

Overview

Quarto is an open-source scientific and technical publishing system built on Pandoc. It supports Python, R, Julia, and Observable, combining code, narrative, and visualization into reproducible documents. A single `.qmd` source file can generate HTML, PDF, Word, presentations, websites, books, and more.

Core CLI Commands

- `quarto render doc.qmd`: Render a document.
- `quarto render doc.qmd --to pdf`: Render to a specific format.
- `quarto preview doc.qmd`: Render and open a live-reloading preview.
- `quarto check`: Check the installation environment.
- `quarto create project <type> <name>`: Create a new project (`type`: default, website, blog, book, manuscript).
- `quarto publish <provider>`: Publish a document or site.
- `quarto convert doc.ipynb`: Convert a Jupyter notebook to `.qmd`.
- `quarto inspect doc.qmd`: Inspect document metadata.

Authoring

Markdown Basics

- Headers: `# H1`, `## H2`, `### H3`, `#### H4`
- Emphasis: `**bold**`, `*italic*`, `***bold italic**`, `~~strikethrough~~`
- Inline code: `code`
- Links: `[text](https://example.com)`, `[text](url "title")`
- Images: `![alt text](image.png)`, `![alt text](image.png){width=50%}`
- Horizontal rule: `---` or `***`
- Line break: two trailing spaces or `\`
- Blockquote: `> quoted text`
- Definition list:


```
Term
: Definition
```
- Footnote: `text[^1] → [^1]: Footnote content`

Figures

```
![[Caption text]](image.png){#fig-label
width="80%"}
```

```
<!-- Reference -->
As shown in @fig-label...
```

```
<!-- Side-by-side layout -->
::: {layout-ncol=2}
![[Left]](left.png)
![[Right]](right.png)
:::
```

```
<!-- Sub-figures -->
::: {#fig-main}
![[A]](a.png){#fig-a}
![[B]](b.png){#fig-b}
Main caption
:::
```

Tables

```
<!-- Basic pipe table -->
| Col 1 | Col 2 | Col 3 |
|-----|:-----|:-----|
| Left  | Center | Right  |
| Data  | Value  | 100   |
: Table caption {#tbl-label}
```

```
<!-- Reference -->
See @tbl-label for details.
```

Diagrams

```
```{mermaid}
flowchart LR
 A[Start] --> B{Condition}
 B -->|Yes| C[Execute]
 B -->|No| D[End]
...

```{dot}
graph G {
  A -- B
  B -- C
  A -- C
}
...

```

Citations & Bibliography

```
---
bibliography: references.bib
csl: apa.csl
---
```

- Cite: `@smith2020`, `[@smith2020]`, `[@smith2020, p.10]`
- Multiple: `[@smith2020; @jones2021]`
- Suppress author: `[-@smith2020]`
- `references.bib` example:


```
@article{smith2020,
  author = {Smith, J.},
  title  = {Title},
  year  = {2020},
  journal = {Journal Name}
}
```

Cross-References

- `{#fig-id}`: Figure label → reference with `@fig-id`
- `{#tbl-id}`: Table label → reference with `@tbl-id`
- `{#eq-id}`: Equation label → reference with `@eq-id`
- `{#sec-id}`: Section label → reference with `@sec-id`
- `{#lst-id}`: Code listing label → reference with `@lst-id`

```
## Introduction {#sec-intro}
```

```
$$ E = mc^2 $$ {#eq-energy}
```

```
As discussed in @sec-intro, @eq-energy shows...
```

Callout Blocks

```
::: {callout-note}
Note: General information to highlight.
:::
```

```
::: {callout-tip}
Tip: Useful advice.
:::
```

```
::: {callout-warning}
Warning: Requires attention.
:::
```

```
::: {callout-important}
Important: Must-know information.
:::
```

```
::: {callout-caution}
Caution: Potentially dangerous content.
:::
```

```
<!-- Custom title and collapsible -->
::: {callout-note collapse="true"
title="Custom Title"}
Content here.
:::
```

Article Layout

```
::: {.column-margin}
Margin note content
:::
```

```
::: {.column-page}
Full page-width content
:::
```

```
::: {.column-screen}
Full screen-width content
:::
```

Shortcodes

- `{{< include file.qmd >}}`: Include another file
- `{{< video url >}}`: Embed video (`{{< video https://youtu.be/xxx >}}`)
- `{{< embed notebook.ipynb >}}`: Embed a Jupyter notebook cell
- `{{< lipsum >}}`: Insert placeholder text
- `{{< placeholder >}}`: Insert placeholder image
- `{{< pagebreak >}}`: Insert a page break

