

# Guide: Miniframes Navigation

## 1 Introduction

The `render-miniframes` function generates a navigation bar showing the progress through sections and subsections using “miniframes” (dots or squares).

Previously, it required manual structure extraction. Now, it is designed to be **configured globally** via `navigator-config`, allowing a clean one-line integration in your document.

## 2 Global Configuration

Instead of passing parameters to every `render-miniframes` call, you can set them once at the beginning of your document:

```
#import "@preview/navigator:0.2.0": navigator-config, render-miniframes

#navigator-config.update(c => {
  c.slide-selector = metadata.where(value: (t: "ContentSlide"))
  c.miniframes = (
    fill: navy,
    active-color: white,
    inactive-color: gray,
    style: "compact"
  )
  c
})

// Now you can use a simple one-liner in your header/footer:
#set page(header: context render-miniframes())
```

## 3 Function documentation

`render-miniframes(structure, current-slide-num, ...)`

### 3.1 Parameters Reference

Most parameters default to `auto`, which means they will be resolved from the global `navigator-config` state.

Option	Type	Effect & Expected Values
<code>structure</code>	array   auto	The presentation structure. If <code>auto</code> , resolved automatically from global <code>slide-selector</code> .
<code>current-slide-num</code>	int   auto	The index of the active slide. If <code>auto</code> , resolved automatically.
<code>style</code>	string   auto	Layout mode: <code>"compact"</code> or <code>"grid"</code> . Defaults to global config.
<code>marker-shape</code>	string	<code>"circle"</code> (default) or <code>"square"</code> .
<code>marker-size</code>	length	Diameter/width of the markers. Default: <code>4pt</code> .
<code>active-color</code>	color   auto	Color of the current slide's marker. Defaults to global config.
<code>inactive-color</code>	color   auto	Color of future slides' markers. Defaults to global config.
<code>fill</code>	color   auto	Background color of the bar block. Defaults to global config.
<code>text-color</code>	color   auto	Color of titles. Defaults to contrast with <code>fill</code> .
<code>text-size</code>	length	Size of the titles. Default: <code>10pt</code> .

Option	Type	Effect & Expected Values
font	string   none	Font family for titles. Uses document default if none.
align-mode	string	Global horizontal alignment of the block: "left", "center", "right".
dots-align	string	Alignment of the dots <b>within</b> their section column: "left", "center", "right".
navigation-pos	string	Vertical position of dots relative to titles: "top" or "bottom". Default: "bottom".
show-level1-titles	bool	Whether to display the names of sections.
show-level2-titles	bool	In grid mode, whether to display subsection names.
show-numbering	bool	Whether to display heading numbers. Default: false.
numbering-format	string	Typst numbering format string (e.g., "1.1"). Default: "1.1".
gap	length	Horizontal space between sections. Default: 1.5em.
line-spacing	length	Vertical space between titles and dots. Default: 4pt.
inset	dict   length	Internal padding of the bar block. Default: (x: 1em, y: 0.5em).
radius	length   dict	Corner rounding of the background block. Default: 0pt.
width	length	Total width of the block. Default: 100%.
outset-x	length	Horizontal bleed. Useful to make the bar touch page edges.
max-length	int   dict   auto	Maximum length before truncation. Defaults to global config.
use-short-title	bool   dict   auto	Whether to use short titles. Defaults to global config.

## 4 Structure Extraction

To work, the navigation bar needs to know the presentation structure. Two functions are provided to extract this data from metadata markers. These are called automatically by `render-miniframes()` if arguments are set to auto.

### 4.1 get-structure

```
get-structure(slide-selector: auto, filter-selector: none)
```

Scans the document for headings and slide markers. Returns a structure dictionary.

### 4.2 get-current-logical-slide-number

```
get-current-logical-slide-number(slide-selector: auto, filter-selector: none)
```

Determines the index of the current slide relative to the extracted structure.

### 4.3 Selection Parameters

Option	Type	Description
slide-selector	selector   auto	The metadata type used to identify slides. Default is (t: "LogicalSlide"). Useful for custom engines (e.g., Polylux).

Option	Type	Description
filter-selector	selector   none	If provided, only pages containing this selector will be counted. Useful to exclude transition slides that might share the same slide metadata.

## 5 Function Signature

```
render-miniframes(structure, current-slide-num, ...)
```

### 5.1 The structure object

The structure argument is an array of section dictionaries. Each section has the following schema:

- **Section:** (title: content, loc: location, subsections: array)
- **Subsection:** (title: content, loc: location, slides: array) OR (title: content, loc: location, subsections: array) (if 3 levels are used).
- **Slide:** (number: integer, loc: location)

#### 5.1.1 What is loc?

The loc field expects a Typst **location** object.

- **Purpose:** It defines the destination for navigation links. If a valid location is provided, clicking on the section title or the dot will take the user to that specific position in the PDF.
- **Disabling links:** If set to none, the element will be displayed normally but will not be clickable. This is used in the mock data of this guide.

### 5.2 The current-slide-num argument

An integer representing the current slide number. The function compares this value with the number field of each slide in the structure to determine its state:

- **Active:** slide.number == current-slide-num
- **Completed:** slide.number < current-slide-num
- **Future:** slide.number > current-slide-num

## 6 Basic usage

By default, the navigation bar uses the "grid" style and shows section titles.

```
render-miniframes(structure, 4, use-short-title: true)
```

Default Grid Style

1. Intro	2. Methods	3. Results
1.1 Context	• •	2.1 Raw Data
1.2 Goals	•	2.2 Tools

## 7 Short Titles & Truncation

Like progressive-outline, Miniframes supports short titles and truncation. This is crucial for navigation bars which have limited horizontal space.

