

ScrivQ

A Scrivener Template for Quarto Publishing

Bernardo C. D. A. Vasconcelos

Copyright © 2024 Bernardo César Diniz Athayde Vasconcelos

POLYTROPIKAL Publishing.

With thanks to the [Typst Orange Template](#).

Licensed under the Apache 2.0 License (the “License”). You may not use this file except in compliance with the License. You may obtain a copy of the License at <https://www.apache.org/licenses/LICENSE-2.0>. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

First printing, 2024-11

Contents

1	Instalation	5
2	ScrivQ	7
3	Compiler Script	8
4	Citations	9
4.1	Basic citations	9
4.2	Citation of specific fields	10
4.3	Multipart Bibliography	11
4.3.1	How to manually create a multipart bibliography	12
4.3.2	How to automatically create a multipart bibliography	13
4.4	Backlinks	13
5	Quarto	15
5.1	Scrivener Project Templates	15
5.2	Cross-referencing	15
5.3	Amsthm	16
5.4	Callouts	17
5.5	Diagrams	19
5.6	Equations	22
5.7	Figures	23
5.8	Listings	24
5.9	Tables	25
5.10	Sections	26
5.10.1	Section (Unnumbered)	26
5.10.1	Section	27
5.10.2	Break + Section	27
5.10.3	Heading	27
5.10.4	Break + Heading	27
6	Templates and partials	28
7	Resources	29
8	Final word	30
8	Bibliography	31
8	Primary Sources	31
8	Secondary Sources	31
8	Workflow	31

8	Songs	31
8	Paintings	31

1 Instalation

To use ScrivQ24, you need the [Quarto](#) open-source scientific and technical publishing system.

💡 Quickstart

- On macOS¹, use [Homebrew](#) to `brew install quarto && brew install chromium`.
- On Windows, use [Chocolatey](#) to `choco install quarto`.

After that, install TinyTex with `quarto install tool tinytex`.

💡 New to Quarto?

- Visit the [get started](#) guide to download.
- Check the [tutorial](#) to learn how to create, build, and preview documents.
- Install the Quarto extension for [VS Code](#) and [R Studio](#).
- Find out more at [Awesome Quarto](#), [Quarto Extensions](#), [Quarto on Github](#), [RStudio Community](#), [Stack Overflow](#), [Twitter](#).
- See also the [Pandoc User's Guide](#), [Pandoc-Discuss](#), and [Pandoc at Stack Overflow](#).

❗ Installation issues

Run a `quarto check` on your installation.

Some Python and R computations will require additional packages (cf. Warning 1).

¹If you're on a Mac, install Homebrew by pasting the following on the Terminal: `/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"`.

⚠ Warning 1.1: Dependencies for computations

To run R computations, install [R](#), [R Studio](#), and (*at least*) the following packages:

Listing 1.1: Additional R packages.

```
py_install(packages = "matplotlib")
install.packages("reticulate")
install.packages("markdown")
install.packages("tidyverse")
install.packages("kableExtra")
install.packages("downlit")
install.packages("xml2")
```

2 ScrivQ

The ScrivQ template is designed to compile [Quarto Books](#) (PDF², DOCX, and HTML) with zero configuration and no dependencies on external files. The bibliography, the template files, extensions, and filters are all included in the project and will be created by the compiler script.

💡 When Quarto is installed...

Hit `Cmd + Opt + E` to publish your Quarto Book right away.

ℹ️ ScrivQ

- [Download](#) the [latest release](#).
- [Sponsor](#) [this project](#).
- [Take part](#) in the discussion at the [Scrivener forum](#).

ℹ️ Thank you

[@iandol](#) for the exceptional [Scrivomatic](#) from which ScrivQ descended.

Check his [writing in Scrivener tutorial](#), if you haven't done so yet.

²Please note that [tinytex](#) is also required for LaTeX to PDF output.

3 Compiler Script

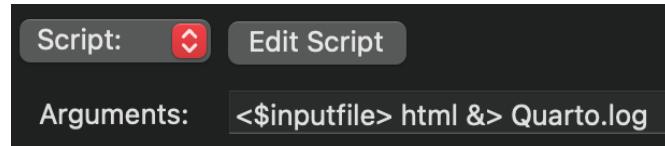


Figure 3.1: There is a ruby script to reformat the text and create the dependencies. It allows you to pass a file name and a format, so you can edit the post-processing panel to change the output format (e.g. change html to pdf / docx / epub etc.). It is embedded in this project's Compiler format.

The script's output is redirected `&>` to a `Quarto.log` file that will be automatically opened when an error happens and no output is produced. If everything runs smoothly, however, you should see only the output file open.

This compiler script descends from [Scrivomatic](#), but with several changes to allow splitting the final document into sub-files. This task entails some challenges as footnotes and image references must be moved around before the file is split. Check the compiler script to see how this got solved and please report any errors.

⚠ Warning

Ruby must be installed on Windows machines.

4 Citations

“I propose a toast, to my self-control. You see it helpless, crawling on the floor.” Morphine, *Cure For Pain* (1993)

In ScrivQ, we use [Citeproc](#) to handle the bibliography and provide consistent output across different formats. Given Citeproc’s lack of features beloved by [BibTeX](#) users, we included [Cite Tools] to enable multipart bibliographies, provide backlinks (see `backref` from [HyperRef](#)), and access bibliographic data from sources (*author, editor, translator, date, edition, number*, and so on).

Official documentation

The official documentation on citations can be found at [Pandoc](#) and [Quarto](#).

