

11. Other Output Formats

PDF, presentations, and HTML

Dr. Paul Schmidt

So far, we have focused on Word documents. But the beauty of Quarto is: most of the knowledge transfers directly to other formats. In this chapter, you will learn how to render the same `.qmd` files to PDF, presentations, and HTML documents.

PDF with Typst

Why Typst?

For PDF export, there are two engines:

Engine	Advantages	Disadvantages
LaTeX	Very powerful, established	Slow, large installation
Typst	Fast, lightweight	Newer, fewer features

For the vast majority of scientific reports, I recommend **Typst**:

- No separate installation needed (integrated in Quarto)
- Rendering in seconds instead of minutes
- Sufficient for 99% of use cases

Creating a simple PDF

```
---
```

```
title: "Penguin Report"
format: typst
---
```

That is it! Quarto renders the document to PDF.

PDF-specific options

```
---
```

```
title: "Penguin Report"
format:
  typst:
    toc: true
    toc-depth: 2
    number-sections: true
    papersize: a4
    margin:
      x: 2.5cm
      y: 2.5cm
---
```

Typst vs. Word: Differences

Aspect	Word	Typst/PDF
Tables	flextable	knitr::kable or gt
Templates	reference-doc	Typst templates
Page layout	officedown	Typst commands

i Note

flextable works primarily for Word. For PDF, use `knitr::kable()` or the **gt** package (which works very well for PDF output).

Typst templates

For custom PDF layouts, you can use Typst templates:

```
format:
  typst:
    template: templates/custom.typ
```

Creating Typst templates requires knowledge of the Typst language, which goes beyond the scope of this tutorial.

Resources:

- Typst Documentation
- Quarto Typst Format

Presentations with Reveal.js

What is Reveal.js?

Reveal.js is a framework for HTML presentations. The slides are viewable in a browser, interactive, and can be presented like a normal presentation.

A simple presentation

```
---
title: "Penguin Presentation"
format: revealjs
---

# Introduction

This is the first slide.

## Subslide

- Point 1
- Point 2

# Methods

A new main slide.
```

The heading hierarchy determines the slide structure:

- `#` → New main slide (horizontal navigation)
- `##` → Subslide (vertical navigation)

Reveal.js options

```
format:
  revealjs:
    theme: simple
    slide-number: true
    transition: fade
    chalkboard: true
    smaller: true
---
```

Code on slides

```
# Code is displayed on the slide
ggplot(penguins, aes(x = bill_length_mm)) +
  geom_histogram()
```

With `echo: true` and `output-location: slide`, the output can appear on a separate slide.

Example presentations

Here are some public Quarto presentations for inspiration:

- Quarto Reveal.js Demo
- Awesome Quarto Talks

Resources:

- Quarto Reveal.js

HTML documents

Simple HTML

```
---
title: "Penguin Report"
format: html
---
```

HTML is the “native” format of Quarto — here all features work best.

HTML advantages

- Interactive tables (sortable, filterable)
- Interactive plots (zoom, hover)
- Code folding
- Search function
- Responsive design

HTML options

```
format:
  html:
    toc: true
```

```

toc-location: left
code-fold: true
code-tools: true
theme: cosmo
self-contained: true
---
```

This website

The website you are reading is a **Quarto Website** project. Instead of individual documents, many `.qmd` files are combined into a coherent website.

```

# _quarto.yml for websites
project:
  type: website

website:
  title: "My Website"
  navbar:
    left:
      - text: "Home"
        href: index.qmd
      - text: "Chapter 1"
        href: chapter1.qmd
```

Resources:

- Quarto Websites

Multiple formats at once

You can configure a document to render to multiple formats:

```

---
title: "Penguin Report"
format:
  docx:
    reference-doc: template.docx
  typst:
    toc: true
  html:
    code-fold: true
---
```

Then via command line:

```

quarto render report.qmd --to docx
quarto render report.qmd --to typst
quarto render report.qmd --to html
```

Format-specific code

Sometimes you need different code for different formats:

```

# Detect format
if (knitr:::is_html_output()) {
  # Interactive table for HTML
  DT:::datatable(my_data)
} else {
  # Static table for Word/PDF
  flextable:::flextable(my_data)
}
```

Summary: Format recommendations

Application	Recommended format
Report for colleagues	Word (docx)
Scientific publication	PDF (typst)
Online documentation	HTML
Presentation (meeting)	Reveal.js
Presentation (to send)	PDF or PowerPoint

The Quarto ecosystem

With the knowledge from this tutorial, you can:

- Create **reports** (Word, PDF)
- Give **presentations** (Reveal.js)
- Build **websites** (HTML)
- Write **books** (Quarto Book)
- Develop **dashboards** (Quarto Dashboard)

All use the same basic syntax — you now have the foundation for everything else.

💡 Final exercise: Multi-format report

1. Take your parameterized penguin report
2. Add `typst` as a second format
3. Render to Word and PDF
4. Compare the outputs
5. Bonus: Create a simple 3-slide presentation on the results

Further resources

- Quarto Guide — Comprehensive documentation
- Quarto Gallery — Examples of all formats
- Awesome Quarto — Community resources

Conclusion

You now have the knowledge to:

- Create reproducible Word documents
- Combine code, text, and results
- Integrate professional tables and plots
- Use templates for consistent branding
- Add cross-references and citations
- Create parameterized reports for different datasets
- Export the same document to different formats

Good luck with Quarto!

Bibliography
