

11. Andere Ausgabeformate

PDF, Präsentationen und HTML

Dr. Paul Schmidt

Bisher haben wir uns auf Word-Dokumente konzentriert. Aber das Schöne an Quarto ist: Das meiste Wissen überträgt sich direkt auf andere Formate. In diesem Kapitel lernt man, wie man dieselben `.qmd`-Dateien zu PDF, Präsentationen und HTML-Dokumenten rendern kann.

PDF mit Typst

Warum Typst?

Für PDF-Export gibt es zwei Engines:

Engine	Vorteile	Nachteile
LaTeX	Sehr mächtig, etabliert	Langsam, große Installation
Typst	Schnell, leichtgewichtig	Neuer, weniger Features

Für die allermeisten wissenschaftlichen Reports empfehle ich **Typst**:

- Keine separate Installation nötig (in Quarto integriert)
- Rendering in Sekunden statt Minuten
- Für 99% der Anwendungsfälle ausreichend

Einfaches PDF erstellen

```
---
```

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

Das war's! Quarto rendert das Dokument zu PDF.

PDF-spezifische Optionen

```
---
```

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

Typst vs. Word: Unterschiede

Aspekt	Word	Typst/PDF
Tabellen	flextable	knitr::kable oder gt
Templates	reference-doc	Typst-Templates
Seitenlayout	officedown	Typst-Befehle

i Hinweis

flextable funktioniert primär für Word. Für PDF verwendet man stattdessen `knitr::kable()` oder das **gt**-Paket (das bei PDF-Output sehr gut funktioniert).

Typst-Templates

Für benutzerdefinierte PDF-Layouts kann man Typst-Templates verwenden:

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

Das Erstellen von Typst-Templates erfordert Kenntnisse der Typst-Sprache, geht aber über den Rahmen dieses Tutorials hinaus.

Ressourcen:

- Typst Documentation
- Quarto Typst Format

Präsentationen mit Reveal.js

Was ist Reveal.js?

Reveal.js ist ein Framework für HTML-Präsentationen. Die Folien sind im Browser anzeigbar, interaktiv, und können wie eine normale Präsentation gehalten werden.

Einfache Präsentation

```
---
title: "Pinguin-Präsentation"
format: revealjs
---

# Einleitung

Dies ist die erste Folie.

## Unterfolie

- Punkt 1
- Punkt 2

# Methoden

Eine neue Hauptfolie.
```

Die Hierarchie der Überschriften bestimmt die Folienstruktur:

- # → Neue Hauptfolie (horizontale Navigation)
- ## → Unterfolie (vertikale Navigation)

Reveal.js-Optionen

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

Code auf Folien

```
# Code wird auf der Folie angezeigt
ggplot(penguins, aes(x = bill_length_mm)) +
  geom_histogram()
```

Mit `echo: true` und `output-location: slide` kann der Output auf einer separaten Folie erscheinen.

Beispiel-Präsentationen

Hier sind einige öffentliche Quarto-Präsentationen als Inspiration:

- Quarto Reveal.js Demo
- Awesome Quarto Talks

Ressourcen:

- Quarto Reveal.js

HTML-Dokumente

Einfaches HTML

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

HTML ist das “native” Format von Quarto — hier funktionieren alle Features am besten.

HTML-Vorteile

- Interaktive Tabellen (sortierbar, filterbar)
- Interaktive Plots (zoomen, hover)
- Code-Folding
- Suchfunktion
- Responsive Design

HTML-Optionen

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

Diese Website

Die Website, die du gerade liest, ist ein **Quarto Website**-Projekt. Statt einzelner Dokumente werden viele `.qmd`-Dateien zu einer zusammenhängenden Website kombiniert.

```
# _quarto.yml für Websites
project:
  type: website

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

Ressourcen:

- Quarto Websites

Mehrere Formate gleichzeitig

Man kann ein Dokument so konfigurieren, dass es in mehrere Formate gerendert werden kann:

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

Dann via Kommandozeile:

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

Format-spezifischer Code

Manchmal braucht man unterschiedlichen Code für verschiedene Formate:

```
# Format erkennen
if (knitr:::is_html_output()) {
```

```

# Interaktive Tabelle für HTML
DT::datatable(my_data)
} else {
# Statische Tabelle für Word/PDF
flextable::flextable(my_data)
}

```

Zusammenfassung: Format-Empfehlungen

Anwendung	Empfohlenes Format
Report für Kollegen	Word (docx)
Wissenschaftliche Publikation	PDF (typst)
Online-Dokumentation	HTML
Präsentation (Meeting)	Reveal.js
Präsentation (zum Verschicken)	PDF oder PowerPoint

Das Quarto-Ökosystem

Mit dem Wissen aus diesem Tutorial kannst du:

- **Reports** erstellen (Word, PDF)
- **Präsentationen** halten (Reveal.js)
- **Websites** bauen (HTML)
- **Bücher** schreiben (Quarto Book)
- **Dashboards** entwickeln (Quarto Dashboard)

Alle verwenden dieselbe Grundsyntax — du hast jetzt das Fundament für alles Weitere.

Abschlussübung: Multi-Format-Report

1. Nimm deinen parametrisierten Pinguin-Report
2. Füge `typst` als zweites Format hinzu
3. Rendere zu Word und PDF
4. Vergleiche die Ausgaben
5. Bonus: Erstelle eine einfache 3-Folien-Präsentation zu den Ergebnissen

Weiterführende Ressourcen

- Quarto Guide — Umfassende Dokumentation
- Quarto Gallery — Beispiele aller Formate
- Awesome Quarto — Community-Ressourcen

Fazit

Du hast jetzt das Wissen, um:

- Reproduzierbare Word-Dokumente zu erstellen
- Code, Text und Ergebnisse zu kombinieren
- Professionelle Tabellen und Plots einzubinden

- Templates für einheitliches Branding zu verwenden
- Cross-Referenzen und Zitationen hinzuzufügen
- Parametrisierte Reports für verschiedene Datensätze zu erzeugen
- Dasselbe Dokument in verschiedene Formate zu exportieren

Viel Erfolg mit Quarto!

Bibliography