Cite Tools needed for ScrivQ

Deleting [Cite Tools](#) from ScrivQ will cause the compilation to fail.

Using [Cite Tools](#) in other projects

Install it with `quarto install extension bcdavasconcelos/citetools`.

4.1 Basic citations

1. The citation syntax is straightforward: `@Citekey` for **Author (Date)** (an *in-text* citation); `[@Citekey]` for **Author, Date**; and `[-@Citekey]` for **Date**.
2. The citation key is optionally followed by a locator, which can be a page number, a line number, a chapter number, or a section number, preceded by a comma, *e.g.* `[@Citekey, p.10]`.
3. Multiple citations can be grouped in brackets separated by semicolons `[@CitekeyA; @CitekeyB]`. The CSL style used by Citeproc will determine the presence (or absence) of parenthesis around the rendered citation. (We favor a style that doesn’t use parenthesis, but that is up to you.)

Character Style	Markdown Source	Rendered output ³
Cite*	[-@Long2004]	2004
Cite*	[-@Long2004, p.15]	2004, p. 15
Cite	[@Long2004]	LONG, 2004
Cite	[@Long2004, p.15]	LONG, 2004, p. 15

Table 4.1: Citation syntax in Quarto and Pandoc.

Here is a short demonstration of the basic citation feature. We suggest grouping the citations using parenthesis and using character styles to apply the correct markup.

i (Date, locator)

Long thinks [...] on the deliberations of the prudent person ([-@Long2004, p.17]).

Long thinks [...] on the deliberations of the prudent person ([2004, p. 17](#)).

i (Author, Date, locator)

...on the deliberations of the prudent person ([@Long2004, p.17]).

...on the deliberations of the prudent person ([LONG, 2004, p. 17](#)).

i (Author, Date, locator; Author, Date, locator)

...on the deliberations of the prudent person ([@Long2004, p.17]; [@hoffman2014, p.15]).

...on the deliberations of the prudent person ([LONG, 2004, p. 17](#); [HOFFMAN & PRAKASH, 2014, p. 15](#)).

4.2 Citation of specific fields

To inject the correct markup – `[@Citekey]{.csl_field}` – and allow us to cite different fields from our bibliographic entry, we rely on **Character Styles** (e.g. *Cite Author*, *Cite Editor*, *Cite Issued*, and so on). We have stuck to the term fields, but the official terminology is [CSL Variables](#), [BibTeX Fields](#), and [RIS Tags](#).

³The rendered citation will appear only in the output document; but not in the Scrivener project.

Character Style	Markdown Source	Rendered Output
Cite Author	[@DA]{.author}	Aristotelis
Cite Editor	[@DA]{.editor}	Bekker
Cite Translator	[@DA]{.translator}	Τατάκης
Cite Issued	[@DA]{.issued}	1834
Cite Title	[@DA]{.title}	<i>De Anima</i>
Cite Title-short	[@DA]{.title-short}	<i>De An.</i>
Cite Original-title	[@DA]{.original-title}	περὶ ψυχῆς
Cite Publisher	[@DA]{.publisher}	Reimer
Cite Publisher-Place	[@DA]{.publisher-place}	Berlin

Table 4.2: All ready-made **Character Styles** for the Cite Field lua filter.

Example

Aristotle's [@DA]{.original-title} ([@DA]{.title}) was first edited by [@DA]{.editor} in [@DA]{.issued}. In [@DABiehl]{.issued}, there was another edition by [@DABiehl]{.editor} (which was reprinted in [@DATheiler]{.translator}'s [@DATheiler]{.issued} translation).

Aristotle's περὶ ψυχῆς (*De Anima*) was first edited by Bekker in 1834. In 1896, there was another edition by Biehl (which was reprinted in Theiler's 1995 translation).

4.3 Multipart Bibliography

Where do I plug my bibliography?

There is no need to keep separate bibliography files in the system. Copy and paste the data from bibliography managers straight into Scrivener following the instructions below and you will be set!

Bibliography Formats

Pandoc and **Quarto** use a **CSL** (*Citation Style Language*) engine and prefer **CSL-YAML** and **CSL-JSON**, performing up to 10x faster, over **BibTeX** and **RIS**, which need to be converted before they can be understood.

💡 What if I don't have a bibliography ready?

- Check [Zotero⁴](#) and [JabRef](#) on all platforms.
- On macOS, check [Bookends](#) and [Bibdesk](#);
- See also, [here](#).

4.3.a How to manually create a multipart bibliography

1. Using the **Section Type** File, we create a representation of our bibliography file to add the data (e.g. [Primary Sources](#) and [Secondary Sources](#)).
2. On the Metadata panel we set the relative path (ID-Prefix + ID) and the extension (Extension) of the actual bibliography file that will be created upon **Compile**.