YAML Front Matter

Defined at the top of the file between `---` delimiters.

```
---
title: "Document Title"
subtitle: "Subtitle"
author:
  - name: "Jane Smith"
    email: jane@example.com
```

```

  affiliation: "University"
date: "2024-01-01"
date-modified: last-modified
abstract: |
  Abstract text here.
keywords: ["keyword1", "keyword2"]
lang: en
format:
  html:
    toc: true
    toc-depth: 3
    toc-location: left
    number-sections: true
    theme: cosmo
    code-fold: true
    df-print: paged
  pdf:
    documentclass: article
    geometry: margin=2cm
    keep-tex: false
  docx:
    reference-doc: template.docx
bibliography: refs.bib
execute:
  echo: true
  warning: false
  cache: true
---
```

Computations

Execution Engines

- **Jupyter**: Python, Julia, R (IRKernel), and other kernels
- **Knitr**: R-only, processes `{r}` code chunks

Code Chunk Options (using `#|` prefix)

```

```{python}
#| label: fig-scatter # Label
(for cross-referencing)
#| fig-cap: "Scatter Plot" # Figure
caption
#| echo: trued # Show code
#| eval: trued # Run code
#| output: trued # Show output
#| warning: false # Hide
warnings
#| error: falsed # Hide errors

```

```

#| include: true # Include
code+output
#| code-fold: true #
Collapsible code (HTML)
#| code-summary: "Show code" # Collapsed
label
#| fig-width: 6d # Figure width (inches)
#| fig-height: 4 # Figure
height (inches)
#| fig-align: center # Figure
alignment
#| cache: trued # Cache results
#| dependson: "other-label" # Cache
dependency
#| column: paged # Layout column
position

```

```

import matplotlib.pyplot as plt
import numpy as np
x = np.linspace(0, 10, 100)
plt.plot(x, np.sin(x))
plt.show()

```

### Using Python

```

engine: jupyter
jupyter: python3

```

- Inline code: `{python} x + 1`
- Use virtual env: `quarto render doc.qmd --execute-python ./venv/bin/python`

### Using R

```

engine: knitr

```{r}
#| label: fig-r-plot
#| fig-cap: "R Plot"
library(ggplot2)
ggplot(mtcars, aes(wt, mpg)) +
  geom_point()
---
```

Using Julia

```

---
engine: julia
---
```

```

```{julia}
using Plots
plot(sin, 0, 2π)

```

### Using Observable JS

```

```{ojs}
viewof bins = Inputs.range([1, 100],
{value: 40, step: 1, label: "Bins"})

Plot.plot({
  marks: [Plot.rectY(data, Plot.binX({y:
"count"}, {x: "value", thresholds:
bins}))]
})
---
```

- Pass data from Python/R: `ojs_define(data = my_df)`
- Access in OJS: `transpose(data)`

Execution Options (Global)

```

---
execute:
  echo: true          # Default for all
chunks
  warning: false
  cache: true
  freeze: auto       # Reuse cached
output unless source changes
---
```

Parameters

```

---
params:
  year: 2024
  region: "Seoul"
  alpha: 0.05
---
# Access in R
params$year
• Inject at render time: quarto render doc.qmd -P year:2025 -P region:"Busan"
---
```

Tools

VS Code

- Install the **Quarto** extension
- Preview: `Ctrl+Shift+K`

- Render: Command Palette → **Quarto: Render**
- Toggle Visual Editor: `Ctrl+Shift+F4`

RStudio IDE

- Create `.qmd` file → Click Render button
- Shortcut: `Ctrl+Shift+K` (render)
- Run chunk: `Ctrl+Shift+Enter`
- Toggle Visual Editor: Source/Visual tab

JupyterLab

- Install extension: `pip install jupyterlab-quarto`
- Edit `.qmd` files like Jupyter notebooks
- Run cells interactively, then `quarto render` for final output

Positron

- Built-in Quarto support
- Both notebook editor and Visual Editor supported

Neovim

- Install `quarto-nvim` plugin
- Supports LSP and cell execution

Output Formats

HTML

```

---
format:
  html:
    toc: true
    toc-depth: 3
    toc-location: left      # left /
right / body
    number-sections: true
    theme: cosmod          # cosmo, flatly,
darkly, etc.
    highlight-style: github
    code-fold: true
    code-tools: true
    code-line-numbers: true
    df-print: paged
    self-contained: true   # Embed
all dependencies
    page-layout: full      #
article / full / custom
---
```

