

Authoring Essentials

Session 2

Mickaël CANOUIL, *Ph.D.* 

pro@mickael.canouil.dev

Independent Contractor

Saturday, the 7th of February, 2026

- Session 2: Authoring Essentials
- Session Objectives
- Learning Objectives

Session Overview

Session 2: Authoring Essentials

- **Markdown Fundamentals**

- Creating structured documents: headings, lists, links, and text formatting.
- Best practices for writing clean, reproducible content in Quarto.
- Working with code blocks, tables, and mathematical expressions.

- **YAML Configuration**

- Document headers and multi-format output.
- Metadata inheritance and project configuration.
- Extensions and customisation options.

- **Quarto Markdown**

- Callout blocks, shortcodes, and Mermaid diagrams.
- Cross-references for figures, tables, and custom elements.
- Code annotations and advanced formatting features.

Session Objectives

This session focuses on building foundational skills in Quarto authoring, covering enhanced markdown capabilities, YAML configuration, and document structuring best practices.

Learning Objectives

Learning Objectives

By the end of this session, participants will be able to:

Learning Objectives

By the end of this session, participants will be able to:

- Create well-structured documents using enhanced Quarto markdown features.

Learning Objectives

By the end of this session, participants will be able to:

- Create well-structured documents using enhanced Quarto markdown features.
- Configure documents using YAML headers for different output formats.

Learning Objectives

By the end of this session, participants will be able to:

- Create well-structured documents using enhanced Quarto markdown features.
- Configure documents using YAML headers for different output formats.
- Use advanced formatting features including callouts, cross-references, and tabsets.

- Text Formatting Essentials
- Headers Structure Your Content
- Lists
- Lists - Nested
- Working with Code
- Tables Made Simple
- Mathematical Expressions
- Footnotes and References
- Figure Basics
- Advanced Formatting: Divs and Spans

Markdown Basics

Text Formatting Essentials

Text Formatting Essentials

Basic text formatting you'll use every day:

Text Formatting Essentials

Basic text formatting you'll use every day:

- **Bold text** with `**bold**` or `__bold__`.
- *Italic text* with `*italic*` or `_italic_`.
- ***Bold and italic*** with `***text***`.
- Superscript² with `superscript^2^`.
- Subscript₂ with `subscript~2~`.
- Verbatim code with backticks.

Headers Structure Your Content

Headers Structure Your Content

Headers create your document structure and enable automatic table of contents generation.

● ● ●

MARKDOWN

```
1 # Heading 1 - Document Title
2 ## Heading 2 - Major Sections
3 ### Heading 3 - Subsections
4 #### Heading 4 - Minor Sections
5 ##### Heading 5 - Rare Usage
6 ##### Heading 6 - Very Rare
```

Headers Structure Your Content

Headers create your document structure and enable automatic table of contents generation.

MARKDOWN

```
1 # Heading 1 - Document Title
2 ## Heading 2 - Major Sections
3 ### Heading 3 - Subsections
4 #### Heading 4 - Minor Sections
5 ##### Heading 5 - Rare Usage
6 ##### Heading 6 - Very Rare
```



Tip

Add a empty lines before and after!

Lists

Lists

Unordered Lists:

MARKDOWN

1 - First item

2 - Second item

3 - Third item

Lists

Unordered Lists:



MARKDOWN

- 1 - First item
- 2 - Second item
- 3 - Third item



Ordered Lists:



MARKDOWN

- 1 1. First step
- 2 2. Second step
- 3 3. Third step



Lists

! Important

Quarto requires empty lines before and after lists to ensure proper rendering!

Unordered Lists:



MARKDOWN

- 1 - First item
- 2 - Second item
- 3 - Third item



Ordered Lists:



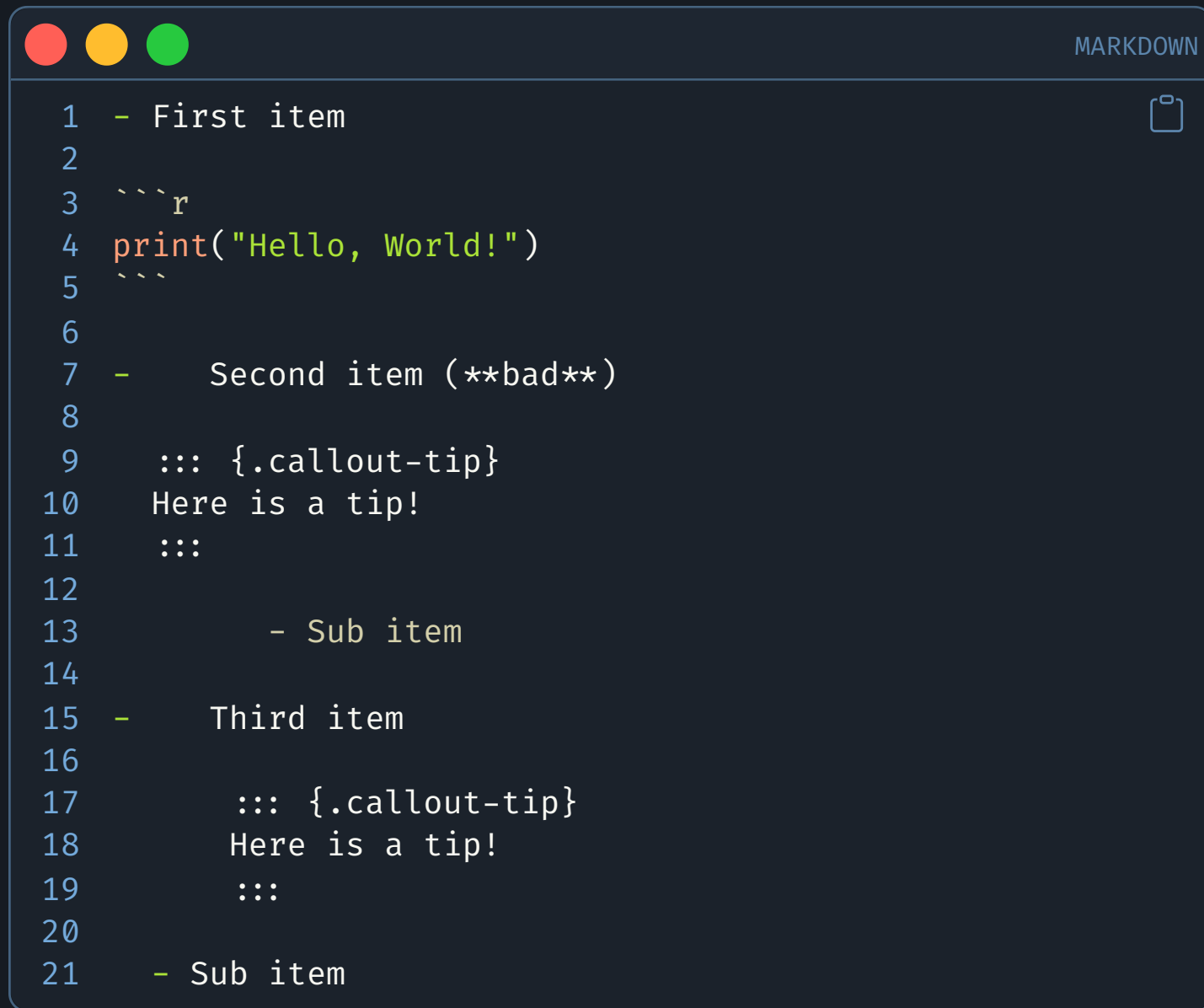
MARKDOWN

- 1 1. First step
- 2 2. Second step
- 3 3. Third step



Lists - Nested

Lists - Nested



```
1 - First item
2
3 ```r
4 print("Hello, World!")
5 ```
6
7 - Second item (**bad**)
8
9 ::: {.callout-tip}
10 Here is a tip!
11 :::
12
13     - Sub item
14
15 - Third item
16
17     ::: {.callout-tip}
18     Here is a tip!
19     :::
20
21 - Sub item
```

Lists - Nested

! Important

The number of spaces for indentation does not matter, but the list marker must be aligned with the first character of the parent item. Items marked **bad** below show incorrect alignment of nested content.



MARKDOWN

```

1 - First item
2
3 ```r
4 print("Hello, World!")
5 ```
6
7 - Second item (**bad**)
8
9   ::: {.callout-tip}
10  Here is a tip!
11   :::
12
13     - Sub item
14
15 - Third item
16
17   ::: {.callout-tip}
18   Here is a tip!
19   :::
20
21 - Sub item
  
```



MARKDOWN

```

1 - First item
2
3 ```r
4 print("Hello, World!")
5 ```
6
7 - Second item (**bad**)
8
9   ::: {.callout-tip}
10  Here is a tip!
11   :::
12
13 - Sub item
14
15 - Third item
16
17   ::: {.callout-tip}
18   Here is a tip!
19   :::
20
21 - Sub item
  
```

Working with Code

Working with Code

Inline code: Use single backticks

MARKDOWN

```
1 The `print()` function outputs text.
```

Working with Code

Inline code: Use single backticks

MARKDOWN

```
1 The `print()` function outputs text.
```



Tip

Code highlighting for inline code:

MARKDOWN

```
1 The `print()`{.python} function outputs text.
```

Working with Code

Inline code: Use single backticks

● ● ●

MARKDOWN

```
1 The `print()` function outputs text.
```

📄



Tip

Code highlighting for inline code:

● ● ●

MARKDOWN

```
1 The `print()`{.python} function outputs text.
```

📄

Code blocks: Use triple backticks with optional language

● ● ●

MARKDOWN

```
1 ```python
2 def hello_world():
3     print("Hello, Quarto!")
4 ```
```

📄

Working with Code

Inline code: Use single backticks

MARKDOWN

```
1 The `print()` function outputs text.
```



Tip

Code highlighting for inline code:

MARKDOWN

```
1 The `print()`{.python} function outputs text.
```

Code blocks: Use triple backticks with optional language

MARKDOWN

```
1 ```python
2 def hello_world():
3     print("Hello, Quarto!")
4 ```
```



Note

The syntax highlighting is provided by [skylighting](#) .