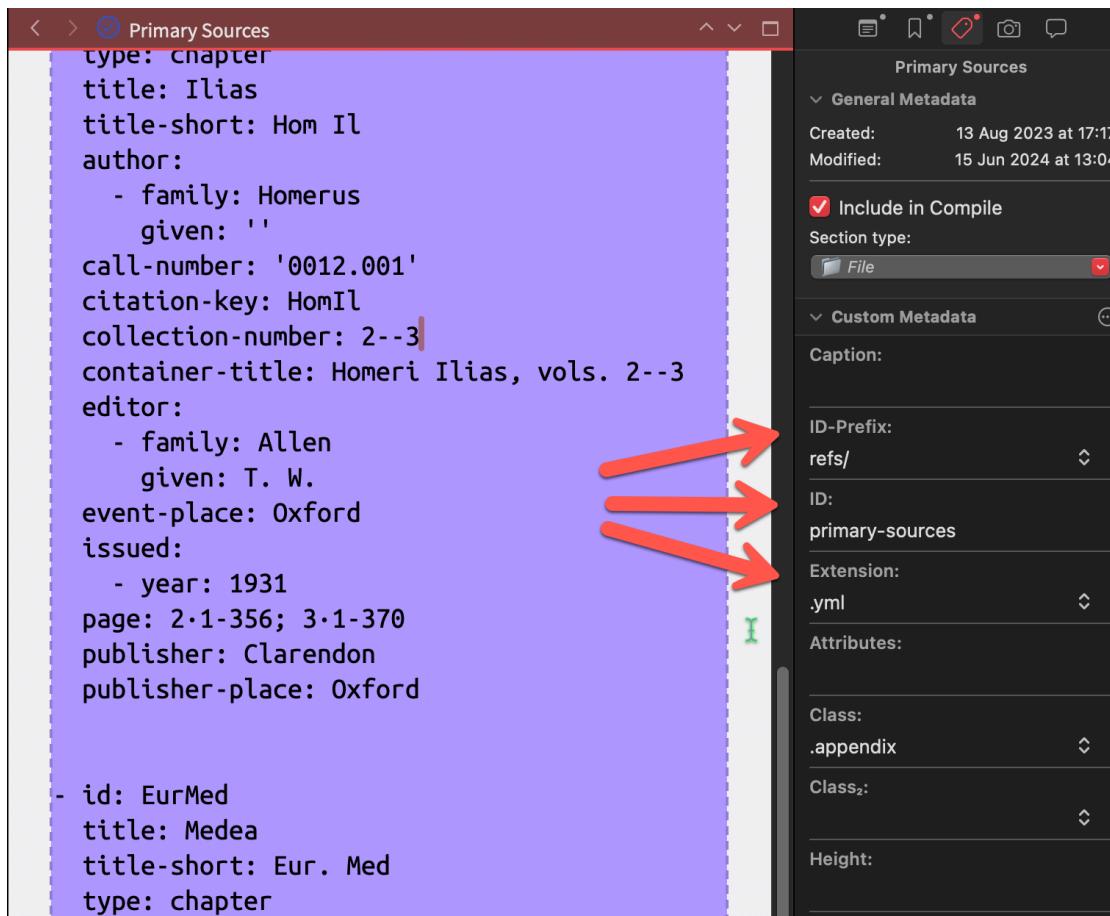


Figure 4.2: The Metadata panel

⁴Zotero even offers an API to download shared libraries by merely accessing a link, such as <https://api.zotero.org/groups/LibraryID/items?format=bibtex&limit=999> where LibraryID corresponds to the library's 7-digit code (visible in the middle of the library URL).

3. We need to tell Quarto about the bibliography file by adding it to the `_quarto` configuration file (there is a bibliography section), then we can print the formatted bibliography using the ID (e.g. “primary-sources”) with the **Paragraph Style Div Bibliography**.

ARISTOTELIS. “De Anima”. Em: BEKKER, I. (Ed.). *Aristotelis Opera*. Trad.: B. Τατάκης. Berlin: Reimer, 1834.

4.3.b How to automatically create a multipart bibliography

We can use the **Section Type** Bibliography to automate steps 3 and 4. This is very convenient for books that need the bibliography to print only once at the very end.

1. Using the **Section Type** File, we create a representation of our bibliography file to add the data (e.g. [Primary Sources](#) and [Secondary Sources](#)).
2. On the Metadata panel we set the relative path (ID-Prefix + ID) and the extension (Extension) of the actual bibliography file that will be created upon **Compile**.
3. The metadata with the file path will be automatically added and the formatted bibliography will be printed in the same section as the data, with the same section title.

4.4 Backlinks

In Citeproc, `link-citations` control whether citations in the body of the text should be clickable links to the reference in the bibliography. **Cite Tools** takes it further and adds a backlink to each citation an entry has received in the document in a crescent ordinal fashion⁵. This allows the reader to easily arrive at sections of the text where the same reference was discussed and quickly see how many times each reference was used with the array of backlinks.

⁵The reader will see the page number instead of a crescent ordinal number in some output formats, such as PDF.

manuel Bekker. 4 vols. Berlim: Reimer. [1, 2]
i, edited by Immanuel Bekker, 402a01–435b25. Berlim:
iehl. Leipzig: Teubner. [1, 2]
ed by Aurél Förster. Budapest: Hungarian Academy of
avid Ross. Oxford: Clarendon. [1, 2]

Figure 4.3: The **Citation Backlinks** filter adds an index of cited references to the bibliography, with links back to all in-text citations. It also allows the user to turn these off globally or in an *ad hoc* fashion.

💡 Turning off undesired linking

If you want to avoid undesired linking when citing specific fields, turn `link-fields` into false

ℹ️ Bibliography links

`link-citations`: Hyperlink citations to the corresponding bibliography entries. Defaults to true.

`link-fields`: Hyperlink citations targeting specific CSL fields to the corresponding entries in the bibliography. Defaults to true.

`link-bibliography`: Hyperlink DOIs, PMCIDs, PMID, and URLs in bibliographies. Defaults to true.

`lang`: Affects the bibliography tags. Defaults to `en-US`.

5 Quarto

5.1 Scrivener Project Templates

All sorts of internal **Scrivener Templates** have been included for convenience. They serve as starting points to create new sections. Click **Project > New From Template** and select the desired **Section Types** from the list, which includes Bibliography, Code, Computation, Diagram Dot, Diagram Mermaid, Div, Equation, File, Metadata, Section, Text, Text (Anchored)⁶.

