Two Columns, One Namespace: Clojure to PDF

Timothy Pratley

2025-09-03

Table of contents

1 Introduction **1**

2 Making a PDF **1**

3 Why PDFs? **2**

4 Quick PDF Export: Using Your Browser **2**

5 Using the PDF format **2**

6 Prerequisites **2**

7 Example: 2-Column Journal Style via Namespace Metadata **2**

8 Showcase: Math, Style, and Substance **3**

9 Visualizing Data: Adding Charts to Your PDF **3**

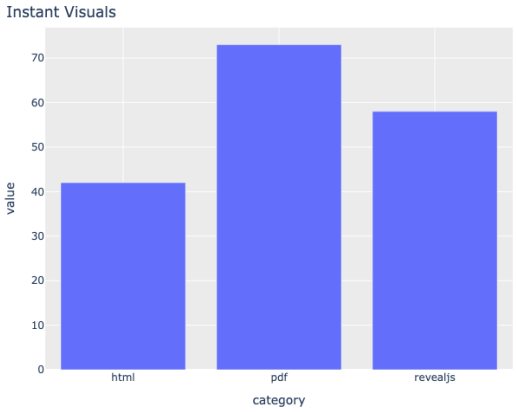
10 Conclusion **3**


```
(require '[tablecloth.api :as tc])
```

```
(require '[scicloj.tableplot.v1.plotly :as tp])
```

```
(def quick-ds
  (tc/dataset {:category ["html" "pdf" "revealjs"]
               :value    [42 73 58]}))
```

```
(-> quick-ds
    (tp/base {:=title "Instant Visuals"})
    (tp/layer-bar {:=x :category
                   :=y :value}))
```



Example math:

$$\int_0^1 x^2 dx = \frac{1}{3}$$

---

1 Introduction

Ever wanted your Clojure project to look like it just rolled off the press at a 19th-century scientific society? Or maybe you want to channel your inner Ada Lovelace or Alan Turing and make something that looks like it belongs in a library archive.

This guide shows how to create a two-column, journal-style PDF from Clojure code using Clay. You'll see how to export a PDF, add math, and include charts and code blocks.

---

2 Making a PDF

This document was created from a Clojure namespace in ClojureCivitas pdf.clj

Make sure you set the quarto metadata on your namespace to `:format [:quarto :pdf]`, or if you prefer to build at the REPL, you can set the format in the options.

```
(comment
  (require '[scicloj.clay.v2.api :as clay])
  (clay/make!
   {:source-path "scicloj/clay/pdf.clj"
    :format [:quarto :pdf]})
  :-)
```

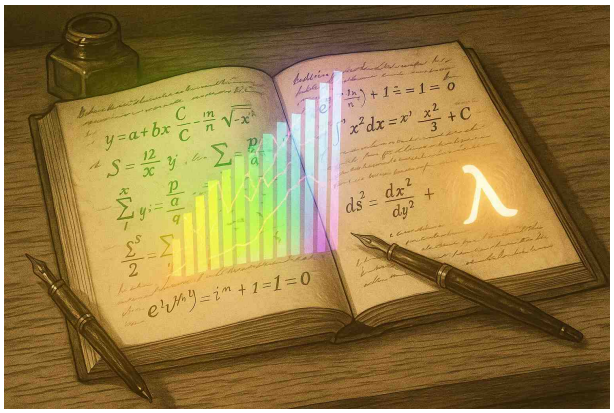


Figure 1: A vintage journal meets Clojure: classic typesetting, code, and creativity.

3 Why PDFs?

PDFs are widely used for academic publishing, grant applications, and official documentation. They look the same everywhere and are easy to share. Sometimes you want a polished, typeset result, something worthy of Ada or Turing!

4 Quick PDF Export: Using Your Browser

In most cases, the fastest and most reliable way to create a PDF is to use your browser's built-in "Print" or "Save as PDF" feature:

1. Open your Clay-generated HTML page in your web browser.
2. Press `Cmd+P` (Mac) or `Ctrl+P` (Windows/Linux) to open the print dialog.
3. Select "Save as PDF" or "Print to PDF" as the destination/printer.
4. Adjust layout, margins, and other options as needed.
5. Click "Save" to export your PDF.

This method works well for most reports, blog posts, and slideshows, and preserves the look of your HTML output.

5 Using the PDF format

But for a traditional, journal-style PDF (e.g., two columns, custom fonts, LaTeX typesetting), use `[:format :pdf]`. Some features won't work quite the same, feel free to let us know if you run into issues.

6 Prerequisites

- Install a TeX distribution (e.g., TinyTeX: `quarto install tinytex`)
- Install Quarto (https://quarto.org/docs/get-started/)
- Python and the modules plotly and kaleido

7 Example: 2-Column Journal Style via Namespace Metadata

The options for a 2-column journal style PDF are now set in the namespace metadata above. See the ^{:clay …} metadata on this namespace for a working example.

- `:documentclass "article"` is standard for journal articles.
- `:classoption ["twocolumn"]` enables two-column layout.
- `:mainfont` sets the main text font (requires XeLaTeX or LuaLaTeX).
- `:geometry` customizes page margins.

- `:toc` and `:number-sections` add a table of contents and section numbering.

8 Showcase: Math, Style, and Substance

Inline math: $E = mc^2$ (because every science article needs it!)

Display math:

$$\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}$$

Or a system of equations:

$$a^2 + b^2 = c^2$$
$$e^{i\pi} + 1 = 0$$

You can add figures, tables, and code blocks using Clojure code.

For more details and advanced options, see the Quarto PDF documentation.

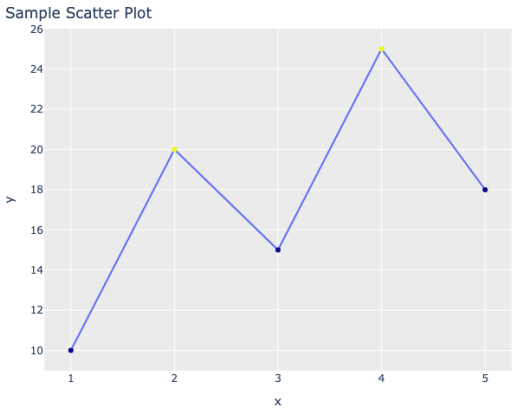
---

9 Visualizing Data: Adding Charts to Your PDF

A journal-style PDF can include charts alongside your narrative and code. Let's create a simple dataset and visualize it using Tablecloth and Tableplot:

```
(def scatter-ds
  (tc/dataset {:x [1 2 3 4 5]
               :y [10 20 15 25 18]
               :z [1 2 1 2 1]}))
```

```
(-> scatter-ds
    (tp/base {:=title "Sample Scatter Plot"})
    (tp/layer-point {:=x :x
                     :=y :y
                     :=color :z}))
```



Tableplot lets you create histograms, scatter plots, bar charts, and more. These charts will appear in your PDF just as they do in your HTML output. For more advanced visualizations, see the Tableplot and Plotly documentation.

---

10 Conclusion

For most needs, browser-based PDF export is fast and easy. For more traditional style typeset PDFs, set `[:format :pdf]` to get a PDF file.