Tables Made Simple

Tables Made Simple

MARKDOWN

1	Default	Left	Right	Center
2	: : :			
3	12	12	12	12
4	123	123	123	123
5	1	1	1	1
6				
7	: Demonstration of pipe table syntax			

Tables Made Simple

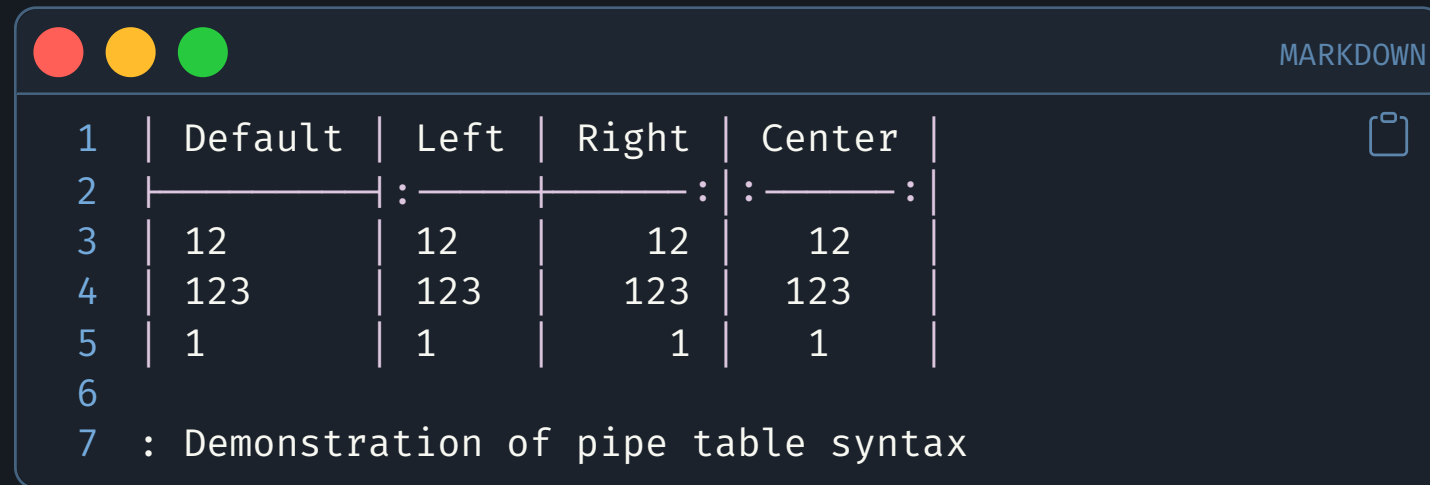
MARKDOWN

1	Default	Left	Right	Center
2	:-----: :-----: :-----:			
3	12	12	12	12
4	123	123	123	123
5	1	1	1	1
6				
7	: Demonstration of pipe table syntax			

Demonstration of pipe table syntax

Default	Left	Right	Center
12	12	12	12
123	123	123	123
1	1	1	1

Tables Made Simple



1	Default	Left	Right	Center
2	:	:	:	:
3	12	12	12	12
4	123	123	123	123
5	1	1	1	1
6				
7	: Demonstration of pipe table syntax			

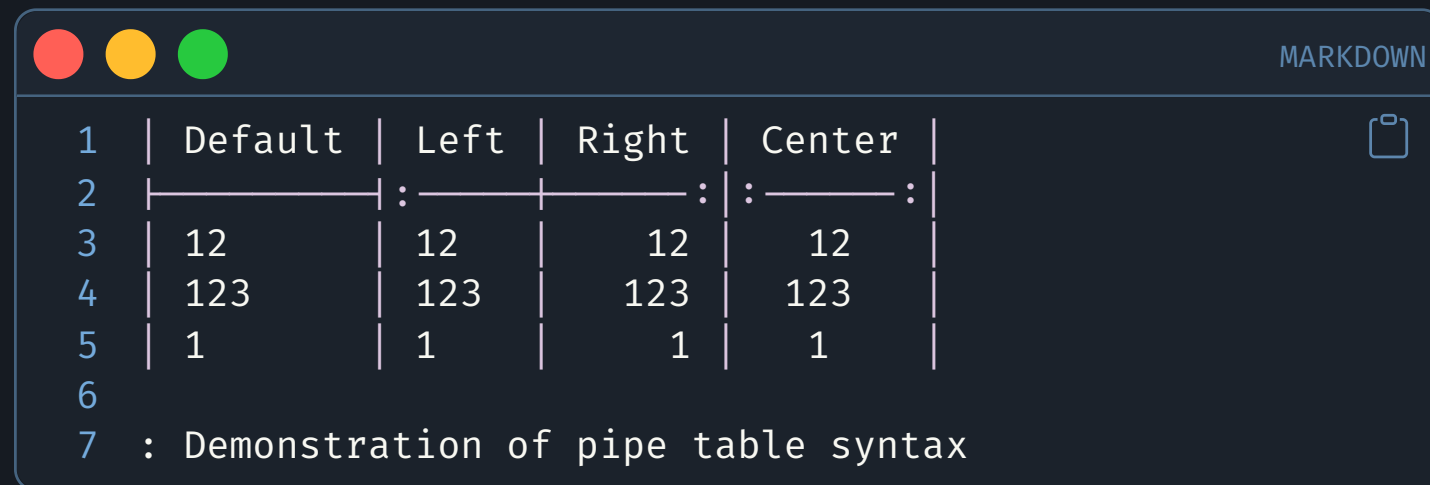
Demonstration of pipe table syntax

Default	Left	Right	Center
12	12	12	12
123	123	123	123
1	1	1	1

Use colons (:) to control alignment:

- : --- for left alignment.
- --- : for right alignment.
- : --- : for centre alignment.

Tables Made Simple



1	Default	Left	Right	Center
2	:	:	:	:
3	12	12	12	12
4	123	123	123	123
5	1	1	1	1
6				
7	: Demonstration of pipe table syntax			

Demonstration of pipe table syntax

Default	Left	Right	Center
12	12	12	12
123	123	123	123
1	1	1	1

Use colons (`:`) to control alignment:

- `: ---` for left alignment.
- `--- :` for right alignment.
- `: --- :` for centre alignment.

See [Table basics](#) for more.

Mathematical Expressions

Mathematical Expressions

Inline maths: `$E = mc^2$` renders as $E = mc^2$.

Mathematical Expressions

Inline maths: `$E = mc^2$` renders as $E = mc^2$.

Display maths: Use double dollar signs (on separate lines).

MARKDOWN

```
1 $$  
2 E = mc^2  
3 $$
```

Mathematical Expressions

Inline maths: `$E = mc^2$` renders as $E = mc^2$.

Display maths: Use double dollar signs (on separate lines).

● ● ●

MARKDOWN

1

2

3

```
$$  
E = mc^2  
$$
```

Define custom LaTeX macros in hidden blocks for reuse across your document, see [Markdown Basics - Equations](#)

Footnotes and References

Footnotes and References

Standard footnotes:



MARKDOWN

```
1 Here is a footnote reference[^1].  
2  
3 [^1]: Here is the footnote content.
```



Footnotes and References

Standard footnotes:



MARKDOWN

```
1 Here is a footnote reference[^1].  
2  
3 [^1]: Here is the footnote content.
```



Inline footnotes:



MARKDOWN

```
1 Here is an inline note^[Much easier to write!].
```



Footnotes and References

Standard footnotes:



MARKDOWN

```
1 Here is a footnote reference[^1].  
2  
3 [^1]: Here is the footnote content.
```



Inline footnotes:



MARKDOWN

```
1 Here is an inline note^[Much easier to write!].
```



⚠ Important

Footnote identifiers must be unique within the document.

Figure Basics

Figure Basics

- Basic

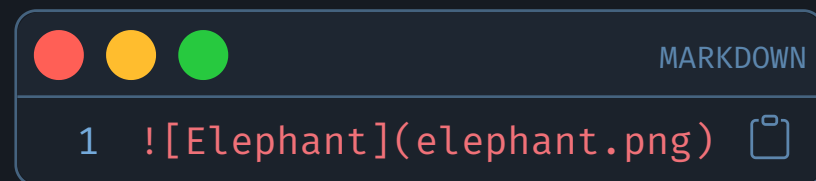


Figure Basics

- Basic

● ● ●

MARKDOWN

```
1 ![Elephant](elephant.png) 📄
```

- Size and alignment

● ● ●

MARKDOWN

```
1 ![Elephant](elephant.png){width=300} 📄
2 ![Elephant](elephant.png){width=80%}
3 ![Elephant](elephant.png){width=4in}
4
5 ![Elephant](elephant.png){fig-align="left"}
```

Figure Basics

- Basic

● ● ●

MARKDOWN

```
1 ![Elephant](elephant.png) 📄
```

- Size and alignment

● ● ●

MARKDOWN

```
1 ![Elephant](elephant.png){width=300} 📄
2 ![Elephant](elephant.png){width=80%}
3 ![Elephant](elephant.png){width=4in}
4
5 ![Elephant](elephant.png){fig-align="left"}
```

- Alternative text and titles

● ● ●

MARKDOWN

```
1 {fig-alt="A drawing of an elephant."} 📄
2 ![Elephant](elephant.png "Title: An elephant"){fig-alt="A drawing of an elephant."}
```

Figure Basics

- Basic

● ● ●

MARKDOWN

```
1 ![Elephant](elephant.png) 📄
```

- Size and alignment

● ● ●

MARKDOWN

```
1 ![Elephant](elephant.png){width=300} 📄
2 ![Elephant](elephant.png){width=80%}
3 ![Elephant](elephant.png){width=4in}
4
5 ![Elephant](elephant.png){fig-align="left"}
```

- Alternative text and titles

● ● ●

MARKDOWN

```
1 {fig-alt="A drawing of an elephant."} 📄
2 ![Elephant](elephant.png "Title: An elephant"){fig-alt="A drawing of an elephant."}
```

See [Figure basics](#) for more.

