Presentate: Themes Reference Manual

A comprehensive guide to the built-in presentation themes

Contents

1 Introduction

Presentate offers a suite of five high-level themes, each designed to handle document structure and transitions automatically. Unlike basic layouts, these themes are aware of your document's hierarchy (Parts, Sections, Subsections) and use a **Unified Transition Engine** to generate roadmap slides between content blocks.

This manual describes the shared configuration interface, the transition engine logic, and provide an exhaustive reference for each theme's specific parameters.

2 Global Configuration

All themes follow a consistent API pattern. They are invoked using a `template` function through a show rule.

```
#import "@preview/presentate:0.2.3": themes
#show: themes.sidebar.template.with(
  title: [My Presentation],
  author: [John Doe],
  // ... options
)
```

2.1 Common Parameters

The following parameters are supported by all five themes:

Parameter	Type	Description
`title`	content	The main title displayed on the title slide.
`subtitle`	content	Displayed below the title on the title slide (optional).
`author`	content	The name of the presenter.
`date`	content	Presentation date. Defaults to `#datetime.today()`.
`text-font`	string	Font family family for the whole document. Default: "Lato".
`text-size`	length	Base text size. Default: `20pt`.
`mapping`	dictionary	Maps logical roles to Typst heading levels. Default: `(section: 1, subsection: 2)`. Supports `part`, `section`, `subsection`.
`auto-title`	bool	If `true`, slides without a manual title will automatically display the current heading body. Default: `false`.
`transitions`	dictionary	Configuration for the transition engine (see Section 3).

2.2 Structural Mapping

The `mapping` dictionary is crucial as it tells the theme which headings should trigger transitions and navigation updates.

- `part`: Typically mapped to level 1 (=).
- `section`: Typically mapped to level 1 or 2.
- `subsection`: Typically mapped to level 2 or 3.

Example:

```
mapping: (part: 1, section: 2, subsection: 3)
```

3 The Unified Transition Engine

The transition engine automatically generates "road-map" slides when your structure changes. It is highly configurable via the `transitions` argument.

3.1 Configuration Options

The `transitions` dictionary accepts the following keys:

Key	Type	Description
`enabled`	bool	Global switch for transitions. Default: `true`.
`max-level`	int	The maximum Typst heading level (1-3) that triggers a transition. Default: `3`.

Key	Type	Description
show-numbering	bool	Display heading numbers in transition outlines. Default: `true`.
background	color | str	Background of transition slides. `"theme"` (primary color), `"none"` (white), or an explicit color.
filter	function	A callback `(heading) => bool`. Headings returning `false` will not trigger slides and will be hidden from outlines.
style	dictionary	Global styling for the transition outline (see below).
parts	dictionary	Override settings for Part transitions.
sections	dictionary	Override settings for Section transitions.
subsections	dictionary	Override settings for Subsection transitions.

3.2 Transition Style

The `transitions.style` dictionary controls the typography of the roadmap:
- `active-color`: Color of the current heading. Defaults to theme accent.
- `active-weight`: Font weight of the active heading. Default: `"bold"`.
- `inactive-opacity`: Opacity (0.0 to 1.0) for future headings. Default: `0.3`.
- `completed-opacity`: Opacity for past headings. Default: `0.6`.

3.3 Visibility Logic

For each roadmap type (`parts`, `sections`, `subsections`), you can define which levels are visible via 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 (e.g., all subsections of the current section).
- `"none"`: Hide this level.

Default Behavior:
- Part transitions: Show all Parts, hide others.
- Section transitions: Show current Part, all Sections, current Subsections.

4 Theme Reference

4.1 Sidebar Theme

The `sidebar` theme provides a persistent navigation bar on the left or right side.

4.1.1 Theme Parameters

side	string	`"left"` or `"right"`. Default: `"left"`.
width	ratio	Width of the sidebar. Default: `22%`.
sidebar-color	color	Background color of the sidebar.
main-color	color	Background color of the content area. Default: `white`.

`active-color`	color	Highlight color for the current item in the sidebar.
`logo`	content	An image or shape to display in the sidebar.
`logo-position`	string	`"top"` or `"bottom"`. Default: `"top"`.
`outline-options`	dictionary	Advanced parameters for the sidebar outline component.

4.2 Miniframes Theme

Inspired by the Beamer Berlin theme, it uses "dots" to show progress within sections.

4.2.1 Navigation Dictionary

The `navigation` parameter is a dictionary containing:

`position`	string	`"top"` or `"bottom"`. Default: `"top"`.
`style`	string	`"compact"` (one row) or `"grid"` (one row per subsection).
`marker-shape`	string	`"circle"`, `"square"`, `"diamond"`, `"rect"`.
`marker-size`	length	Size of the dots. Default: `4pt`.
`fill`	color	Color of the navigation bar block.
`show-level1-titles`	bool	Show names of sections in the bar. Default: `true`.

4.3 Split Theme

Features a horizontal header divided into two contrasting areas for Section and Subsection titles.

4.3.1 Theme Parameters

`primary`	color	Background color for the Section block.
`secondary`	color	Background color for the Subsection block.
`navigation-style`	string	`"all"` (show all siblings) or `"current"`.
`header-columns`	tuple	Width ratios for the header blocks. Default: `(1fr, 1fr)`.
`section-align`	alignment	Horizontal alignment for section text. Default: `right`.

4.4 Progressive Outline Theme

A clean theme focused on document progression with a breadcrumb-style header.

4.4.1 Theme Parameters

`header`	content	auto	Override the top breadcrumb fil d'ariane.
`footer`	content	auto	Override the bottom page number area.

4.5 Custom Transition Theme

A minimalist theme with no default navigation, designed for presenters who want full control over the content area while still utilizing the transition engine.

5 Hooks API

If you need to generate a transition slide that doesn't fit the roadmap pattern (e.g., a full-page image), use hooks. They intercept the transition engine.

- `on-part-change(heading)`
- `on-section-change(heading)`
- `on-subsection-change(heading)`

Example:

```
#show: template.with(
  on-section-change: (h) => {
    empty-slide(fill: black)[
      #set align(center + horizon)
      #text(white, size: 2em)[Next Chapter: #h.body]
    ]
  }
)
```