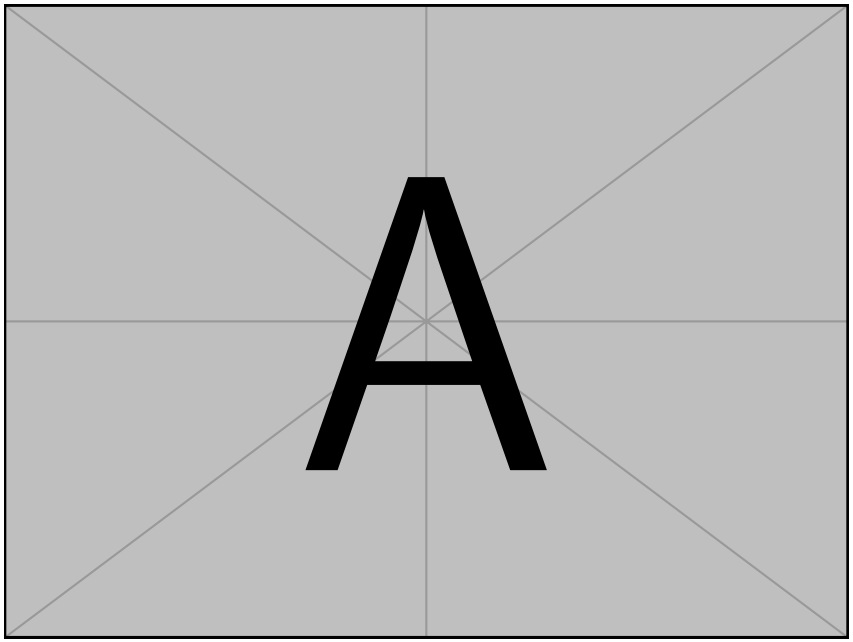
- Watch the text above
- It's now active
- Active phase ending...

Using \temporal

After activation

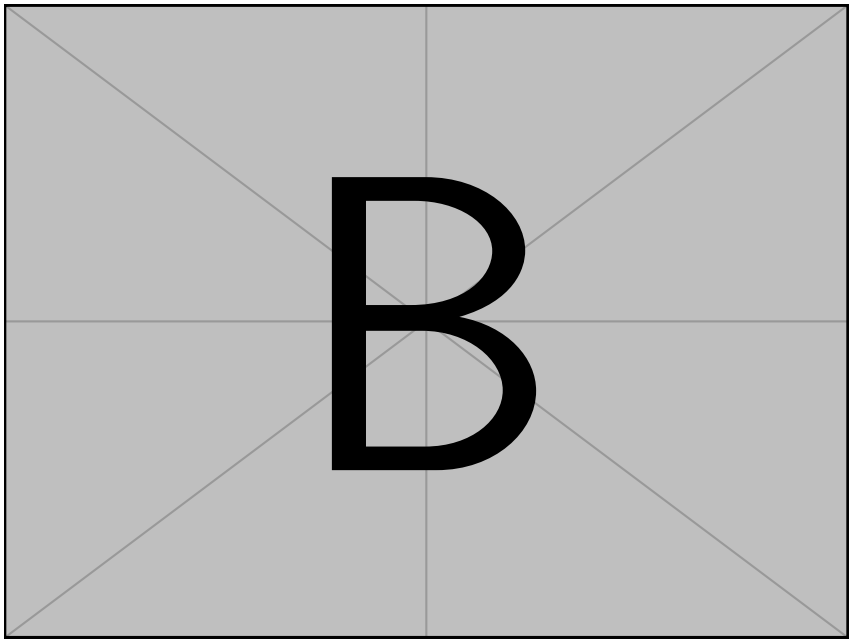
- Watch the text above
- It's now active
- Active phase ending...
- Back to inactive state

Complex Temporal Example



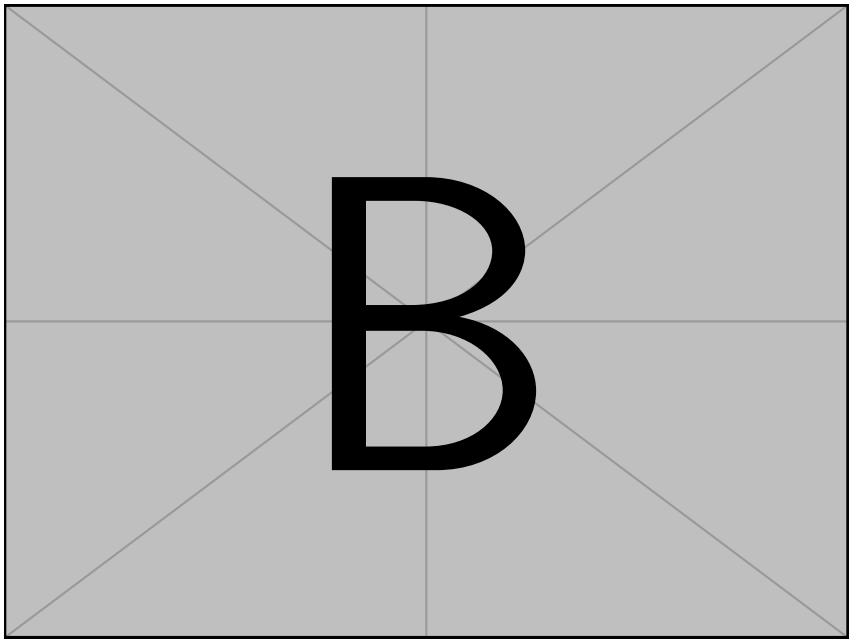
Initial state image

Complex Temporal Example



Processing...