Advanced Formatting: Divs and Spans

Advanced Formatting: Divs and Spans

Divs for block-level content:



MARKDOWN

```
1 ::: {#ID .class attribute=value}  
2 This is a "fenced div".  
3 :::
```



Advanced Formatting: Divs and Spans

Divs for block-level content:



MARKDOWN

```
1 ::: {#ID .class attribute=value}  
2 This is a "fenced div".  
3 :::
```



Spans for inline formatting:



MARKDOWN

```
1 [This is a "span"]{#ID .class attribute=value}
```



Advanced Formatting: Divs and Spans

Divs for block-level content:



MARKDOWN

```
1 ::: {#ID .class attribute=value}  
2 This is a "fenced div".  
3 :::
```



Spans for inline formatting:



MARKDOWN

```
1 [This is a "span"]{#ID .class attribute=value}
```



See [Divs and Spans](#) for more.

- Enhanced Markdown Capabilities
- Shortcodes
- The include Shortcode
- Mermaid Diagrams
- Callout Blocks
- Code Annotations
- Cross-references: “figures”
- Cross-references: “tables”
- Cross-references: “anything”
- Citations

Quarto Markdown

Enhanced Markdown Capabilities

Quarto extends standard markdown with powerful features designed specifically for computational documents.

Shortcodes

What is a Shortcode?

Shortcodes are special **markdown directives** that **generate various types of content**.

They provide a simple, standardised way to **insert dynamic or complex content** into your Quarto documents without requiring complex markdown syntax or HTML knowledge.

Shortcodes

Built-in Shortcodes

Table 1: Shortcodes

Shortcode	Description
<u>version</u>	Print Quarto CLI version.
<u>var</u>	Print value from <code>_variables.yml</code> file.
<u>meta</u>	Print value from document metadata.
<u>env</u>	Print system environment variable.
<u>pagebreak</u>	Insert a native page-break.
<u>kbd</u>	Describe keyboard shortcuts.
<u>video</u>	Embed a video in a document.
<u>include</u>	Include contents of another qmd.
<u>embed</u>	Embed cells from a Jupyter Notebook.
<u>placeholder</u>	Add placeholder images to your document.
<u>lipsum</u>	Add placeholder text to your document.
<u>contents</u>	Rearrange content in your document.

The `include` Shortcode

The **include** Shortcode

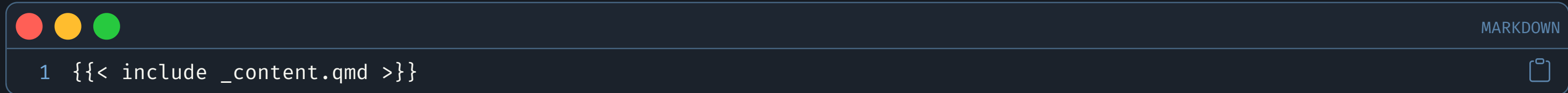
Reuse content across multiple documents effortlessly:

MARKDOWN

```
1 {{< include _content.qmd >}}
```

The **include** Shortcode

Reuse content across multiple documents effortlessly:



```
1 {{< include _content.qmd >}}
```

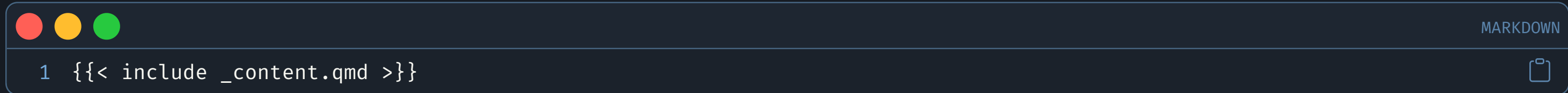


Key Benefits

- **DRY Principle:** Write once, use everywhere.
- **Consistency:** Shared content stays synchronised.
- **Maintenance:** Update in one place, changes propagate automatically.
- **Modularity:** Break complex documents into manageable pieces.

The **include** Shortcode

Reuse content across multiple documents effortlessly:



```
1 {{< include _content.qmd >}}
```

Key Benefits

- **DRY Principle:** Write once, use everywhere.
- **Consistency:** Shared content stays synchronised.
- **Maintenance:** Update in one place, changes propagate automatically.
- **Modularity:** Break complex documents into manageable pieces.

Common Use Cases

- Reusable code.
- Common configuration snippets.

The **include** Shortcode

Reuse content across multiple documents effortlessly:

● ● ●

MARKDOWN

```
1 {{< include _content.qmd >}}
```

📄

Key Benefits

- **DRY Principle:** Write once, use everywhere.
- **Consistency:** Shared content stays synchronised.
- **Maintenance:** Update in one place, changes propagate automatically.
- **Modularity:** Break complex documents into manageable pieces.

Common Use Cases

- Reusable code.
- Common configuration snippets.



Tip

Prefix your file or directory with an underscore (`_`) to hide from Quarto's rendering process.

The `include` Shortcode

Reuse content across multiple documents effortlessly:

● ● ●

MARKDOWN

```
1 {{< include _content.qmd >}}
```

📄

Key Benefits

- **DRY Principle:** Write once, use everywhere.
- **Consistency:** Shared content stays synchronised.
- **Maintenance:** Update in one place, changes propagate automatically.
- **Modularity:** Break complex documents into manageable pieces.

Common Use Cases

- Reusable code.
- Common configuration snippets.



Tip

Prefix your file or directory with an underscore (`_`) to hide from Quarto's rendering process.

! Important

The `include` shortcode is a plain copy-paste of the file content. Any YAML header in the included file will replace the current document's YAML header.

Mermaid Diagrams

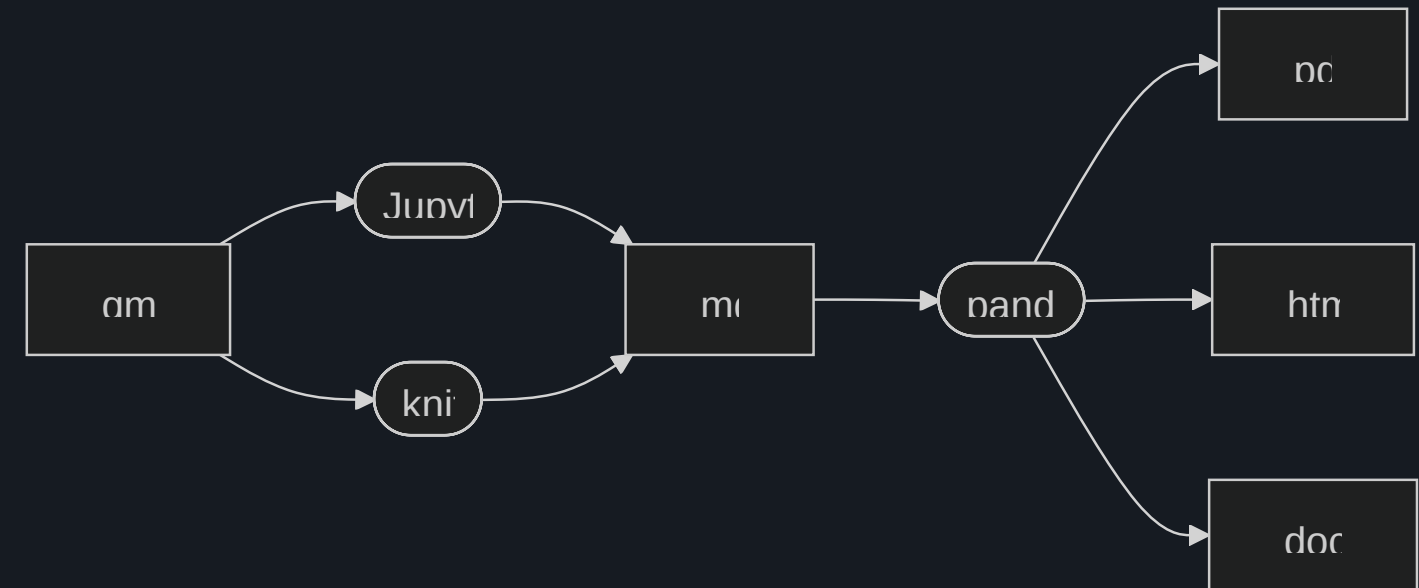
Mermaid Diagrams

MARKDOWN

```
1  ```{mermaid}
2  %%| label: quarto-mermaid-example-1
3  %%| fig-width: 5
4  flowchart LR
5      qmd → J([Jupyter])
6      qmd → K([knitr])
7      J → md
8      K → md
9      md → P([pandoc])
10     P → typst
11     P → html
12     P → docx
13  ```
```

Mermaid Diagrams

```
1  ```{mermaid}
2  %%| label: quarto-mermaid-example-1
3  %%| fig-width: 5
4  flowchart LR
5    qmd --> J([Jupyter])
6    qmd --> K([knitr])
7    J --> md
8    K --> md
9    md --> P([pandoc])
10   P --> typst
11   P --> html
12   P --> docx
13  ```
```