### 7.1 Collecting Short Titles

You must pass the short titles to the structure extractor.

```
#let struct = get-structure(
  all-shorts: query(<short>)
)
```

### 7.2 Truncation & Short Titles

Then, configure the rendering. By default, use-short-title is false.

```
render-miniframes(structure, 4)
```

### 1. Original Titles (Default)

- |  |   |                                 |
|--|---|---------------------------------|
| 1. General<br>Introduction to<br>the Project<br>1.1 Background<br>Context<br>1.2 Objectives<br>and Goals | 2. Scientific<br>Methodology<br>and Data<br>Collection<br>2.1 Raw Data<br>2.2 Analysis<br>Tools | 3. Preliminary<br>Results<br>•• |
|--|---|---------------------------------|

```
render-miniframes(  
  structure, 4,  
  max-length: 12  
)
```

### 2. Automatic Truncation

- |                 |                 |                 |
|-----------------|-----------------|-----------------|
| General Intr... | Scientific M... | Preliminary ... |
| 1.1 Backgrou... | 2.1 Raw Data    | •••             |
| 1.2 Objectiv... | 2.2 Analysis    | •               |

```
render-miniframes(  
  structure, 4,  
  use-short-title: true  
)
```

### 3. Manual Short Titles

- |                                      |   |                   |
|--------------------------------------|---|-------------------|
| 1. Intro<br>1.1 Context<br>1.2 Goals | 2. Methods<br>2.1 Raw Data<br>2.2 Tools | 3. Results<br>••• |
|--------------------------------------|---|-------------------|

```
render-miniframes(  
  structure, 4,  
  use-short-title: true,  
  max-length: 8  
)
```

### 4. Combined: Short Titles + Truncation

- |                                     |                                       |                |
|-------------------------------------|---------------------------------------|----------------|
| Intro<br>1.1 Cont...<br>1.2 Goal... | Methods<br>2.1 Raw ...<br>2.2 Tool... | Results<br>••• |
|-------------------------------------|---------------------------------------|----------------|

## 8 Layout Styles

### 8.1 Compact Mode

In "compact" mode, all slide markers of a section are grouped on a single line, regardless of subsections. This is useful for saving space in the header or footer.

```
render-miniframes(  
  structure, 4,  
  style: 'compact',  
  use-short-title: true  
)
```

#### Compact Mode

- |                 |                    |                  |
|-----------------|--------------------|------------------|
| 1. Intro<br>••• | 2. Methods<br>•••• | 3. Results<br>•• |
|-----------------|--------------------|------------------|

### 8.2 Grid Mode

The "grid" style is ideal for presentations with many subsections, as it aligns them vertically.

```
render-miniframes(  
  structure, 4,  
  style: 'grid',  
  show-level2-titles: true,  
  use-short-title: true  
)
```

#### Grid Mode with Titles

- |                                      |   |                   |
|--------------------------------------|---|-------------------|
| 1. Intro<br>1.1 Context<br>1.2 Goals | 2. Methods<br>2.1 Raw Data<br>2.2 Tools | 3. Results<br>••• |
|--------------------------------------|---|-------------------|

## 8.3 Hiding Titles

You can hide titles at different levels to obtain a minimalist bar.

```
render-miniframes(  
  structure, 4,  
  style: 'grid',  
  show-level2-titles: false,  
  use-short-title: true  
)
```

### Hiding Subsection Titles (Grid)

1. Intro    2. Methods    3. Results  


```
render-miniframes(  
  structure, 4,  
  show-level1-titles: false,  
  use-short-title: true  
)
```

### Hiding Section Titles (Dots Only)

1.1 Context    1.2 Goals    2.1 Raw Data    2.2 Tools    ...  


## 9 Customization

### 9.1 Markers

Change the shape and size of the progress indicators.

```
render-miniframes(  
  structure, 4,  
  marker-shape: 'square',  
  marker-size: 6pt,  
  use-short-title: true  
)
```

### Square Markers

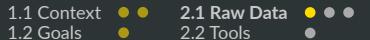
1. Intro    1.1 Context    2. Methods    2.1 Raw Data    3. Results  
1.2 Goals    2.2 Tools  


### 9.2 Colors & Typography

Fine-tune the appearance of markers and labels.

```
render-miniframes(  
  structure, 4,  
  active-color: yellow,  
  inactive-color: gray,  
  text-color: luma(200),  
  text-size: 8pt,  
  fill: rgb('#2d3436'),  
  use-short-title: true  
)
```

### Colors & Fonts

1. Intro    1.1 Context    2. Methods    2.1 Raw Data    3. Results  
1.2 Goals    2.2 Tools  


### 9.3 Alignment & Spacing

Control the rhythm and positioning of the navigation elements.

```
render-miniframes(  
  structure, 4,  
  align-mode: 'center',  
  dots-align: 'center',  
  gap: 3em,  
  line-spacing: 8pt,  
  use-short-title: true  
)
```

### Centered & Airy

1. Intro    1.1 Context    2. Methods    2.1 Raw Data    3. Results  
1.2 Goals    2.2 Tools  


### 9.4 Advanced Layout

Use inset and width to integrate the bar into specific layout zones.

```
render-miniframes(  
  structure, 4,  
  width: 60%,  
  align-mode: 'center',  
  inset: 15pt,  
  show-level1-titles: false,  
  use-short-title: true  
)
```

### Compact Centered Bar



```
render-miniframes(  
  structure, 4,  
  radius: 10pt,  
  fill: rgb('#34495e'),  
  inset: (x: 2em, y: 1em),  
  use-short-title: true  
)
```

### Rounded Corners