Complex Temporal Example



Processing...

Code Reveal

Python Function

```
def fibonacci(n):
```

Code Reveal

Python Function

```
def fibonacci(n):  if n <= 1:
```

- Base case check

Code Reveal

Python Function

```
def fibonacci(n):  if n <= 1:  return n
```

- Base case check
- Return for base cases

Code Reveal

Python Function

```
def fibonacci(n):  if n <= 1:  return n  else:
```

- Base case check
- Return for base cases

Code Reveal

Python Function

```
def fibonacci(n):  if n <= 1:  return n  else:
return fibonacci(n-1) + fibonacci(n-2)
```

- Base case check
- Return for base cases
- Recursive calls

Mathematical Proof

Theorem

$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$

Proof.

We'll prove this by induction.

□

Mathematical Proof

Theorem

$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$

Proof.

Base case: $n = 1$

$$\sum_{i=1}^{1} i = 1 = \frac{1(1+1)}{2} = 1\checkmark$$

Mathematical Proof

Theorem

$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$

Proof.

Inductive step: Assume true for $n = k$

$$\sum_{i=1}^{k} i = \frac{k(k+1)}{2}$$

Mathematical Proof

Theorem

$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$

Proof.

Show true for $n = k + 1$:

$$\sum_{i=1}^{k+1} i = \sum_{i=1}^{k} i + (k+1) \tag{1}$$

$$= \frac{k(k+1)}{2} + (k+1) \tag{2}$$

$$= \frac{k(k+1) + 2(k+1)}{2} \tag{3}$$

$$(k+1)(k+2)$$

Mode-Specific Content

Presentation Mode

This content only appears in presentation mode.

Mode-Specific Content

Presentation Mode

This content only appears in presentation mode.

- Interactive elements

This appears on slide 2+ in presentation, always in handout.

Mode-Specific Content

Presentation Mode

This content only appears in presentation mode.

- Interactive elements
- Animations

This appears on slide 2+ in presentation, always in handout.

Mode-Specific Content

Presentation Mode

This content only appears in presentation mode.

- Interactive elements
- Animations
- Step-by-step reveals

This appears on slide 2+ in presentation, always in handout.

Summary of Overlay Commands

\pause Simple sequential reveal

Summary of Overlay Commands

\pause Simple sequential reveal

\only Content replacement

Summary of Overlay Commands

\pause	Simple sequential reveal
\only	Content replacement
\uncover	Space-preserving reveal

Summary of Overlay Commands

\pause	Simple sequential reveal
\only	Content replacement
\uncover	Space-preserving reveal
\alert	Dynamic highlighting

Summary of Overlay Commands

\pause	Simple sequential reveal
\only	Content replacement
\uncover	Space-preserving reveal
\alert	Dynamic highlighting
\temporal	Three-state transitions

Summary of Overlay Commands

\pause	Simple sequential reveal
\only	Content replacement
\uncover	Space-preserving reveal
\alert	Dynamic highlighting
\temporal	Three-state transitions
¡+-¿	Automatic incrementing

Best Practices

Do

- Use overlays purposefully

Don't

Best Practices

Do

- Use overlays purposefully
- Keep animations smooth

Don't

Best Practices

Do

- Use overlays purposefully
- Keep animations smooth
- Test in presentation mode

Don't

Best Practices

Do

- Use overlays purposefully
- Keep animations smooth
- Test in presentation mode
- Provide handout version

Don't

Best Practices

Do

- Use overlays purposefully
- Keep animations smooth
- Test in presentation mode
- Provide handout version

Don't

- Overuse animations

Best Practices

Do

- Use overlays purposefully
- Keep animations smooth
- Test in presentation mode
- Provide handout version

Don't

- Overuse animations
- Make content too complex

Best Practices

Do

- Use overlays purposefully
- Keep animations smooth
- Test in presentation mode
- Provide handout version

Don't

- Overuse animations
- Make content too complex
- Forget about timing

Best Practices

Do

- Use overlays purposefully
- Keep animations smooth
- Test in presentation mode
- Provide handout version

Don't

- Overuse animations
- Make content too complex
- Forget about timing
- Ignore accessibility

Thank You!

Questions?

Thank You!

Questions?