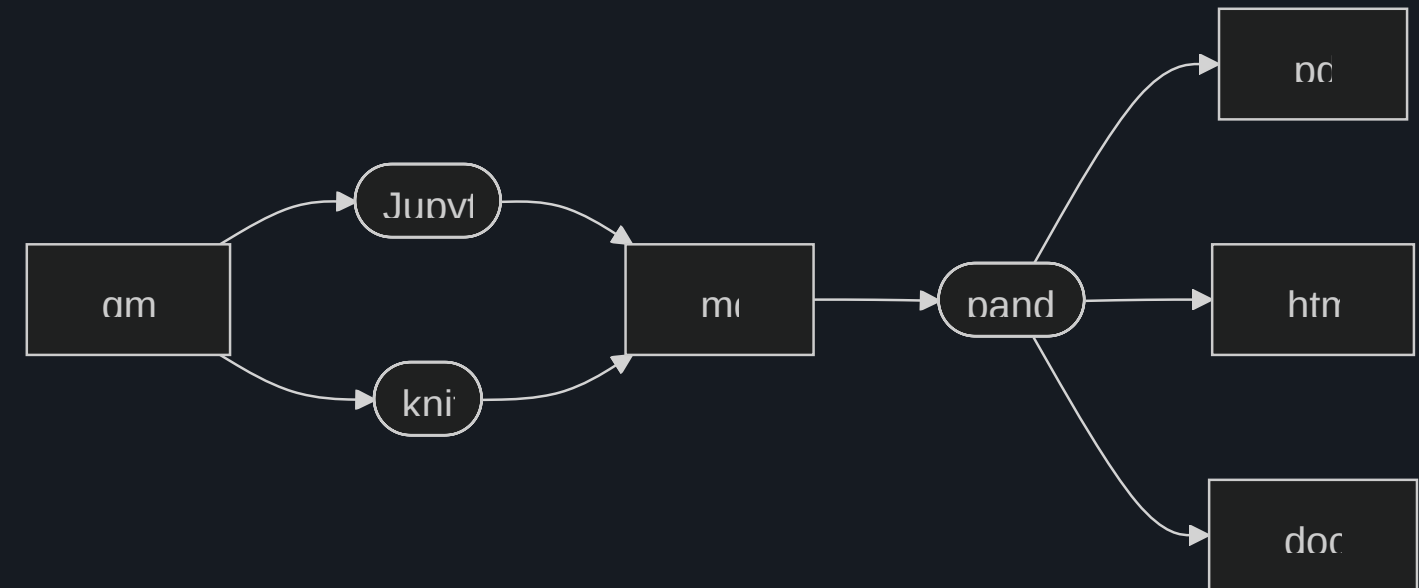


Mermaid Diagrams

```

1  ```{mermaid}
2  %%| label: quarto-mermaid-example-1
3  %%| fig-width: 5
4  flowchart LR
5    qmd --> J([Jupyter])
6    qmd --> K([knitr])
7    J --> md
8    K --> md
9    md --> P([pandoc])
10   P --> typst
11   P --> html
12   P --> docx
13  ```

```



See [Diagrams](#) for more.

Callout Blocks

Callout Blocks

Five types of callouts for better communication: `note`, `tip`, `warning`, `caution`, `important`.

Callout Blocks

Five types of callouts for better communication: `note`, `tip`, `warning`, `caution`, `important`.

```
1 ::: {.callout-note}
2 ## Note
3
4 Important information for readers.
5 :::
6
7 ::: {.callout-warning}
8 ## Warning
9
10 Something to be careful about.
11 :::
```

Callout Blocks

Five types of callouts for better communication: `note`, `tip`, `warning`, `caution`, `important`.

```
1 ::: {.callout-note}
2 ## Note
3
4 Important information for readers.
5 :::
6
7 ::: {.callout-warning}
8 ## Warning
9
10 Something to be careful about.
11 :::
```

Note

Important information for readers.

Warning

Something to be careful about.

Callout Blocks

Five types of callouts for better communication: `note`, `tip`, `warning`, `caution`, `important`.

```
MARKDOWN
1 ::: {.callout-note}
2 ## Note
3
4 Important information for readers.
5 :::
6
7 ::: {.callout-warning}
8 ## Warning
9
10 Something to be careful about.
11 :::
```

Note

Important information for readers.

Warning

Something to be careful about.

See [Callout Blocks](#) for more.

Code Annotations

Code Annotations

- Source

```
1  ```r
2  library(tidyverse)
3  library(palmerpenguins)
4  penguins ># <1>
5    mutate(  # <2>
6      bill_ratio = bill_depth_mm / bill_length_mm,  # <2>
7      bill_area  = bill_depth_mm * bill_length_mm  # <2>
8    )  # <2>
9  ```
10
11 1. Take `penguins`, and then,
12 2. add new columns for the bill ratio and bill area.
```

Code Annotations

- Source

```

1  ```r
2  library(tidyverse)
3  library(palmerpenguins)
4  penguins ▶ # <1>
5    mutate( # <2>
6      bill_ratio = bill_depth_mm / bill_length_mm, # <2>
7      bill_area  = bill_depth_mm * bill_length_mm # <2>
8    )
9  ```
10
11  1. Take `penguins`, and then,
12  2. add new columns for the bill ratio and bill area.

```

Output

```

1  library(tidyverse)
2  library(palmerpenguins)
3  penguins ▶ ①
4    mutate( ②
5      bill_ratio = bill_depth_mm / bill_length_mm,
6      bill_area  = bill_depth_mm * bill_length_mm
7    )

```

Code Annotations

- Source

```

1  ```r
2  library(tidyverse)
3  library(palmerpenguins)
4  penguins ▶ # <1>
5    mutate( # <2>
6      bill_ratio = bill_depth_mm / bill_length_mm, # <2>
7      bill_area  = bill_depth_mm * bill_length_mm # <2>
8    )
9  ```
10
11  1. Take `penguins`, and then,
12  2. add new columns for the bill ratio and bill area.

```

Output

```

1  library(tidyverse)
2  library(palmerpenguins)
3  penguins ▶
4    mutate(
5      bill_ratio = bill_depth_mm / bill_length_mm,
6      bill_area  = bill_depth_mm * bill_length_mm
7    )

```

See [Code Annotation](#) for more.

Code Annotations

- Source

```

1  ```r
2  library(tidyverse)
3  library(palmerpenguins)
4  penguins ▶                                # <1>
5    mutate(                                # <2>
6      bill_ratio = bill_depth_mm / bill_length_mm,  # <2>
7      bill_area  = bill_depth_mm * bill_length_mm  # <2>
8    )
9  ```
10
11  1. Take `penguins`, and then,
12  2. add new columns for the bill ratio and bill area.

```

- Output

```

1  library(tidyverse)
2  library(palmerpenguins)
3  penguins ▶
4    mutate(
5      bill_ratio = bill_depth_mm / bill_length_mm,
6      bill_area  = bill_depth_mm * bill_length_mm
7    )

```

See [Code Annotation](#) for more.

1 Take `penguins`, and then,
2

Code Annotations

- Source

```
1  ```r
2  library(tidyverse)
3  library(palmerpenguins)
4  penguins ># <1>
5    mutate(  # <2>
6      bill_ratio = bill_depth_mm / bill_length_mm,  # <2>
7      bill_area  = bill_depth_mm * bill_length_mm  # <2>
8    )
9  ```
10
11 1. Take `penguins`, and then,
12 2. add new columns for the bill ratio and bill area.
```

- Output

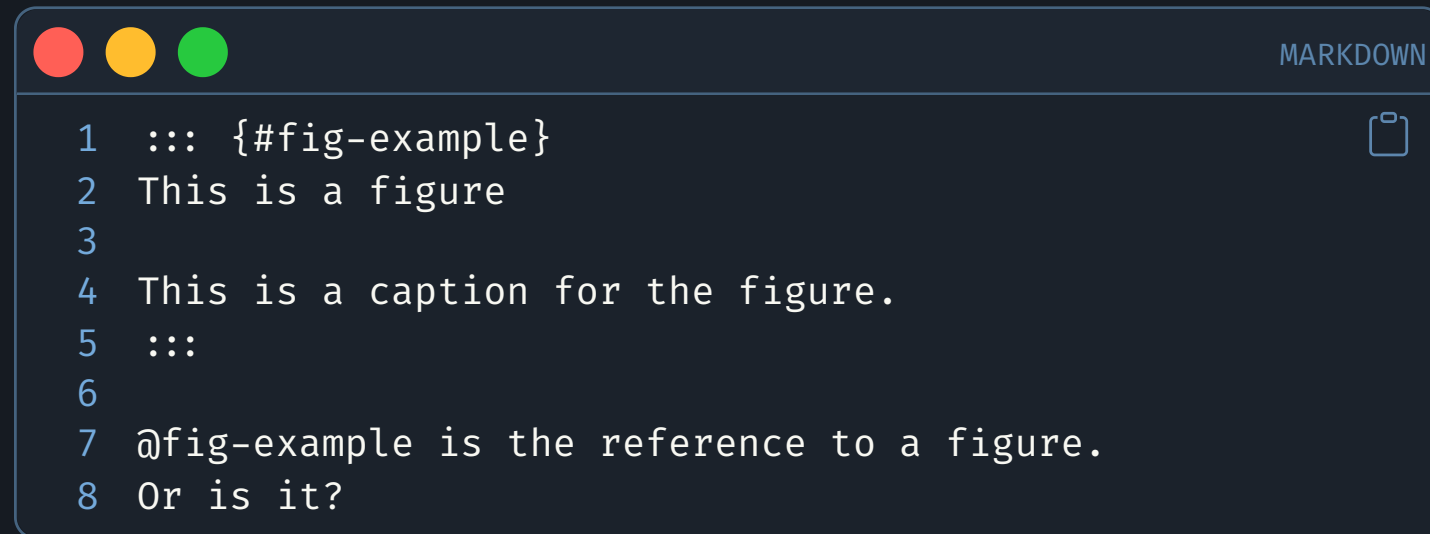
```
1  library(tidyverse)
2  library(palmerpenguins)
3  penguins >
4    mutate(
5      bill_ratio = bill_depth_mm / bill_length_mm,
6      bill_area  = bill_depth_mm * bill_length_mm
7    )
```

See [Code Annotation](#) for more.

add new columns for the bill ratio and bill area.

Cross-references: “figures”

Cross-references: “figures”



```
1 ::: {#fig-example}
2 This is a figure
3
4 This is a caption for the figure.
5 :::
6
7 @fig-example is the reference to a figure.
8 Or is it?
```

Cross-references: “figures”

```
1 ::: {#fig-example}
2 This is a figure
3
4 This is a caption for the figure.
5 :::
6
7 @fig-example is the reference to a figure.
8 Or is it?
```

This is a figure

Figure 1: This is a caption for the figure.

Figure 1 is the reference to a figure.

Cross-references: “figures”

```
1 ::: {#fig-example}
2 This is a figure
3
4 This is a caption for the figure.
5 :::
6
7 @fig-example is the reference to a figure.
8 Or is it?
```

This is a figure

Figure 1: This is a caption for the figure.

Figure 1 is the reference to a figure.

Any label/ID starting with `fig-` will be treated as a figure cross-reference.

Cross-references: “figures”

```
1 ::: {#fig-example}
2 This is a figure
3
4 This is a caption for the figure.
5 :::
6
7 @fig-example is the reference to a figure.
8 Or is it?
```

This is a figure

Figure 1: This is a caption for the figure.

