

소개 (Overview)

Quarto는 Pandoc 기반의 오픈 소스 과학 및 기술 출판 시스템입니다. Python, R, Julia, Observable을 지원하며, 코드-내러티브-시각화를 결합한 재현 가능한 문서를 만들 수 있습니다. 하나의 .qmd 소스 파일로 HTML, PDF, Word, 프레젠테이션, 웹사이트, 책 등 다양한 결과물을 생성합니다.

CLI 기본 명령어

- `quarto render doc.qmd`: 문서를 렌더링합니다.
- `quarto render doc.qmd --to pdf`: 특정 포맷으로 렌더링합니다.
- `quarto preview doc.qmd`: 라이브 프리뷰를 실행합니다.
- `quarto check`: 설치 환경을 점검합니다.
- `quarto create project <type> <name>`: 새 프로젝트를 생성합니다 (`type`: default, website, blog, book, manuscript).
- `quarto publish <provider>`: 문서 또는 사이트를 배포합니다.
- `quarto convert doc.ipynb`: Jupyter 노트북을 .qmd로 변환합니다.
- `quarto inspect doc.qmd`: 문서 메타데이터를 확인합니다.

마크다운 작성 (Authoring)

마크다운 기본 (Markdown Basics)

- 제목: # H1, ## H2, ### H3, #### H4
- 강조: ****굵게****, *기울임*, ***굵고 기울임***, ~~ 취소선~~
- 인라인 코드: `코드`
- 링크: [텍스트](https://example.com), [텍스트](https://example.com "제목")
- 이미지: ![대체 텍스트](image.png), ![대체 텍스트](image.png){width=50%}
- 수평선: --- 또는 ***
- 줄바꿈: 줄 끝에 공백 두 칸 또는 \
- 블록 인용: > 인용문
- 정의 목록:
 - 용어
 - : 정의 내용
- 각주: 텍스트[^1] → [^1]: 각주 내용

그림 (Figures)

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

```
<!-- 참조 -->
@fig-label 에서 볼 수 있듯이...
```

```
<!-- 그림 레이아웃 (나란히) -->
::: {layout-ncol=2}
![[왼쪽 그림](left.png)]
![[오른쪽 그림](right.png)]
:::
```

```
<!-- 서브 그림 -->
::: {#fig-main}
![[A(a.png)]{#fig-a}]
![[B(b.png)]{#fig-b}]
메인 캡션
:::
```

표 (Tables)

```
<!-- 기본 파이프 테이블 -->
| 열 1 | 열 2 | 열 3 |
|-----|:----|:-----|
| 왼쪽 | 가운데 | 오른쪽 |
| 데이터 | 값 | 100 |
: 표 캡션 {#tbl-label}
```

```
<!-- 참조 -->
@tbl-label 을 참고하세요.
```

다이아그램 (Diagrams)

```
```{mermaid}
flowchart LR
 A[시작] --> B[조건]
 B -->|예| C[실행]
 B -->|아니오| D[종료]
```
```

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

인용 및 참고문헌 (Citations)

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

- 인용: @smith2020, [smith2020], [smith2020, p.10]
- 다중 인용: [smith2020; jones2021]
- 저자 억제: [-smith2020]
- references.bib 예시:


```
@article{smith2020,
  author = {Smith, J.},
  title = {제목},
  year = {2020},
  journal = {학술지}
}
```

상호 참조 (Cross-References)

- {#fig-id}: 그림 레이블, @fig-id로 참조
 - {#tbl-id}: 표 레이블, @tbl-id로 참조
 - {#eq-id}: 수식 레이블, @eq-id로 참조
 - {#sec-id}: 섹션 레이블, @sec-id로 참조
 - {#lst-id}: 코드 리스팅 레이블, @lst-id로 참조
- ```
소개 {#sec-intro}
```

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

앞서 @sec-intro 에서 설명한 @eq-energy 는...