PDF

```

---
format:
  pdf:
    documentclass: article # article,
report, book
    classoption: [twocolumn]
    geometry:
      - margin=2cm
    fontsize: 11pt
    linestretch: 1.5
    keep-tex: false
    cite-method: biblatex
    include-in-header: header.tex
---

```

MS Word

```

---
format:
  docx:
    reference-doc: my-template.docx
    toc: true
    number-sections: true
    fig-width: 6
    fig-height: 4
---

```

Typst

```

---
format:
  typst:
    toc: true
    section-numbering: "1.1.a"
    columns: 2
---

```

Presentations

Revealjs (HTML Slides)

```

---
format:
  revealjs:
    theme: moon
    slide-number: true
    chalkboard: true
    transition: slide
    incremental: false
---

```

- Slide separator: --- (horizontal), ## (vertical)

- Incremental lists: ::: {.incremental}
- Fragments: ::: {.fragment}
- Speaker notes: ::: {.notes} Notes here :::
- Speaker view: S key

PowerPoint

```

---
format:
  pptx:
    reference-doc: template.pptx
---

```

Beamer (PDF Slides)

```

---
format:
  beamer:
    theme: Madrid
    colortheme: dolphin
---

```

Dashboards

```

---
format: dashboard
---

```

- Rows: ##, Cards: ###
- Tabs: {.tabset}
- Card height: {height=300}
- Value boxes:


```

      #| content: valuebox
      #| title: "Total Sales"
      #| icon: currency-dollar
      #| color: success
      dict(value = "$1.2M")
      
```
- Sidebar: ## {.sidebar}

Websites

```

# _quarto.yml
project:
  type: website
website:
  title: "My Site"
  navbar:
    left:
      - text: "Home"
        href: index.qmd
      - text: "About"
        href: about.qmd
  sidebar:
    - title: "Guide"

```

```

    contents:
      - guide/intro.qmd
      - guide/advanced.qmd
  page-footer:
    left: "@ 2024 My Site"
format:
  html:
    theme: cosmo

```

- Create blog: `quarto create project blog my-blog`
- Post listing pages use `listing` type

Books

```

# _quarto.yml
project:
  type: book
book:
  title: "My Book"
  author: "Author Name"
  date: "2024"
  chapters:
    - index.qmd
    - intro.qmd
    - part: "Part 1"
      chapters:
        - chapter1.qmd
        - chapter2.qmd
    - references.qmd
format:
  html:
    theme: cosmo
  pdf:
    documentclass: book

```

Manuscripts

```

---
title: "Paper Title"
format:
  html: default
  pdf:
    journal: nature
    keep-tex: true
  docx: default
---

```

- Embed notebook output: `{< embed notebook.ipynb#fig-id >}`
- Publish: `quarto publish quarto-pub` or create submission zip

Interactivity

Observable JS

```

```{ojjs}
viewof bins = Inputs.range([1, 100],
{value: 40, step: 1, label: "Bins"})

Plot.plot({
 marks: [Plot.rectY(data, Plot.binX({y:
"count"}, {x: "value", thresholds:
bins}))]}
})
```

```

Shiny (R)

```

---
server: shiny
---
```{r}
sliderInput("n", "Sample Size:", min=10,
max=1000, value=100)
```
```{r}
#| context: server
output$plot ←
renderPlot({ hist(rnorm(input$n)) })
```
```{r}
plotOutput("plot")
```

```

Shiny for Python

```

---
server: shiny
---
```{python}
from shiny import ui
ui.input_slider("n", "Sample Size:", 10,
1000, 100)
```
```{python}
#| context: server
from shiny import render
import numpy as np, matplotlib.pyplot as plt

@render.plot
def hist():

```

```
plt.hist(np.random.normal(size=input.n()))
```

```

Jupyter Widgets

```
```{python}
import ipywidgets as widgets
w = widgets.IntSlider(value=5, min=0,
max=10)
display(w)
```
```

Publishing

Quarto Pub (Free)

```
quarto publish quarto-pub doc.qmd
# Or the entire project
quarto publish quarto-pub
```

GitHub Pages

```
quarto publish gh-pages
# Creates _publish.yml automatically
# .github/workflows/publish.yml
on:
  push:
    branches: [main]
jobs:
  build-deploy:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
      - uses: quarto-dev/quarto-actions/
setup@v2
      - uses: quarto-dev/quarto-actions/
publish@v2
    with:
      target: gh-pages
```