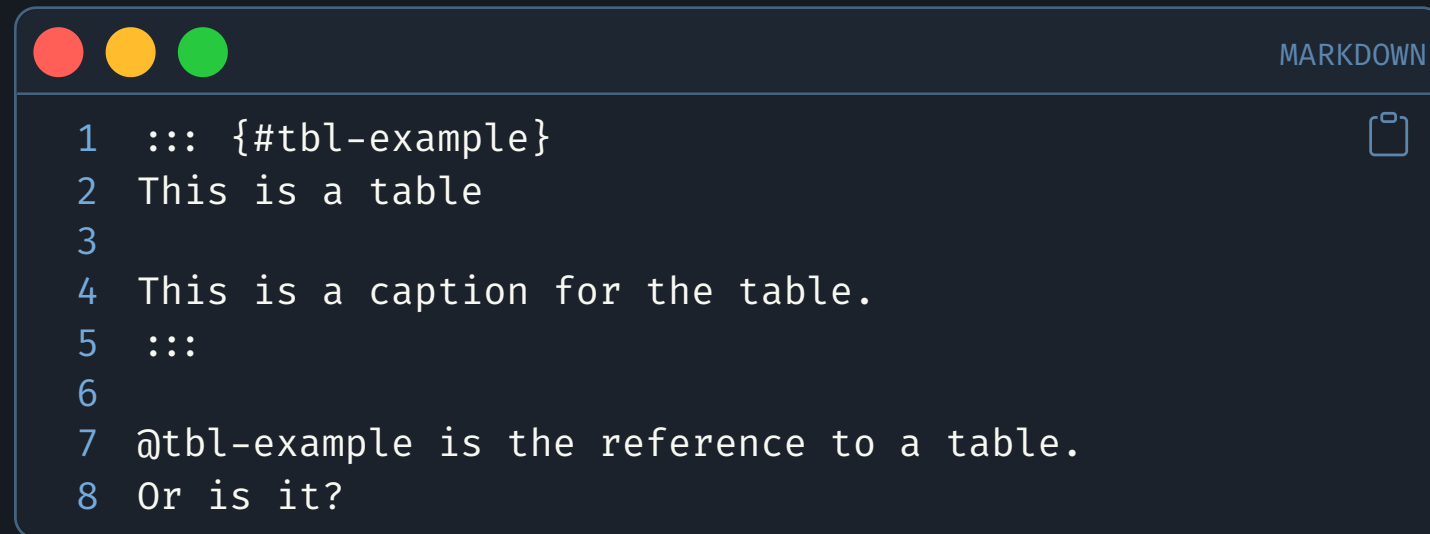
Figure 1 is the reference to a figure.

Any label/ID starting with `fig-` will be treated as a figure cross-reference.

See [Basic Cross-Reference - Figures](#) for more.

Cross-references: “tables”

Cross-references: “tables”



```
1  ::: {#tbl-example}
2  This is a table
3
4  This is a caption for the table.
5  :::
6
7  @tbl-example is the reference to a table.
8  Or is it?
```

Cross-references: “tables”

```
1 ::: {#tbl-example}
2 This is a table
3
4 This is a caption for the table.
5 :::
6
7 @tbl-example is the reference to a table.
8 Or is it?
```

Table 2: This is a caption for the table.

This is a table

Table 2 is the reference to a table.

Cross-references: “tables”

```
1 ::: {#tbl-example}
2 This is a table
3
4 This is a caption for the table.
5 :::
6
7 @tbl-example is the reference to a table.
8 Or is it?
```

Table 2: This is a caption for the table.

This is a table

Table 2 is the reference to a table.

Any label/ID starting with `tbl-` will be treated as a table cross-reference.

Cross-references: “tables”

```
1 ::: {#tbl-example}
2 This is a table
3
4 This is a caption for the table.
5 :::
6
7 @tbl-example is the reference to a table.
8 Or is it?
```

Table 2: This is a caption for the table.

This is a table

Table 2 is the reference to a table.

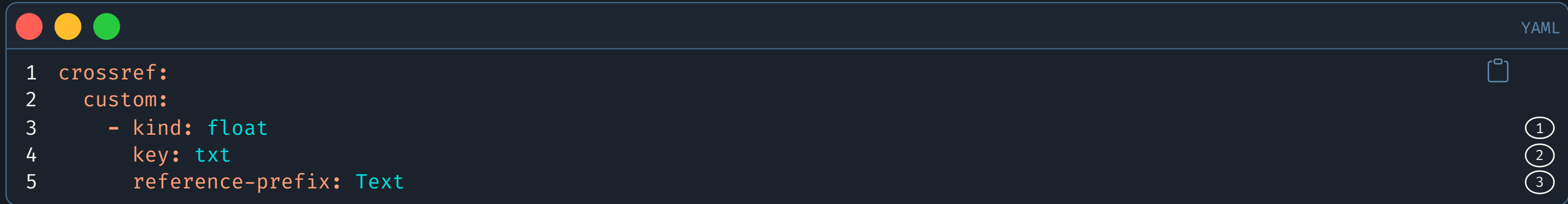
Any label/ID starting with `tbl-` will be treated as a table cross-reference.

See [Basic Cross-Reference - Tables](#) for more.

Cross-references: “anything”

Cross-references: “anything”

Define a new custom cross-reference type:



```
1 crossref:
2   custom:
3     - kind: float
4       key: txt
5       reference-prefix: Text
```

1
2
3
4
5

Cross-references: “anything”

Define a new custom cross-reference type:

YAML

```
1 crossref:
2   custom:
3     - kind: float
4       key: txt
5       reference-prefix: Text
```

1

2

3

MARKDOWN

```
1 ::: {#txt-example}
2 This is text
3
4 This is a caption for the text.
5 :::
6
7 @txt-example is the reference to a text.
```

1

Cross-references: “anything”

Define a new custom cross-reference type:

YAML

```
1 crossref:
2   custom:
3     - kind: float
4       key: txt
5       reference-prefix: Text
```

1

2

3

MARKDOWN

```
1 ::: {#txt-example}
2 This is text
3
4 This is a caption for the text.
5 :::
6
7 @txt-example is the reference to a text.
```

1

2

3

4

5

6

7

This is text

Text 1: This is a caption for the text.

Text 1 is the reference to a text.

Cross-references: “anything”

Define a new custom cross-reference type:

```

1 crossref:
2   custom:
3     - kind: float
4       key: txt
5       reference-prefix: Text
  
```

```

1 ::: {#txt-example}
2 This is text
3
4 This is a caption for the text.
5 :::
6
7 @txt-example is the reference to a text.
  
```

This is text

Text 1: This is a caption for the text.

Text 1 is the reference to a text.

See [Custom Float Cross-Reference Types](#) for more.

Cross-references: “anything”

Define a new custom cross-reference type:

```

1 crossref:
2   custom:
3     - kind: float
4       key: txt
5       reference-prefix: Text

```

Cross-referenceable elements with captions are float.

1
2
3

```

1 ::: {#txt-example}
2 This is text
3
4 This is a caption for the text.
5 :::
6
7 @txt-example is the reference to a text.

```

This is text

Text 1: This is a caption for the text.

Text 1 is the reference to a text.

See [Custom Float Cross-Reference Types](#) for more.

Cross-references: “anything”

Define a new custom cross-reference type:

YAML

```

1 crossref:
2   custom:
3     - kind: float
4       key: txt
5       reference-prefix: Text

```

2

3

The identifier prefix used in labels (e.g., `@txt-example`).

MARKDOWN

```

1 ::: {#txt-example}
2 This is text
3
4 This is a caption for the text.
5 :::
6
7 @txt-example is the reference to a text.

```

This is text

Text 1: This is a caption for the text.

Text 1 is the reference to a text.

See [Custom Float Cross-Reference Types](#) for more.

Cross-references: “anything”

Define a new custom cross-reference type:

```

1 crossref:
2   custom:
3     - kind: float
4       key: txt
5       reference-prefix: Text
  
```

YAML

The text prefix shown in output (e.g., “Text 1”).

```

1 ::: {#txt-example}
2 This is text
3
4 This is a caption for the text.
5 :::
6
7 @txt-example is the reference to a text.
  
```

MARKDOWN

This is text

Text 1: This is a caption for the text.

Text 1 is the reference to a text.

See [Custom Float Cross-Reference Types](#) for more.

Citations

Citations

Setup Required

- Bibliography file (`.bib` , `.bibtex` , etc.).
- Optional: CSL file for custom styling.

Citations

Setup Required

- Bibliography file (`.bib` , `.bibtex` , etc.).
- Optional: CSL file for custom styling.

YAML Configuration

YAML

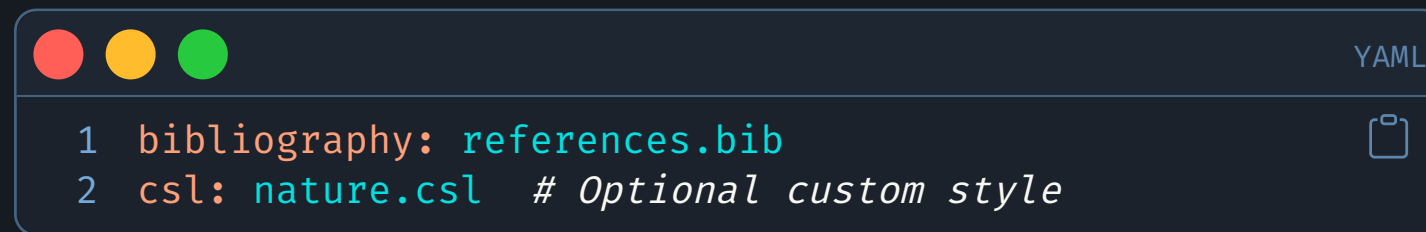
```
1 bibliography: references.bib
2 csl: nature.csl # Optional custom style
```

Citations

Setup Required

- Bibliography file (`.bib`, `.bibtex`, etc.).
- Optional: CSL file for custom styling.