### 콜아웃 블록 (Callout Blocks)

```
::: {callout-note}
참고: 일반 정보를 강조할 때 사용합니다.
:::
```

```
::: {callout-tip}
팁: 유용한 조언을 제공합니다.
:::
```

```
::: {callout-warning}
경고: 주의가 필요한 내용입니다.
:::
```

```
::: {callout-important}
중요: 반드시 알아야 할 내용입니다.
:::
```

```
::: {callout-caution}
주의: 위험할 수 있는 내용입니다.
:::
```

```
<!-- 제목 변경, 접기 -->
::: {callout-note collapse="true"}
```

```
title="커스텀 제목"
내용
:::
```

### 아티클 레이아웃 (Article Layout)

```
::: {column-margin}
여백 노트 내용
:::

::: {column-page}
페이지 전체 너비로 표시되는 콘텐츠
:::
```

```
::: {column-screen}
화면 전체 너비 콘텐츠
:::
```

### 숏코드 (Shortcodes)

- {< include file.qmd >}: 다른 파일 삽입
- {< video url >}: 비디오 삽입 ({< video https://youtu.be/xxx >})
- {< embed notebook.ipynb >}: Jupyter 노트북 셀 삽입
- {< lipsum >}: 더미 텍스트 삽입
- {< placeholder >}: 더미 이미지 삽입
- {< pagebreak >}: 페이지 나누기

## YAML 프론트 매터 (Front Matter)

문서 상단에 ---로 구분하여 메타데이터를 정의합니다.

```

title: "문서 제목"
subtitle: "부제"
author:
 - name: "홍길동"
 email: hong@example.com
 affiliation: "소속 기관"
date: "2024-01-01"
date-modified: last-modified
abstract: |
 이 문서의 요약입니다.
keywords: ["키워드1", "키워드2"]
lang: ko
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

```

## 코드 연산 (Computations)

### 실행 엔진

- **Jupyter:** Python, Julia, R (IRKernel), 기타 언어 지원
- **Knitr:** R 전용, {r} 코드 청크 실행

### 코드 청크 옵션 (#| 접두사)

```

```{python}
#| label: fig-scatter# 레이블 (상호참조용)
#| fig-cap: "산점도" # 그림 캡션
#| echo: true # 코드 출력 여부
#| eval: true # 코드 실행 여부
#| output: true # 결과 출력 여부
#| warning: false # 경고 숨김
#| error: false # 오류 숨김
#| include: true # 코드+결과 포함 여부
#| code-fold: true # 코드 접기 (HTML)
#| code-summary: "코드 보기" # 접힌 코드 라벨
#| fig-width: 6 # 그림 너비(인치)
#| fig-height: 4 # 그림 높이(인치)
#| fig-align: center # 그림 정렬
#| cache: true # 결과 캐시
#| dependson: "other-label" # 캐시 의존성
#| column: page # 레이아웃 열 위치

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

```

```

plt.plot(x, np.sin(x))
plt.show()
```
```

## Python 사용

```

engine: jupyter
jupyter: python3

```

- 인라인 코드: {python} x + 1
- 가상환경 지정: quarto render doc.qmd -- execute-python ./venv/bin/python

## R 사용

```

engine: knitr

```

```

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

## Julia 사용

```

engine: julia

```

```

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

## Observable JS 사용

```

```{ojjs}
viewof x = Inputs.range([0, 10], {step: 0.1, value: 5})
Plot.plot({marks:
  [Plot.line(d3.range(100), {x: d => d, y: d => Math.sin(d/10 * x)})]})
```
```

## 실행 옵션 (전역)

```

execute:
 echo: true # 전체 문서 기본값
 warning: false
 cache: true

```

```

freeze: auto # 캐시된 결과 재사용 여부

```

## 파라미터 (Parameters)

```

params:
 year: 2024
 region: "서울"
 alpha: 0.05

```

# R에서 접근  
params\$year

# Python에서 접근 (Jupyter)

- 렌더 시 주입: quarto render doc.qmd -P year:2025 -P region:"부산"

## 지원 도구 (Tools)

### VS Code

- 확장: Quarto 익스텐션 설치
- 프리뷰: Ctrl+Shift+K
- 렌더: 명령 팔레트 → Quarto: Render
- Visual Editor: Ctrl+Shift+F4로 전환

### RStudio IDE

- .qmd 파일 생성 후 Render 버튼 클릭
- 단축키: Ctrl+Shift+K (렌더)
- 청크 실행: Ctrl+Shift+Enter
- Visual Editor: 소스/비주얼 탭 전환

### JupyterLab

- JupyterLab 익스텐션 설치: pip install jupyterlab-quarto
- .qmd 파일을 Jupyter 노트북처럼 편집
- 셀 단위 실행 후 quarto render로 최종 렌더링

### Positron

- Quarto 내장 지원
- 노트북 편집기와 Visual Editor 모두 지원

### Neovim

- quarto-nvim 플러그인 설치
- LSP 및 코드 실행 지원

## 핵심 출력물 (Output Formats)

### HTML 문서

```

format:
 html:
 toc: true
 toc-depth: 3
 toc-location: left # left / right / body
 number-sections: true
 theme: cosmo # cosmo, flatly, darkly 등
 highlight-style: github
 code-fold: true
 code-tools: true
 code-line-numbers: true
 df-print: paged
 fig-cap-location: bottom
 self-contained: true # 외부 의존성 내장
 page-layout: full # article / full / custom

