Guide: Transition Engine

1 Introduction

The `render-transition` function is designed to automatically generate "roadmap" or "summary" slides when the document structure changes (e.g., entering a new section). It is typically invoked within a `show heading` rule.

2 Function documentation

`render-transition(h, transitions: (:), mapping: (:), ...)`

2.1 Parameters Reference

Option	Type	Description
`h`	heading	**Mandatory**. The heading object intercepted by the show rule.
`slide-func`	function	**Mandatory**. A callback (`fill: color, body: content) => content` used to create the slide. It should wrap the presentation engine's slide function (e.g., `polylux.slide`).
`transitions`	dict	Detailed configuration for the transition engine (see below).
`mapping`	dict	Maps heading levels to roles. Example: (`section: 1, subsection: 2`).
`theme-colors`	dict	Dictionary containing `primary` and `accent` colors used for the transition slide style.
`show-heading-numbering`	bool	Whether to display heading numbers in the roadmap. Default: `true`.
`numbering-format`	string | auto	Typst numbering format string. If `auto`, uses the document's global heading numbering. Default: `auto`.
`base-text-size`	length | auto	Base font size for the roadmap text. Default: `auto`.
`base-text-font`	string | auto	Font family for the roadmap text. Default: `auto`.
`top-padding`	relative | length	Vertical spacing added above the roadmap on the transition slide. Default: `40%`.
`content-padding`	length | dict	Padding around the roadmap content. Default: (`x: 10%`).
`content-align`	alignment	Alignment of the roadmap content. Default: `top + left`.
`content-wrapper`	function	A callback (`roadmap, heading, active-state) => content` to completely override the layout logic.

2.2 The `transitions` dictionary

This parameter allows fine-tuning the behavior and appearance of transition slides.

Key	Type	Description
`enabled`	bool	Global switch for transitions. Default: `true`.
`max-level`	int	The maximum heading level that triggers a transition. Default: `3`.

Key	Type	Description
`background`	color | string	Background type: `"theme"` (uses `primary` color), `"none"`, or an explicit color.
`filter`	function	A callback `(heading) => bool` to programmatically enable/disable transitions for specific headings.
`style`	dict	Controls typography: `inactive-opacity` (default `0.3`), `completed-opacity` (default `0.6`), `active-weight` (default `"bold"`).
`sections`	dict	Override for section-level transitions. Contains `enabled`, `visibility` (dict), and `background`.
`subsections`	dict	Override for subsection-level transitions. Contains `enabled`, `visibility` (dict), and `background`.

2.2.1 Visibility Logic

For each transition role (`parts`, `sections`, `subsections`), you can define which hierarchy levels are visible using the `visibility` key:

- `"all"`: Show all headings at this level.
- `"current"`: Only show the active heading at this level.
- `"current-parent"`: Show siblings of the active heading.
- `"none"`: Hide this level.

3 Basic usage

```
render-transition(
  h,
  mapping: (section: 1, subsection: 2),
  theme-colors: (primary: navy),
  slide-func: my-slide-func,
)
```

Default Transition

2 Function documentation
2.1 Parameters Reference
2.2 The `transitions` dictionary

4 Layout & Positioning Control

4.1 Padding and Alignment

Use `content-padding` and `content-align` to place the roadmap exactly where you want it.

```
render-transition(
  h,
  top-padding: 0pt,
  content-align: center + horizon,
  content-padding: 0pt,
  slide-func: my-slide-func,
)
```

Centered Roadmap

1 Introduction

4.2 The Wrapper Mode (Total Control)

The `content-wrapper` parameter offers complete freedom over the slide layout. It allows you to place the roadmap alongside other elements (images, titles, decorations) or to wrap it in complex layouts.

The callback function receives three arguments:

1. `roadmap`: The progressive outline component generated by Navigator.
2. `heading`: The specific heading object that triggered this transition.
3. `active-state`: A dictionary `{ h1, h2, h3 }` containing the active hierarchy (useful to retrieve the parent Part title when entering a Section).

4.2.1 Example 1: Split Layout (Title + Roadmap)

In this example, we place the current section title on the left and the roadmap on the right. We also disable the section display in the roadmap to avoid repetition.

```
content-wrapper: (roadmap, h, active) => {
  grid(
    columns: (1fr, 1.5fr),
    align(center + horizon,
      text(size: 1.5em, weight: 'bold',
format-heading(h))
    ),
    align(left + horizon, roadmap)
  )
},
transitions: (
  sections: (visibility: (section: 'none',
subsection: 'all'))
)
```

Split Layout

2 Function documentation

2.1 Parameters Reference
2.2 The transitions dictionary

4.2.2 Example 2: Contextual Header

Use `active-state` to display the parent heading (e.g., the Part) above the roadmap.

```
content-wrapper: (roadmap, h, active) => {
  set align(left + top)
  if active.h1 != none {
    text(size: 0.8em, fill:
white.transparentize(40%),
smallcaps(active.h1.body))
    v(0.5em)
  }
  roadmap
}
```

Contextual Info

Function documentation

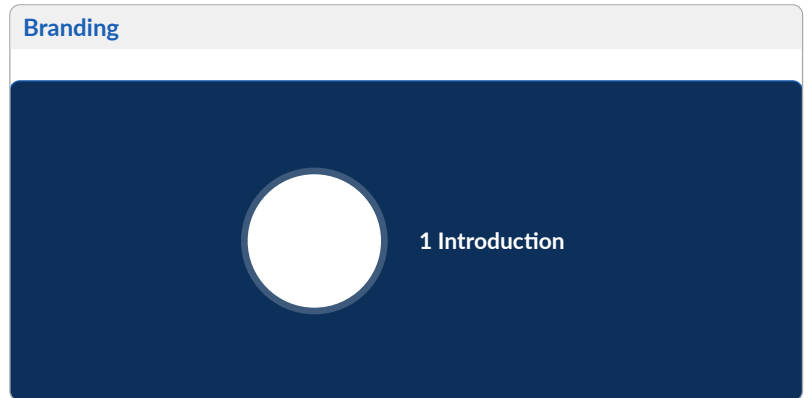**2 Function documentation**
  2.1 Parameters Reference
  2.2 The transitions dictionary

4.2.3 Example 3: Branding (Logo + Roadmap)

You can easily integrate graphical elements.

```
content-wrapper: (roadmap, h, active) => {
  stack(dir: ltr, spacing: 2em,
    align(horizon, image('logo.png')),
    align(horizon, roadmap)
  )
}
```

Branding

1 Introduction

5 Advanced Customization

5.1 Selective visibility

You can configure different roadmap layouts for sections and subsections.

```
transitions: (
  subsections: (
    visibility: (
      section: 'current',
      subsection: 'current-parent'
    )
  )
)
```

Subsection roadmap

2 Function documentation
  2.1 Parameters Reference
  2.2 The `transitions` dictionary