YAML Configuration



```
1 bibliography: references.bib
2 csl: nature.csl # Optional custom style
```

Bibliography Generation

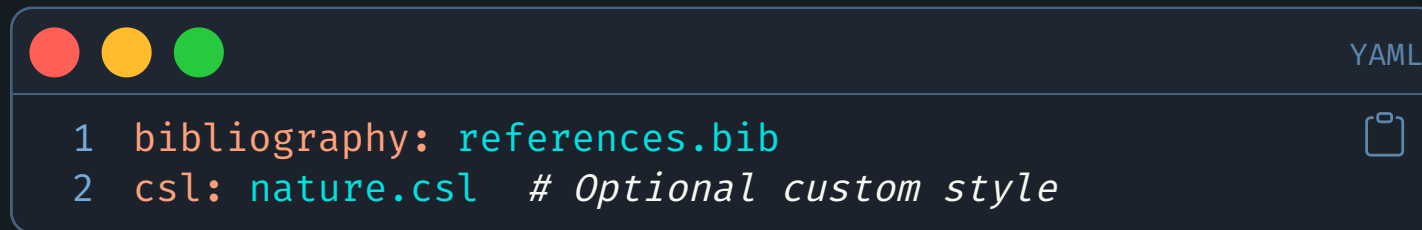
- Automatic placement at document end.
- Use `## References {#refs}` for custom placement.
- `nocite: "@*"` includes all references.

Citations

Setup Required

- Bibliography file (`.bib`, `.bibtex`, etc.).
- Optional: CSL file for custom styling.

YAML Configuration



```
1 bibliography: references.bib
2 csl: nature.csl # Optional custom style
```

Bibliography Generation

- Automatic placement at document end.
- Use `## References {#refs}` for custom placement.
- `nocite: "@*"` includes all references.

Citation Syntax

- `[@author2023]` → (Author 2023)
- `[@author2023, p. 15]` → (Author 2023, 15)
- `[@smith2020; @jones2021]` → (Smith 2020; Jones 2021)
- `@author2023 says ...` → Author (2023) says...
- `[-@author2023]` → (2023) - suppress author

- Overview
- Document Header
- Multi-Format Output
- Extensions

YAML Configuration

Overview

Overview

The YAML header serves as the configuration hub for your documents, controlling output formats, execution behaviour, and visual presentation.

Overview

The YAML header serves as the configuration hub for your documents, controlling output formats, execution behaviour, and visual presentation.

Metadata Inheritance:

- **Project level:** `_quarto.yml` (*lowest priority*).
- **Directory level:** `_metadata.yml` (*medium priority*).
- **Document level:** YAML header (*highest priority*).

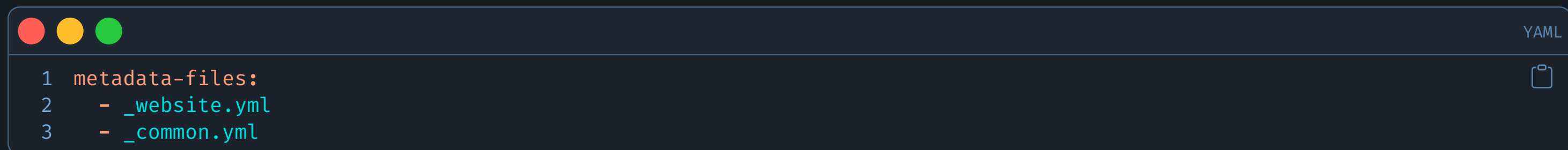
Overview

The YAML header serves as the configuration hub for your documents, controlling output formats, execution behaviour, and visual presentation.

Metadata Inheritance:

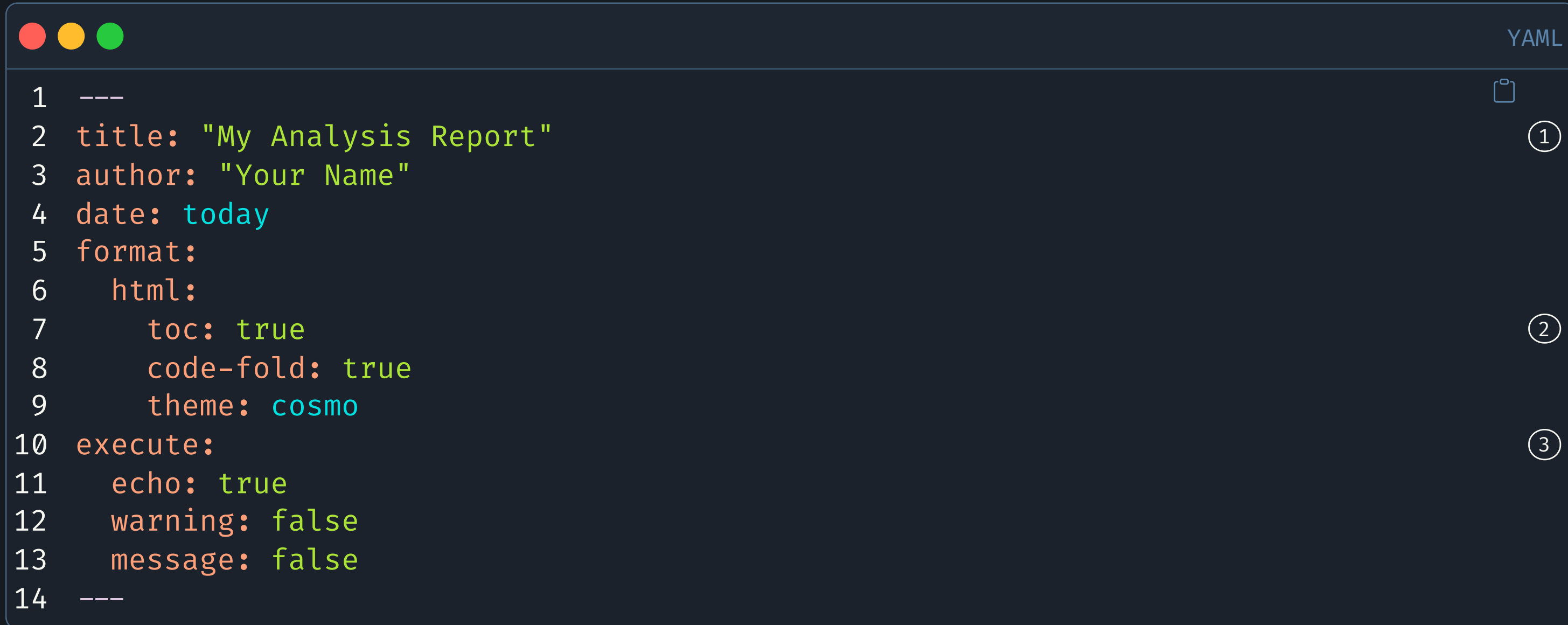
- **Project level:** `_quarto.yml` (*lowest priority*).
- **Directory level:** `_metadata.yml` (*medium priority*).
- **Document level:** YAML header (*highest priority*).

Multiple Metadata Files:



```
1 metadata-files:
2   - _website.yml
3   - _common.yml
```

Document Header



```
1 ---
2 title: "My Analysis Report"
3 author: "Your Name"
4 date: today
5 format:
6   html:
7     toc: true
8     code-fold: true
9     theme: cosmo
10 execute:
11   echo: true
12   warning: false
13   message: false
14 ---
```

Document Header



```
1 ---
2 title: "My Analysis Report"
3 author: "Your Name"
4 date: today
5 format:
6   html:
7     toc: true
8     code-fold: true
9     theme: cosmo
10 execute:
11   echo: true
12   warning: false
13   message: false
14 ---
```

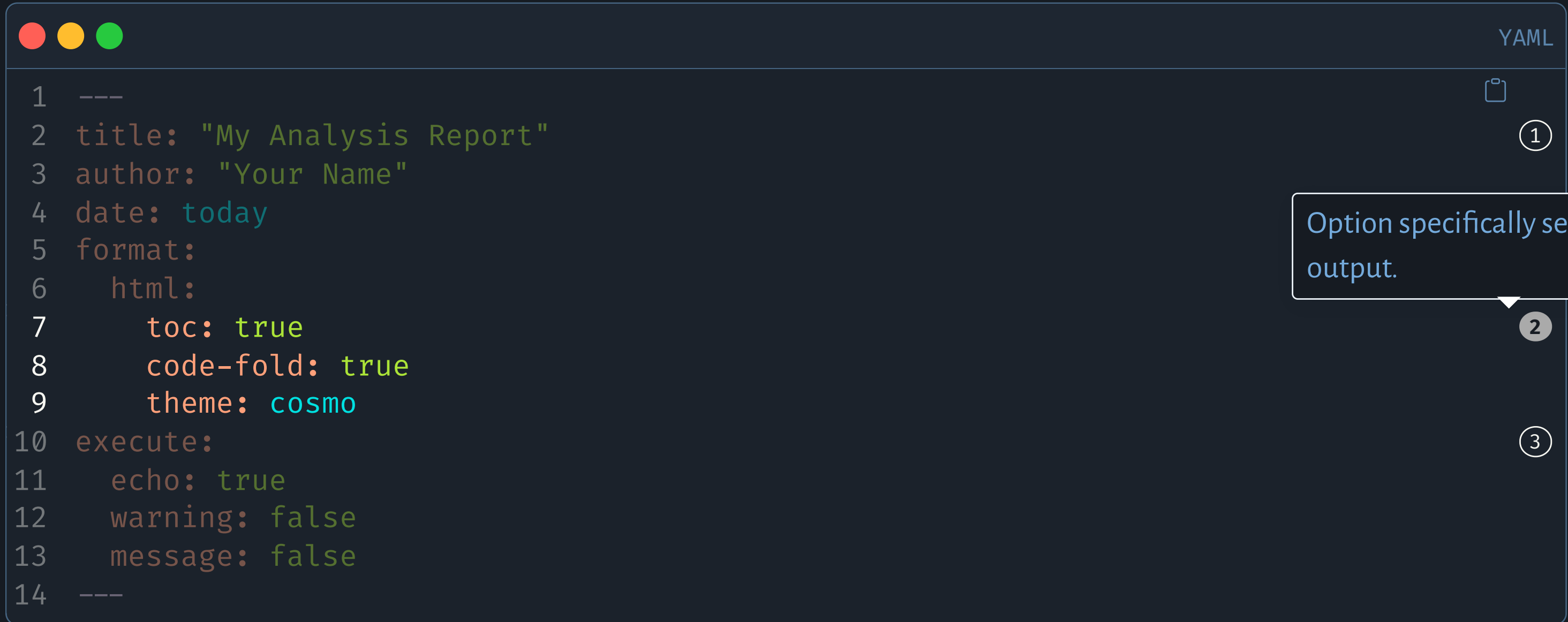
Common metadata fields. Option shared by all formats.