```

### PDF 문서

```

format:
 pdf:
 documentclass: article # article, report, book
 classoption: [twocolumn]
 geometry:
 - margin=2cm
 fontsize: 11pt
 linestretch: 1.5
 indent: true
 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

```

```
highlight-style: github
fig-width: 6
fig-height: 4

```

## Typst

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

```

## 프레젠테이션 (Presentations)

### Revealjs (HTML 슬라이드)

```
format:
 revealjs:
 theme: moon # default, moon, sky,
 beige 등
 slide-number: true
 chalkboard: true
 multiplex: true
 transition: slide
 incremental: false

```

- 슬라이드 구분: --- (수평), ## (수직)
- 점진적 목록: ::: {.incremental} 또는 -- incremental
- 단편 (Fragment): ::: {.fragment}
- 발표자 노트: ::: {.notes} 노트 내용 :::
- 발표자 뷰: S 키

### PowerPoint

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

```

### Beamer (PDF 슬라이드)

```
format:
 beamer:
 theme: Madrid
 colortheme: dolphin
 fonttheme: structurebold

```

## 대시보드 (Dashboards)

```
format: dashboard

```

- 행/열 레이아웃: ## 로 행, ### 으로 카드 구분
- {.tabset}: 탭 패널 생성
- 카드 높이: {height=300}
- 값 박스:

```
content: valuebox
title: "총 매출"
icon: currency-dollar
color: success
dict(value = "₩1.2억")
```

- 사이드바: ## {.sidebar}

## 웹사이트 (Websites)

```
_quarto.yml
project:
 type: website
website:
 title: "내 사이트"
 navbar:
 left:
 - text: "홈"
 href: index.qmd
 - text: "소개"
 href: about.qmd
 sidebar:
 - title: "가이드"
 contents:
 - guide/intro.qmd
 - guide/advanced.qmd
 page-footer:
 left: "© 2024 My Site"
```

- ```
format:
  html:
    theme: cosmo
```
- 블로그 생성: `quarto create project blog my-blog`
 - 포스트 목록: `listing` 타입 페이지 사용

책 (Books)

```
# _quarto.yml
project:
  type: book
book:
  title: "내 책"
  author: "저자명"
```

```
date: "2024"
chapters:
  - index.qmd
  - intro.qmd
  - part: "파트 1"
    chapters:
      - chapter1.qmd
      - chapter2.qmd
  - references.qmd
```

```
format:
  html:
    theme: cosmo
  pdf:
    documentclass: book
```

- 책 전체 상호 참조: 챕터 번호가 자동으로 접두사에 포함됨

원고 (Manuscripts)

```
title: "논문 제목"
format:
  html: default
  pdf:
    journal: nature
    keep-tex: true
  docx: default
---
```

- 노트북 삽입: `{{< embed notebook.ipynb#fig-id >}}`
- 배포: `quarto publish quarto-pub` 또는 저널 제출용 zip 생성

인터랙티브 기능 (Interactivity)

Observable JS

```
```{ojjs}
// 슬라이더로 대화형 시각화
viewof bins = Inputs.range([1, 100],
{value: 40, step: 1, label: "구간 수"})
```

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

- Python/R 데이터를 OJS로 전달: `ojs_define(data = my_df)`

- OJS에서 접근: `transpose(data)`

Shiny (R)

```
server: shiny
```{r}
sliderInput("n", "표본 수:", 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", "표본 수:", 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
from IPython.display import display
```

Last updated: 2026-05-08

```
w = widgets.IntSlider(value=5, min=0,
max=10)
display(w)
````
```

## 배포 (Publishing)

### Quarto Pub (무료)

```
quarto publish quarto-pub doc.qmd
또는 프로젝트 전체
quarto publish quarto-pub
```

### GitHub Pages

```
프리뷰 후 배포
quarto publish gh-pages
```

```
_publish.yml 자동 생성됨
CI/CD: .github/workflows/publish.yml
추가
```

```
.github/workflows/publish.yml 예시
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
```

## 프로젝트 관리 (Projects)

### 프로젝트 기본 구조

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

### \_quarto.yml 주요 옵션