This provides a huge number of options as the metadata can be customized to create many **Quarto** elements. Using the **Section Type Div**, for example, one could create 8 different **Amsthm** elements, 5 different **Callouts**, and several **Column** environments. Using the **Computation**, one can create executable code blocks with R, Python, and Ruby. The **Section Type Section** can be numbered, unnumbered, or part of the appendix (with the use of classes).

Look at the ready-made examples to see what else is possible.

5.2 Cross-referencing

When a **Section** is created, select the correct **ID-Prefix** (*e.g.* sec-), and fill the **ID** metadata field with a value (*e.g.* xref). Then, use Scrivener placeholders, such as @<\\$Custom:ID-Prefix><\\$Custom:ID> with a link to the cited element, so that this gets replaced with \@sec-xref. This works regardless of the element being cross-referenced (*e.g.* section, table, figure, listing) because this strategy ensures the citation will use the <\\$Custom:ID-Prefix> pulled from the targeted element (*e.g.* sec-, tbl-, fig-, lst-), making it compatible with all element types.

⚠ Link anchor

To be less verbose, we have set up a replacement rule that allows a shorter label to be used as the link anchor.

- s\crivlink is replaced with <\\$Custom:ID-Prefix><\\$Custom:ID>.
- s\crivpath and \$!\! are replaced with <\\$Custom:ID-Prefix><\\$Custom:ID><\\$Custom:Extension>.

⁶Text section with ID for cross-referencing.

 Cross-referencing an element

1. Type your-keyword-of-choice or `s\crivlink`, select it, and hit Command + L;
2. Link to the document that contains the table.
3. Apply a **Character Style** called *Cite*.

 Known limitation

Scrivener Placeholders can only pull information from the section properties, so this works when we are referencing elements created using **Section Types**.

When referencing elements created using **Raw Markup** or a **Character Style**, we must use the same ID we gave the element (*e.g.* `fig-ulysses`) instead of our generic link label (*e.g.* `s\crivlink`).

5.3 Amsthm

Species	Markdown Source	Rendered Output
Conjecture	<code>[@cnj -demo]</code>	Conjecture 5.3.1
Corollary	<code>[@cor -demo]</code>	Corollary 5.3.1
Definition	<code>[@def -demo]</code>	Definition 5.3.1
Example	<code>[@exm -demo]</code>	Example 5.3.1
Exercise	<code>[@exr -demo]</code>	Exercise 5.3.1
Lemma	<code>[@lem -demo]</code>	Lemma 5.3.1
Proposition	<code>[@prp -demo]</code>	Proposition 5.3.1
Theorem	<code>[@thm -demo]</code>	Theorem 5.3.1

Table 5.3: Cross-referencing **Amsthm** elements in ScrivQ.

Conjecture 5.3.1: Demonstration of the **Conjecture** theorem environment using the **Section Type** `Div` with **ID #cnj -demo**.

Corollary 5.3.1: Demonstration of the **Corollary** theorem environment using the **Section Type** `Div` with **ID #cor -demo**.

Definition 5.3.1: Demonstration of the **Definition** theorem environment using the **Section Type** Div with ID #def-demo.

Example 5.3.1: Demonstration of the **Example** theorem environment using the **Section Type** Div with ID #exm-demo.

Exercise 5.3.1: Demonstration of the **Exercise** theorem environment using the **Section Type** Div with ID #exr-demo.

Lemma 5.3.1: Demonstration of the **Lemma** theorem environment using the **Section Type** Div with ID #lem-demo.

Proposition 5.3.1: Demonstration of the **Proposition** theorem environment using the **Section Type** Div with ID #prp-demo.

Theorem 5.3.1: Demonstration of the **Theorem** theorem environment using the **Section Type** Div with ID #thm-demo.

$$[x^2 + y^2 = z^2] \quad (5.1)$$

5.4 Callouts

Species	Markdown Source	Rendered Output
Caution	[@cau-caution]	Caution 1
Important	[@imp-important]	Important 1
Note	[@nte-note]	Note 1
Tip	[@tip-tip]	Tip 1
Warning	[@wrn-warning]	Warning 2

Table 5.4: Cross-referencing callouts.

Using styles, you can create normal or collapsed callouts.

⚠ Caution (collapsed)

This is a Callout Caution using a **Paragraph Style**.

⚠ Caution

This is a Callout Caution using a **Paragraph Style**.

⚠ Caution 5.1

Demonstration of a **Callout Caution** using the **Section Type Div** with class `.callout-caution` and with ID `#cau-caution`.

❗ Important 5.1

Demonstration of a **Callout Important** using the **Section Type Div** with class `.callout-important` and with ID `#imp-important`.

ℹ Note 5.1

Demonstration of a **Callout Note** using the **Section Type Div** with class `.callout-note` and with ID `#nte-note`.

💡 Tip 5.1

Demonstration of a **Callout Tip** using the **Section Type Div** with class `.callout-tip` and with ID `#tip-tip`.

⚠ Warning 5.2

Demonstration of a **Callout Warning** using the **Section Type Div** with class `.callout-warning` and with ID `#wrn-warning`.

5.5 Diagrams

Similarly, we can create **Dot** and **Mermaid** diagrams using **Section Types** (*Diagram Dot*, *Diagram Mermaid*), **Paragraph Styles** (*Diagram Dot*, *Diagram Mermaid*), and **Raw Markdown**.