1

2

3

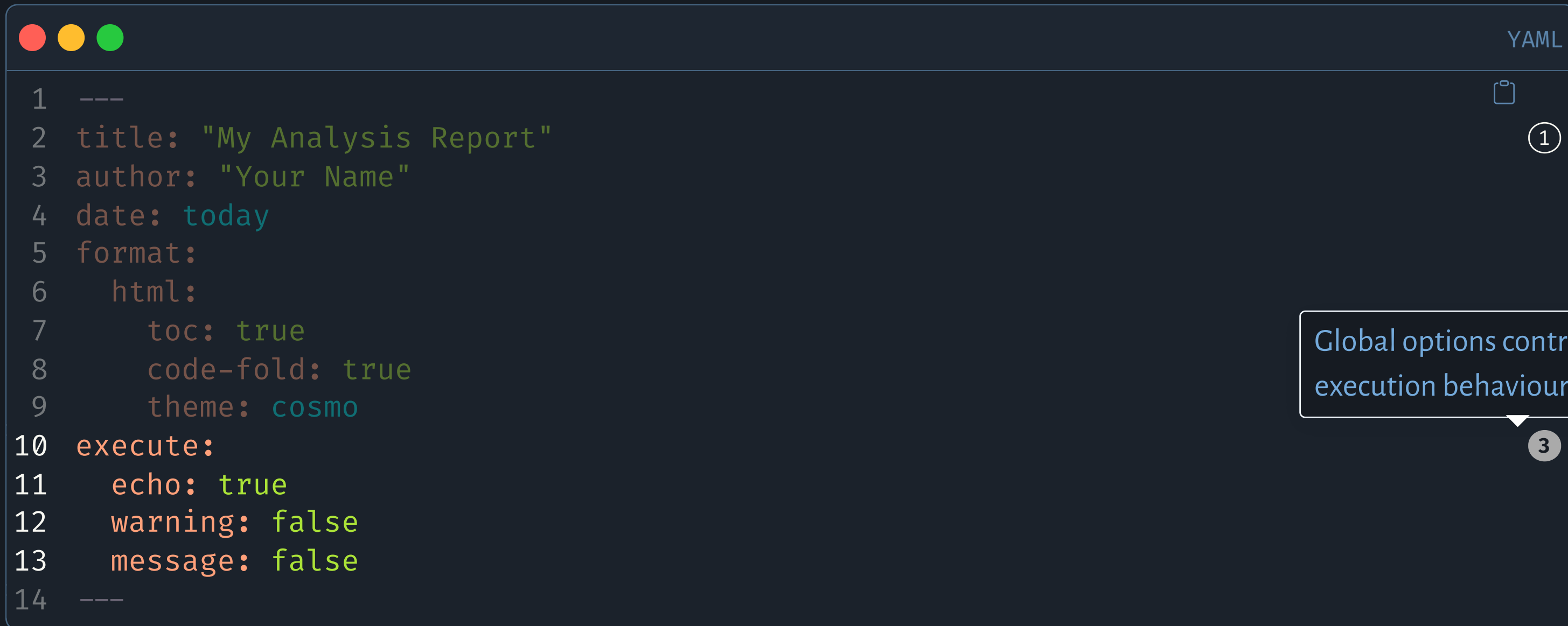
Document Header



```
1 ---
2 title: "My Analysis Report"
3 author: "Your Name"
4 date: today
5 format:
6   html:
7     toc: true
8     code-fold: true
9     theme: cosmo
10 execute:
11   echo: true
12   warning: false
13   message: false
14 ---
```

Option specifically set for HTML output.

Document Header



```
1 ---
2 title: "My Analysis Report"
3 author: "Your Name"
4 date: today
5 format:
6   html:
7     toc: true
8     code-fold: true
9     theme: cosmo
10 execute:
11   echo: true
12   warning: false
13   message: false
14 ---
```

Global options controlling code execution behaviour.

Document Header

● ● ●

YAML

```
1 ---
2 title: "My Analysis Report"
3 author: "Your Name"
4 date: today
5 format:
6   html:
7     toc: true
8     code-fold: true
9     theme: cosmo
10 execute:
11   echo: true
12   warning: false
13   message: false
14 ---
```

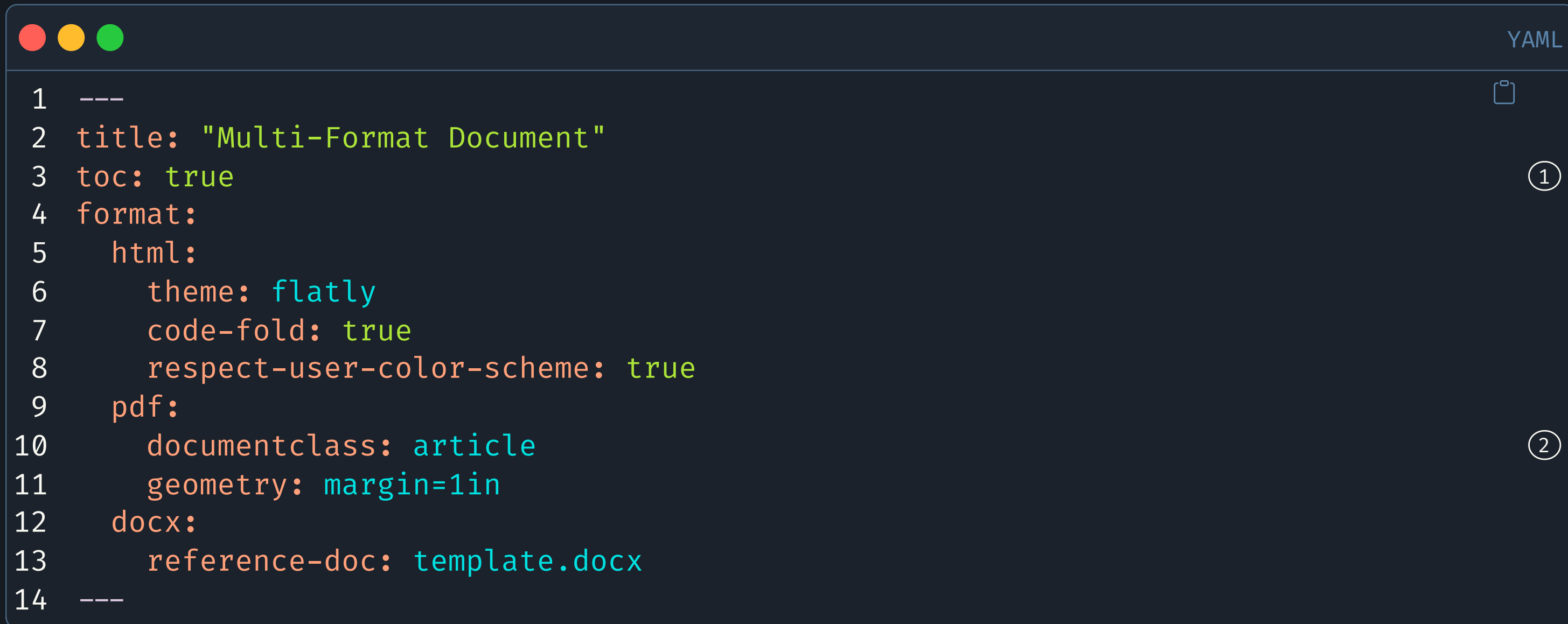
①

②

③

Multi-Format Output

Configure multiple output formats simultaneously and share options:



```
1 ---
2 title: "Multi-Format Document"
3 toc: true
4 format:
5   html:
6     theme: flatly
7     code-fold: true
8     respect-user-color-scheme: true
9   pdf:
10    documentclass: article
11    geometry: margin=1in
12  docx:
13    reference-doc: template.docx
14 ---
```

Multi-Format Output

Configure multiple output formats simultaneously and share options:

```
1 ---
2 title: "Multi-Format Document"
3 toc: true
4 format:
5   html:
6     theme: flatly
7     code-fold: true
8     respect-user-color-scheme: true
9   pdf:
10     documentclass: article
11     geometry: margin=1in
12   docx:
13     reference-doc: template.docx
14 ---
```