Netlify

```
quarto publish netlify
```

Posit Connect / Posit Connect Cloud

```
quarto publish connect
quarto publish posit-cloud
```

Confluence

```
quarto publish confluence doc.qmd
```

Projects

Project Structure

```
myproject/
├── _quarto.yml      # Project
├── configuration
├── index.qmd
├── about.qmd
├── posts/
│   └── my-post.qmd
```

._quarto.yml Key Options

```
project:
  type: website
  output-dir: _site
  render:
    - "*.qmd"
    - "!drafts/"

execute:
  freeze: auto      # Rerun only when
source changes
  cache: true
```

Project Profiles

```
quarto render --profile dev
# _quarto-dev.yml
format:
  html:
    theme: sketchy
execute:
  cache: false
• Also set via QUARTO_PROFILE environment
variable
```

Environment Variables

```
# .env file
API_KEY=mykey
DB_URL=localhost:5432
• Quarto automatically reads .env into the envi-
ronment
```

Virtual Environments

```
# Python venv
python -m venv .venv
quarto render # Auto-detected
```

```
# R renv
```

```
renv::init()
quarto render
```

```
# conda/mamba
conda env create -f environment.yml
quarto render --execute-python /path/to/
conda/python
```

Project Scripts

```
# _quarto.yml
project:
  pre-render: setup.R      # Run before
rendering
  post-render: cleanup.sh # Run after
rendering
```

Execution Management

- `quarto render --cache-refresh`: Force cache refresh
- `quarto render --no-cache`: Disable cache
- `quarto render --execute-debug`: Debug execution output

Advanced

Includes

```
{{< include _header.qmd >}}
```

Body content...

```
{{< include _footer.qmd >}}
```

- Write shared content in `_`-prefixed files; they won't render standalone.

Variables

```
# _variables.yml
version: "1.0.0"
email: "contact@example.com"
Version {{< var version >}}, contact {{<
var email >}}.
```

Page Layout

```
::: {.column-margin}
Margin note
:::
```

```
::: {.column-body-outset}
Slightly wider than body
```

```
:::
::: {.column-page}
Full page width
:::
```

```
::: {.column-screen}
Full screen width
:::
```

Document Language

```
---
lang: en      # BCP 47 language tag
---
```

- Auto-translates cross-reference labels: “Figure”, “Table”, “Section”, etc.
- Supported: `en`, `ko`, `ja`, `zh`, `de`, `fr`, `es`, and more

Conditional Content

```
::: {.content-visible when-
format="html"}
Only visible in HTML.
:::
```

```
::: {.content-hidden when-format="pdf"}
Hidden in PDF output.
:::
```

```
::: {.content-visible unless-
format="docx"}
Visible in all formats except Word.
:::
```

Code Annotation

```
```python
x = 1 + 1 # <1>
y = x * 2 # <2>
```
```

1. Calculate x.
2. Calculate y.

Pandoc Filters

```
---
filters:
  - my-filter.lua
  - quarto
---
```

```
-- my-filter.lua
function Para(e1)
```

```
  return e1
end
```

Notebook Filters

```
---
notebook-filters:
- cleanup-filter.py
---
```

- Transform or remove cells before rendering `.ipynb` files.

Scholarly Writing

Math

- Inline: $E = mc^2$
- Block:
$$\int_{-\infty}^{\infty} e^{-x} dx = 1$$
- HTML math engine: `html-math-method: mathjax` or `katex`

Title Blocks

```
---
title: "Paper Title"
author:
- name: Jane Smith
  orcid: 0000-0000-0000-0000
  email: jane@example.com
  affiliations:
  - name: University of Example
    city: New York
    country: USA
date: 2024-01-01
abstract: |
  Abstract text here.
---
```

Citeable Articles

```
---
citation:
  url: https://example.com/my-article
  doi: "10.1234/example"
google-scholar: true
---
```

Appendices

```
# References {.unnumbered}
```

```
::: {#refs}
:::
```

```
# Appendix A {.appendix}
```

Appendix content here.

Code Block Styling

Syntax Highlighting

```
---
highlight-style: github # github,
monokai, zenburn, etc.
code-line-numbers: true
---
```

Code Folding (HTML)

```
---
format:
  html:
    code-fold: true
    code-summary: "Show code"
    code-tools: true # Adds copy
button
---
```