An Introduction to Living Papers

The Living Papers Team

April 1, 2023

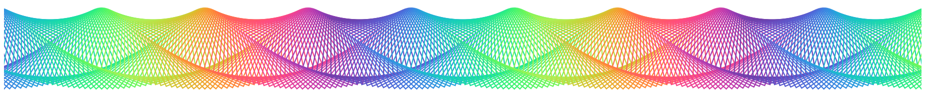


Figure 1: A teaser image for the beginning of our article.

Abstract

With Living Papers, you can author an interactive web page or standard reseach paper from Markdown source documents. This example article demonstrates the basic syntax and functionality of a Living Papers document. Compare the source markup and the resulting rendered web page!

Basic Formatting

Basic formatting includes:

- Inline text with *italics*, **bold**, or ***both***.
- Inline code `x = Math.PI * r * r`.
- Inline code with syntax highlighting jsx = Math.PI * r * r.
- Inline math $x = \pi * r^2$.
- Super$^{\text{script}}$, sub$_{\text{script}}$, and strikethrough.
- Hyperlinks and styled spans.
- Lists (like this one!)

Math & Equations

Mathematical notation is specified using TeX syntax. Inline math ($e^{i\pi}$), `math` blocks, and numbered `equation` blocks are supported. Compare a `math` block:

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

And an `equation` block:

Symbol	Value
ϕ	1.618033...
π	3.141519...
e	2.718282...

Table 1: Some irrational numbers.

$$\frac{a+b}{a} = \frac{a}{b} = \phi \tag{1}$$

Figures & Tables

Basic images and tables can be included without adornment using standard Markdown. To create numbered and captioned elements, place content within `:::`-fenced `figure` or `table` blocks.

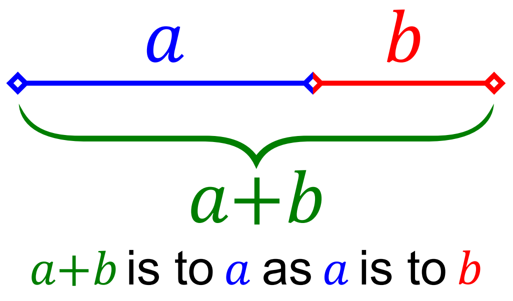


Figure 2: A depiction of the Golden Ratio (ϕ).

Citations

Give credit where credit is due! Living Papers was inspired by earlier work by Conlen & Heer [1] on the Idyll language. Living Papers supports citations references with author name(s) or with reference number only [1]. Citations are also *interactive*: click/tap a citation to view a pop-up with more information.

Citation information can be automatically retrieved using a unique ID:

- A DOI `[@doi:10.1145/3242587.3242600]` → [1]
- A Semantic Scholar ID `[@s2id:4fca64e6dc4e803d3ed904c04c6845a9e6adc53e]` → [1]

Citations can also be defined in BibTeX format, either in an external file (listed under the `references` key of the article metadata) or included anywhere in the document in a `bibliography` block:

```
~~~ bibliography
@inproceedings{conlen2021,
```

```
  title={Idyll Studio: A structured editor for authoring interactive \& data-driven articles},
  author={Conlen, Matthew and Vo, Megan and Tan, Alan and Heer, Jeffrey},
  booktitle={The 34th Annual ACM Symposium on User Interface Software and Technology},
  pages={1--12},
  year={2021}
}
~~~
```

For example, Conlen et al. [2] (`@conlen2021`) extends Idyll with a graphical structured editor to create interactive articles without writing markup code.

To create a list of citations [1, 2], separate references with semi-colons: `[@doi:10.1145/3242587.3242600; @conlen2021]`.