Species	Markdown Source	Rendered Output
Dot	<code>[@fig-dot-a]</code>	Figure 5.4
Dot	<code>[@fig-dot-b]</code>	Figure 5.5
Dot	<code>[@fig-dot-c]</code>	Figure 5.6
Mermaid	<code>[@fig-mermaid-a]</code>	Figure 5.7
Mermaid	<code>[@fig-mermaid-b]</code>	Figure 5.8
Mermaid	<code>[@fig-mermaid-c]</code>	Figure 5.9

Table 5.5: Cross-referencing **Dot** and **Mermaid** diagrams.

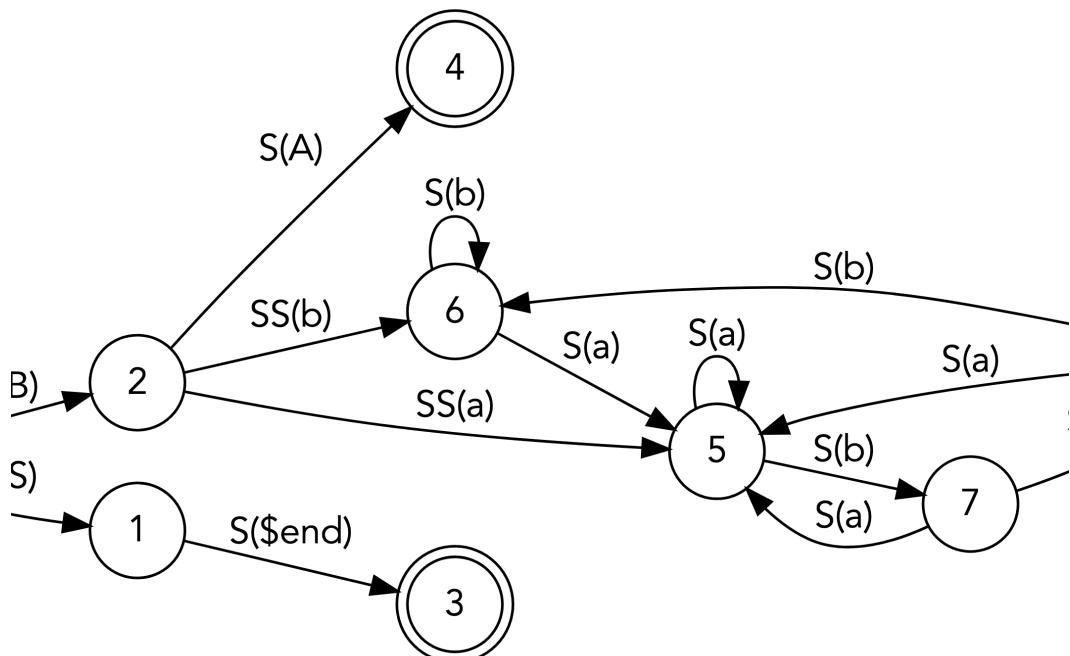


Figure 5.4: Figure caption

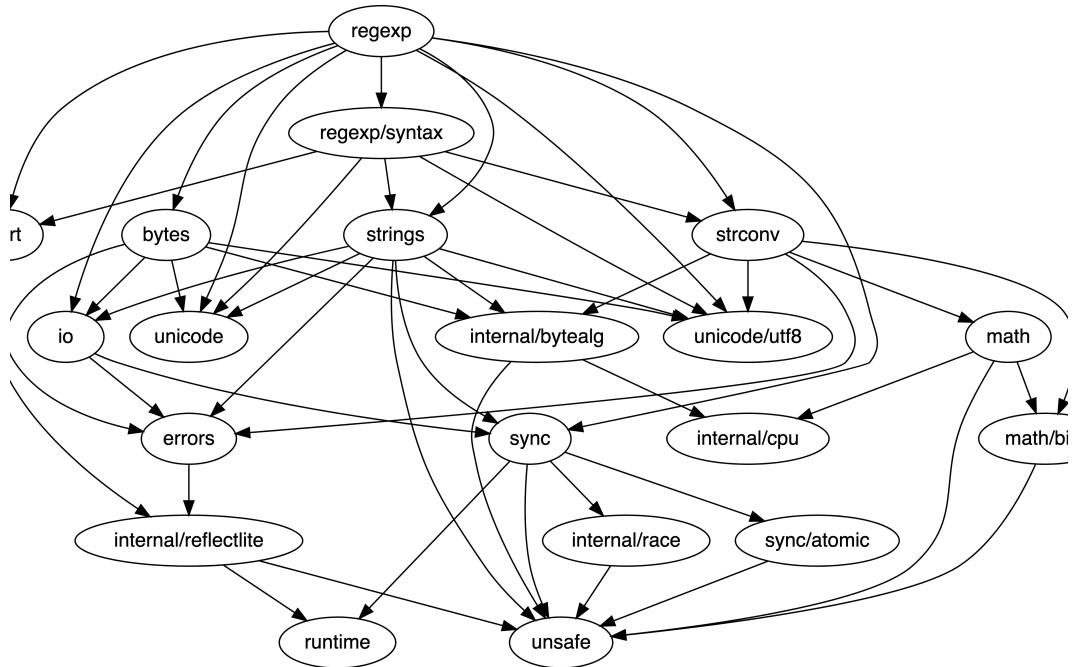


Figure 5.5: A graphviz graph with figure reference and caption, using raw markup.
See <https://quarto.org/docs/authoring/diagrams.html#sizing> for more details...