```
project:
 type: website# default / website /
blog / book / manuscript
 output-dir: _site # 출력 디렉토리
 render: # 렌더링 포함/제외 패턴
 - "*.qmd"
 - "!drafts/"
```

```
execute:
 freeze: auto # auto: 변경 시만 재실행 /
true: 항상 캐시 사용
 cache: true
```

### 프로파일 (Project Profiles)

```
dev 프로파일로 렌더링
quarto render --profile dev
```

```
프로파일 파일: _quarto-dev.yml
_quarto-dev.yml
```

```
format:
 html:
 theme: sketchy
```

```
execute:
 cache: false
```

• `QUARTO_PROFILE` 환경변수로도 설정 가능

### 환경 변수

```
.env 파일
API_KEY=mykey
DB_URL=localhost:5432
```

• Quarto가 자동으로 .env 파일을 읽어 환경변수로 설정

### 가상 환경 연동

```
Python venv
python -m venv .venv
quarto render # 자동 감지
```

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

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

### 프로젝트 스크립트

```
_quarto.yml
project:
 pre-render: setup.R # 렌더링 전 스
 크립트
 post-render: cleanup.sh # 렌더링 후 스
 크립트
```

### 실행 관리 (Managing Execution)

- `quarto render --cache-refresh`: 캐시 강제 갱신
- `quarto render --no-cache`: 캐시 비사용
- `quarto render --execute-debug`: 실행 디버깅 출력

## 고급 기능 (Advanced)

### Includes (파일 삽입)

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

본문 내용...

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

- 공통 내용을 \_ 접두사 파일에 작성 후 여러 문서에서 재사용
- 삽입된 파일은 단독으로 렌더링되지 않음 (\_ 접두사)

### 변수 (Variables)

```
_variables.yml
version: "1.0.0"
email: "contact@example.com"
```

버전 `{{< var version >}}` 이며 문외는 `{{< var email >}}` 로 하세요.

### 페이지 레이아웃

```
::: {.column-margin}
여백에 표시될 내용
:::
```

```
::: {.column-body-outset}
본문 너비보다 약간 넓게
:::
```

```
::: {.column-page}
페이지 전체 너비
:::
```

```
::: {.column-screen}
화면 전체 너비
:::
```

### 문서 언어 (Document Language)

```

lang: ko # BCP 47 언어 태그

```

- 교차 참조 레이블 자동 번역: “Figure” → “그림”, “Table” → “표”
- 지원 언어: `ko, ja, zh, de, fr, es` 등

### 조건부 콘텐츠 (Conditional Content)

```
::: {.content-visible when-
format="html"}
HTML에서만 보이는 내용
:::
```

```
::: {.content-hidden when-format="pdf"}
PDF에서는 숨겨지는 내용
:::
```

```
::: {.content-visible unless-
format="docx"}
Word 제외 모든 포맷에서 표시
:::
```

### 코드 어노테이션 (Code Annotation)

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

1. x를 계산합니다.
2. y를 계산합니다.

### Pandoc 필터 (Filters)

```

filters:
 - my-filter.lua

```

```
- quarto

-- my-filter.lua 예시
function Para(e1)
 return e1
end
```

## 노트북 필터 (Notebook Filters)

```

notebook-filters:
- cleanup-filter.py

```

- .ipynb 렌더링 전에 셀을 변환하거나 제거하는 데 활용

## 코드 블록 스타일 (Code Blocks)

### 코드 하이라이팅

```

highlight-style: github
code-line-numbers: true

```

### 코드 접기 (HTML)

```

format:
 html:
 code-fold: true
 code-summary: "코드 보기"
 code-tools: true

```

### 실행 없이 표시

```
```python
# 이 코드는 실행되지 않습니다
print("Hello!")
```
```

## 학술 글쓰기 (Scholarly Writing)

### 수식 (Math)

- 인라인:  $E = mc^2$
- 블록:  
$$\int_0^{\infty} e^{-x} dx = 1$$
- HTML 수식 엔진: `html-math-method: mathjax` 또는 `katex`

## 제목 블록 (Title Blocks)

```

title: "논문 제목"
author:
- name: 홍길동
 orcid: 0000-0000-0000-0000
 email: hong@example.com
 affiliations:
 - name: 서울대학교
 city: 서울
 country: 대한민국
date: 2024-01-01
abstract: |
 논문 요약입니다.

```

## 인용 가능한 아티클 (Citeable Articles)

```

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

```

## 부록 (Appendices)

```
참고문헌 {.unnumbered}
```

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

```
부록 A {.appendix}
```

부록 내용