Cross-References and Notes

Cross-references use a syntax similar to citations:

- Figures: `@fig:teaser` → Figure 1, `@fig:goldenratio` → Figure 2
- Tables: `@tbl:irrational` → Table 1
- Equations: `@eqn:ratio` → Equation 1

By default, a descriptive prefix like "Figure" is included.[1]

Interactive Figures

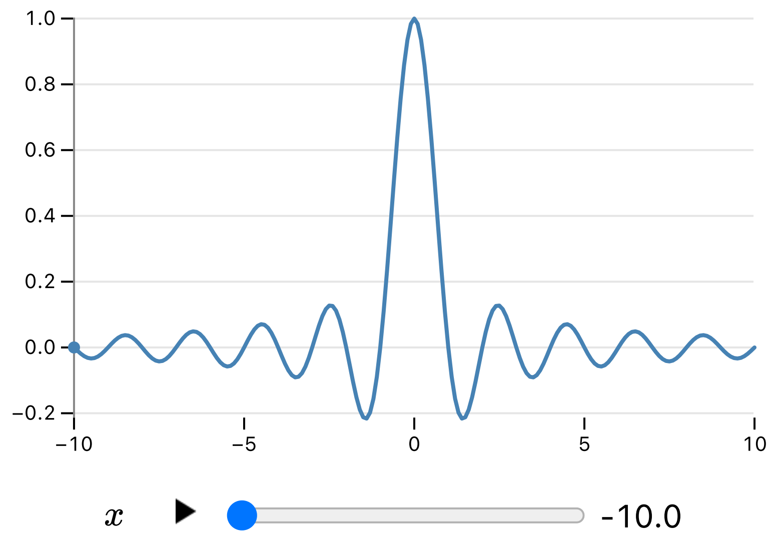
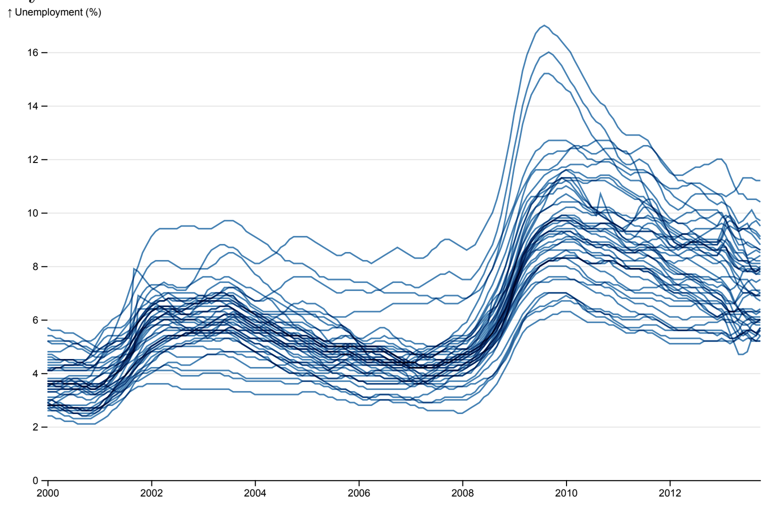


Figure 3: $sinc(x)$ at $x =$ -10.0 → $sinc(x) = 0.00$.

[1]To show a reference number only, wrap the reference like so: `[-@fig:goldenratio]` → 2.

Living Papers support *interactive* content using JavaScript code blocks and an extensible component system, all connected via a shared reactive runtime. The runtime automatically re-evaluates page content in response to interactive updates.

Living Papers uses the same JavaScript dialect as Observable notebooks. We can define variables, add input widgets, and generate figures (e.g., using Vega-Lite or Observable Plot, as in Figure 3) just as we would in a notebook. We can also directly import content from public Observable notebooks, like this D3-based line chart of unemployment rates by U.S. county:



And more...

Living Papers also supports other types of content (not demonstrated here) through custom plugins. One or more plugins can be specified under the `plugins` article metadata key. For example:

- The `knitr` plugin evaluates R code blocks (` ```r `) at compile time and includes the results in the output document (similar to RMarkdown).
- The `pyodide` plugin evaluates Python code blocks (` ```py `) *in the browser* using Pyodide. These Python blocks follow the same reactive logic as Observable JavaScript, and computed Python results are directly accessible from JavaScript.

References

[1] Matthew Conlen and Jeffrey Heer. Idyll. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. ACM, oct 11 2018.

[2] Matthew Conlen, Megan Vo, Alan Tan, and Jeffrey Heer. Idyll Studio: A structured editor for authoring interactive data-driven articles. In

The 34th Annual ACM Symposium on User Interface Software and Technology, pages 1–12, 2021.