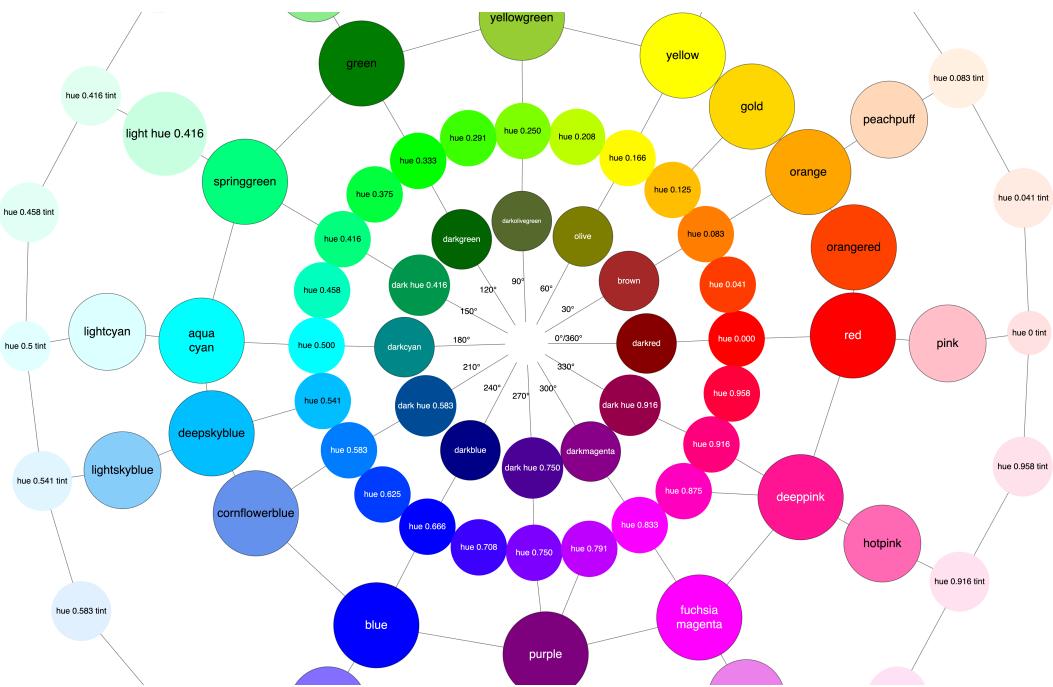


Figure 5.6: Color wheel diagram

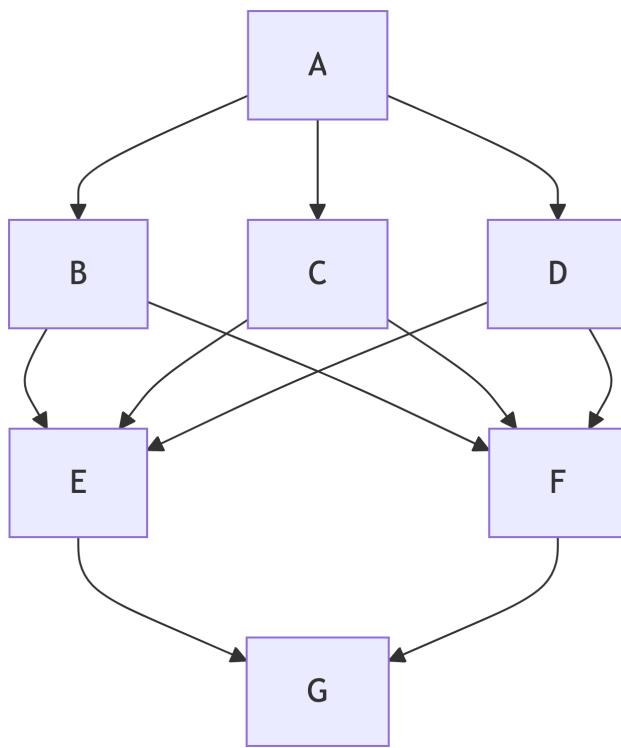


Figure 5.7: Figure caption

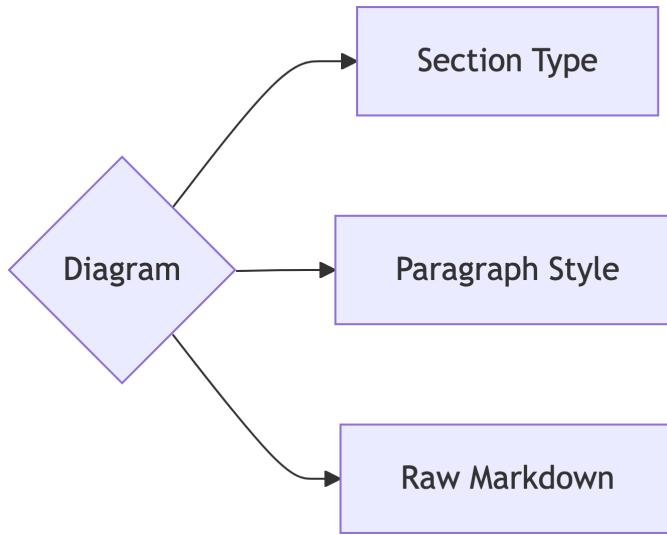


Figure 5.8: Figure caption

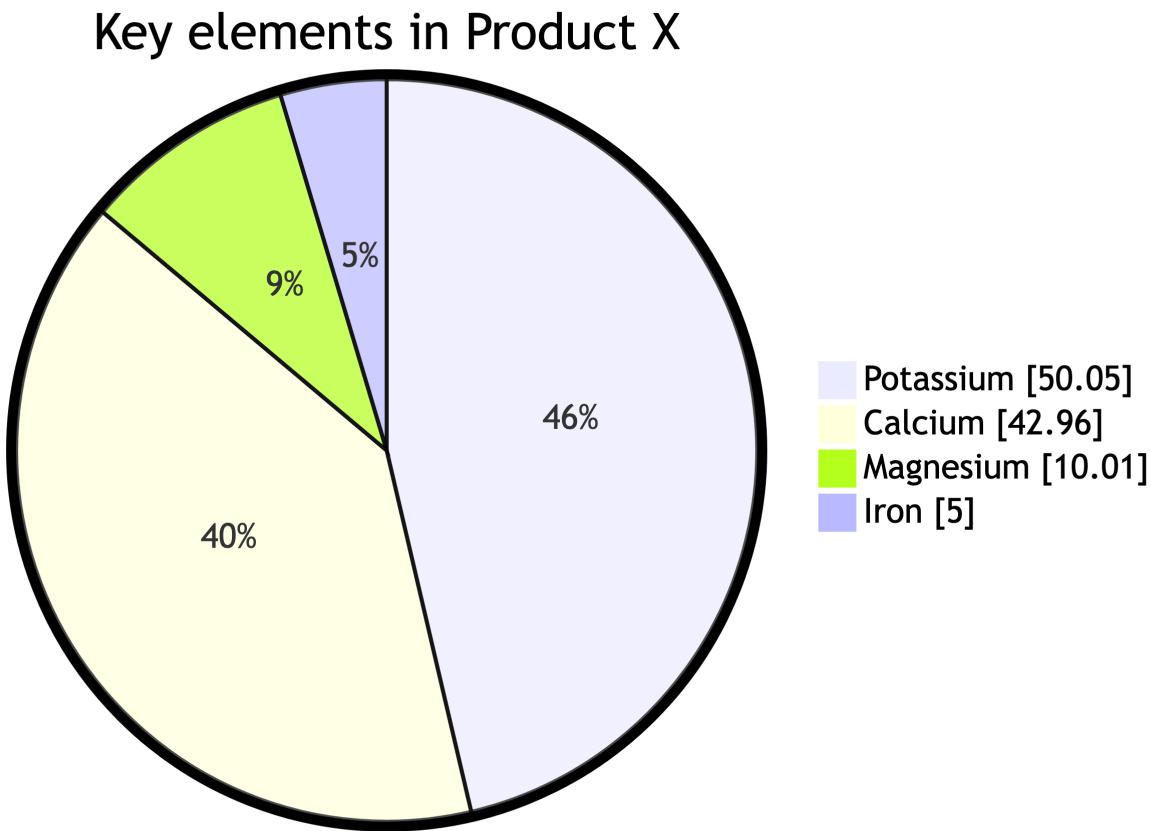


Figure 5.9: A Mermaid figure using a Scrivener Section Type [Computation] with class [mermaid], see <https://quarto.org/docs/authoring/diagrams.html> for more details

5.6 Equations

Species	Markdown Source	Rendered Output
Equation	[@eq-demo-a]	Equation 2
Equation	[@eq-demo-b]	Equation 3

Table 5.6: Cross-referencing **equations**.

$$t' = \frac{t - \frac{v}{c^2}x}{\sqrt{1 - \frac{v^2}{c^2}}} \quad (2)$$

$$t' = \frac{t - \frac{v}{c^2}x}{\sqrt{1 - \frac{v^2}{c^2}}} \quad (3)$$

5.7 Figures

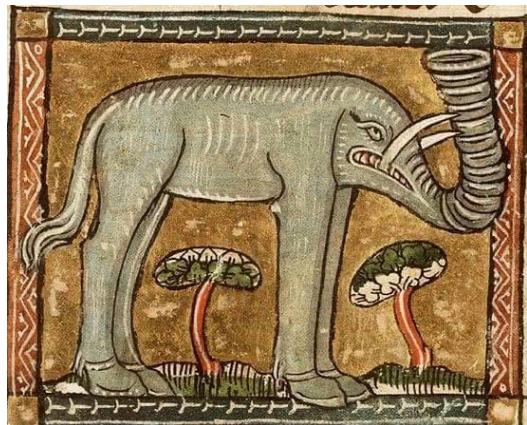
Species	Markdown Source	Rendered Output
	[@fig-ulysses]	Figure 5.10
Multipart Figure	[@fig-panel-a]	Figure 5.11
Multipart Figure	[@fig-panel-a-item-a]	Figure 11a
Multipart Figure	[@fig-panel-a-item-b]	Figure 11b

Table 5.7: Cross-referencing figures.



Figure 5.10: Ulysses and the Sirens.

Painting by [DRAPER, 1909](#).



(b) Angry elephant with a big trunk.

(a) Place the label first in the caption,
and use the `Caption` style.

Figure 5.11: This multi-figure panel uses the `Section Type Div` instead of raw markdown as shown here. ID, Class, and Attributes specific to the block `[#fig-panel-a .column-body layout-ncol=2 layout-valign="bottom"]` are saved to Custom Metadata->ID, Class & Attributes, and then inserted into the markup for this chunk by the Section Layout at compile time.

5.8 Listings

Species	Markdown Source	Rendered Output
Listing	<code>[@lst-demo-a]</code>	Listing 5.2
Listing	<code>[@lst-demo-b]</code>	Listing 5.3

Table 5.8: Cross-referencing listings.