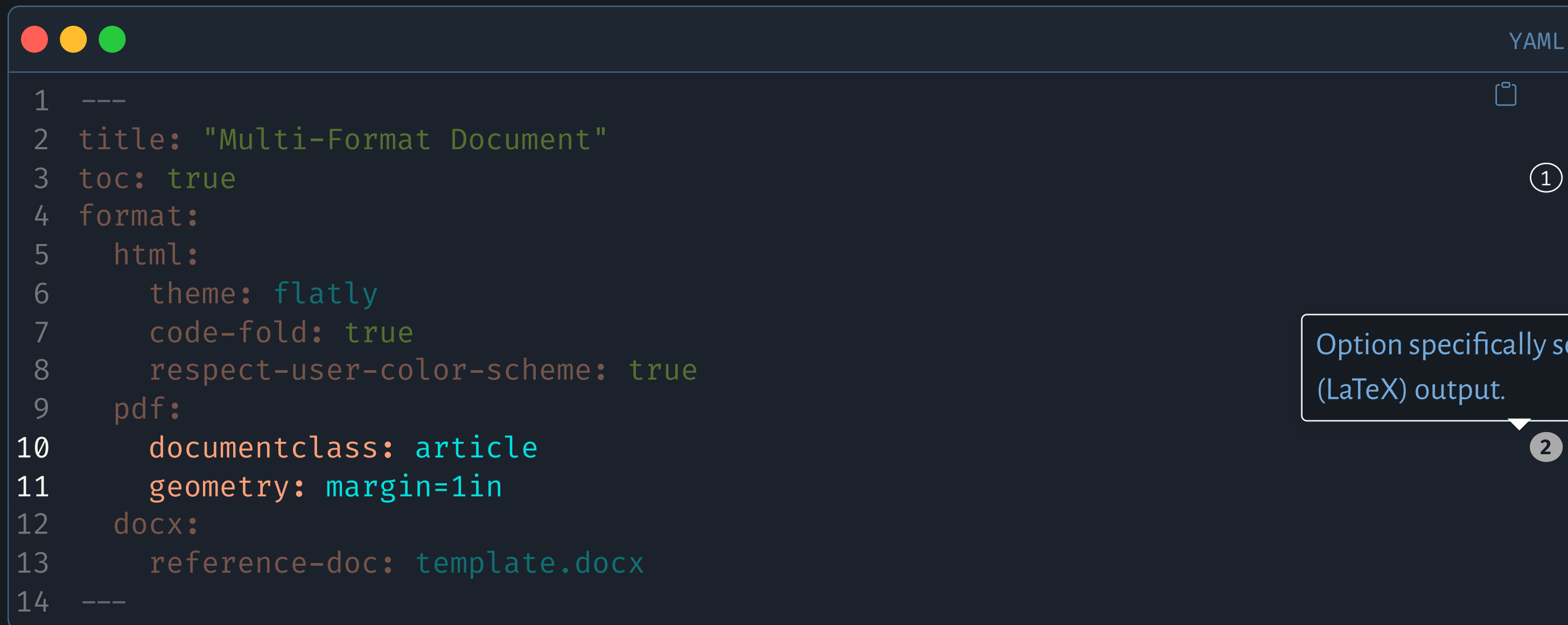
Option shared by all formats.

1

2

Multi-Format Output

Configure multiple output formats simultaneously and share options:

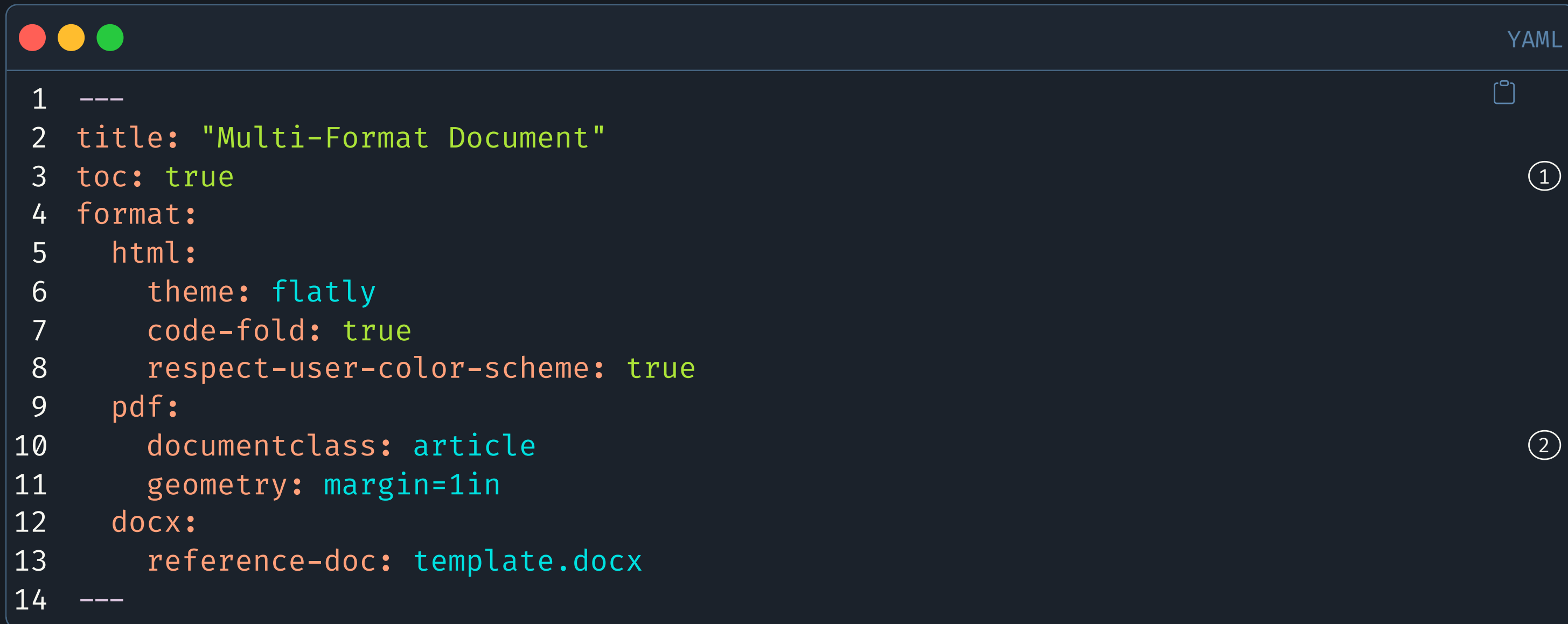


```
1 ---
2 title: "Multi-Format Document"
3 toc: true
4 format:
5   html:
6     theme: flatly
7     code-fold: true
8     respect-user-color-scheme: true
9   pdf:
10    documentclass: article
11    geometry: margin=1in
12  docx:
13    reference-doc: template.docx
14 ---
```

Option specifically set for PDF
(LaTeX) output.

Multi-Format Output

Configure multiple output formats simultaneously and share options:



```
1 ---
2 title: "Multi-Format Document"
3 toc: true
4 format:
5   html:
6     theme: flatly
7     code-fold: true
8     respect-user-color-scheme: true
9   pdf:
10    documentclass: article
11    geometry: margin=1in
12   docx:
13     reference-doc: template.docx
14 ---
```

Extensions

Extensions

💡 Installing the `highlight-text` extension



Bash

```
1 quarto add mcanouil/quarto-highlight-text@2.0.1
```



YAML

```
1 filters:  
2   - highlight-text
```



Extensions

💡 Installing the `highlight-text` extension



Bash

```
1 quarto add mcanouil/quarto-highlight-text@2.0.1
```



YAML

```
1 filters:
2   - highlight-text
```



MARKDOWN

```
1 [White text, Blue background]{
2   colour="#FFFFFF" bg-colour="#0000FF"
3 }
```



Extensions

💡 Installing the `highlight-text` extension

```
1 quarto add mcanouil/quarto-highlight-text@2.0.1
```

```
1 filters:
2   - highlight-text
```

```
1 [White text, Blue background]{
2   colour="#FFFFFF" bg-colour="#0000FF"
3 }
```

White text, Blue background

Extensions

Installing the `highlight-text` extension

```
1 quarto add mcanouil/quarto-highlight-text@2.0.1
```

```
1 filters:
2   - highlight-text
```

```
1 [White text, Blue background]{
2   colour="#FFFFFF" bg-colour="#0000FF"
3 }
```

White text, Blue background

Discover more extensions at:

- <https://quarto.org/docs/extensions/>.
- <http://m.canouil.dev/quarto-extensions/>.

- Exercise Overview
- Part 1: Document Enhancement
- Part 2: Advanced Formatting Features
- Part 3: Skills Showcase Section
- Part 4: Multi-Format Testing
- Success Criteria

Hands-On Exercise: Building Your Quarto Portfolio Document

Exercise Overview

Objective: Enhance your Session 1 project with advanced Quarto markdown features, demonstrating authoring essentials and creating a comprehensive personal portfolio document.

Example Code:  [Exercises](#)



BASH

```
1 tar -xzf "02-exercises.tar.gz" -C "02-authoring-essentials"
```



Part 1: Document Enhancement

Building on your Session 1 project, enhance your document with these new authoring features:

Part 1: Document Enhancement

Building on your Session 1 project, enhance your document with these new authoring features:

1. **Create a new document** with multi-format support:

▶ Code

Part 1: Document Enhancement

Building on your Session 1 project, enhance your document with these new authoring features:

2. **Add a numbered table of contents** structured by using:

▶ Code

Part 2: Advanced Formatting Features

Add these enhanced markdown elements to your “Learning Progress” section:

Part 2: Advanced Formatting Features

Add these enhanced markdown elements to your “Learning Progress” section:

1. Create a skills tracking table:

▶ Code

Part 2: Advanced Formatting Features

Add these enhanced markdown elements to your “Learning Progress” section:

2. **Add callout blocks** for key insights:

▶ Code

Part 3: Skills Showcase Section

Create a “Skills Showcase” section demonstrating your new capabilities:

Part 3: Skills Showcase Section

Create a “Skills Showcase” section demonstrating your new capabilities:

- **Mathematical expressions:**
 - ▶ Code
- **Code blocks with annotations:**
 - ▶ Code

Part 3: Skills Showcase Section

Create a “Skills Showcase” section demonstrating your new capabilities:

- **Mathematical expressions:**
 - ▶ Code
- **Code blocks with annotations:**
 - ▶ Code
- **Cross-referenced figure:**
 - ▶ Code
- **Footnotes for additional context:**
 - ▶ Code

Part 4: Multi-Format Testing

Test your enhanced document:

- Render to HTML

```
 BASH  
1 quarto render your-document.qmd --to html 
```

- Try PDF output

```
 BASH  
1 quarto render your-document.qmd --to typst 
```

- Compare the outputs

Success Criteria

You've successfully completed the exercise if you can:

- Customise your existing project with YAML configuration.
- Implement cross-references with tables and figures.
- Use callout blocks effectively for highlighting information.
- Create annotated code blocks.
- Integrate mathematical expressions appropriately.
- Successfully render to multiple formats.



Mickaël CANOUIL, *Ph.D.*

Independent Contractor

pro@mickael.canouil.dev



Website

mickael.canouil.fr



GitHub

[@mcanouil](https://github.com/mcanouil)



LinkedIn

[MickaelCanouil](https://www.linkedin.com/company/MickaelCanouil)



Bluesky

[@mickael.canouil.fr](https://bsky.app/profile/@mickael.canouil.fr)



Mastodon

[@MickaelCanouil](https://mastodon.social/@MickaelCanouil)