Listing 5.2: Decomposition of Unicode characters.

```
require "unicode/name"

characters = %w(α β Ӯ ӹ ӻ)

# characters = 'ӻ'
characters.each do |character|
  puts character.unpack('U*').map { |i|
    "U+#{i.to_s(16).rjust(4, '0')}.upcase"
  }.join
  puts Unicode::Name.of character
end
```

Listing 5.3: The caption

```
#!/usr/bin/env ruby
# frozen_string_literal: false

Encoding.default_external = Encoding::UTF_8

Dir["#{__dir__}/Ruby/**/*.rb"].each do |file|
  require_relative file
end
```

5.9 Tables

Species	Markdown Source	Rendered Output
	[@tbl-demo-a]	Table 5.10
	[@tbl-demo-b]	Table 5.11
Multipart Table	[@tbl-panel-a]	Table 5.12
Multipart Table	[@tbl-panel-a-item-a]	Table 12a
Multipart Table	[@tbl-panel-a-item-b]	Table 12b

Table 5.9: Cross-referencing tables.

GRC	SKT
ἐν ἀρχῇ ἦν ὁ λόγος	ଆଦୌ ବାଦ ଆସିତ୍ର

Table 5.10: This table with a passage from John 1.1 uses the **Section Type** Text and **Paragraph Style** Table Caption.

GRG	SKT
ἐν ἀρχῇ ἦν ὁ λόγος	आदौ वाद आसीत्

Table 5.11: “This is an example of **Section Type Table**. The caption and the remaining attributes are added as part of the Section Type markup.”

Element	Prefix	Markdown Source	Rendered Output
Equation A	eq	A	B
Equation A	eq	C	D
Listing A	lst	E	F

(a) The first table of the multipart table panel.

Element	Prefix	Markdown Source	Rendered Output
Equation B	eq	A	B
Equation B	eq	C	D
Listing B	lst	E	F

(b) The second table of the multipart table panel.

Table 5.12: This is a markdown multi-table panel with two sub-tables generated using a **Section Type Div**. The **Custom Metadata** holds the cross-referencing label, classes, and other attributes.

5.10 Sections

Genus	Markdown Source	Rendered Output
Section	[@sec-demo-a]	Section 5.10.a
Break + Section	[@sec-demo-e]	Section 5.10.b
Heading	[@sec-demo-c]	Section 5.10.c
Break + Heading	[@sec-demo-d]	Section 5.10.d

Table 5.13: Note that the unnumbered section cannot be referenced.

Section (Unnumbered)

Demonstration of the **Section Type Section** using **Class .unnumbered**.

5.10.a Section

Demonstration of the **Section Type** *Section* with **ID** #sec-demo-a.

5.10.b Break + Section

Demonstration of the **Section Type** *Break + Section* with **ID** #sec-demo-e.

5.10.c Heading

5.10.d Break + Heading

6 Templates and partials



Quarto Templates optionally edited in Scrivener

Users needing control over the parameters in the native **Quarto** templates shouldn't have to deal with external files. We imported all the templates and partials for the main file types (TeX, HTML, Typst) so they can be edited directly in Scrivener.



PDF

- [doc-class](#)
- [title](#)
- [toc](#)
- [before-body](#)
- [before-bib](#)
- [biblio](#)
- [after-body](#)

And the Pandoc sub-partials:

- [tightlist](#)
- [tables](#)
- [graphics](#)
- [citations](#)



HTML

- [title-block](#)
- [styles](#)
- [html-template](#)
- [html-styles](#)
- [toc](#)
- [metadata](#)

7 Resources

- [Bootstrap Icons](#) - These are available in Quarto documents using the **Shortcode Font Awesome** style as in . (There is also **Shortcode Env**, **Shortcode Meta**, **Shortcode Var**).
- [The Plain Person's Guide to Plain Text Social Science](#)
- [Quarto Reference](#)
- The easiest way to [publish to Github Pages](#)
- [Example of Quarto Book](#)
- [Quarto with GH Pages](#)

8 Final word

If you like what you see, consider sponsoring [this project on Github](#).

⚠ Known problems & random errors

- Compilation fails for **LaTeX → PDF** when citations are placed in Table/Figure captions. The cause seems to be the **Citation Backlinks** filter.
- For **Typst → PDF** output some **Quarto** features (e.g. margin notes, column classes) are not yet implemented. Hopefully this will change in future Quarto versions.

Bibliography

Primary Sources

ARISTOTELIS. "De Anima". Em: BEKKER, I. (Ed.). *Aristotelis Opera*. Trad.: B. Τατάκης. Berlin: Reimer, 1834.

Secondary Sources

ARISTOTELIS. *De Anima*. Ed: W. Biehl. Leipzig: Teubner, 1896.

ARISTOTELIS. *De Anima*. Ed: W. Biehl. Trads.: Willy Theiler & Horst Seidl. Harmburg: Felix Meiner, 1995.

LONG, C. *Ethics of Ontology*. Albany: SUNY, 2004. [[1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#)]

Workflow

HOFFMAN, D. D.; PRAKASH, C. [Objects of consciousness](#). *Frontiers in Psychology*, v. 5, p. 577, 2014. [[1](#)]

Songs

MORPHINE et al. *Cure For Pain*. : Cure For Pain.Rykodisc, 1993. Disponível em: <<https://open.spotify.com/track/3h09gaVixKDoYDrlTBrEWf?si=0668baf1aab345d4>>

Paintings

DRAPER, H. J. *Ulysses and the Sirens*., 1909. [[1](#)]