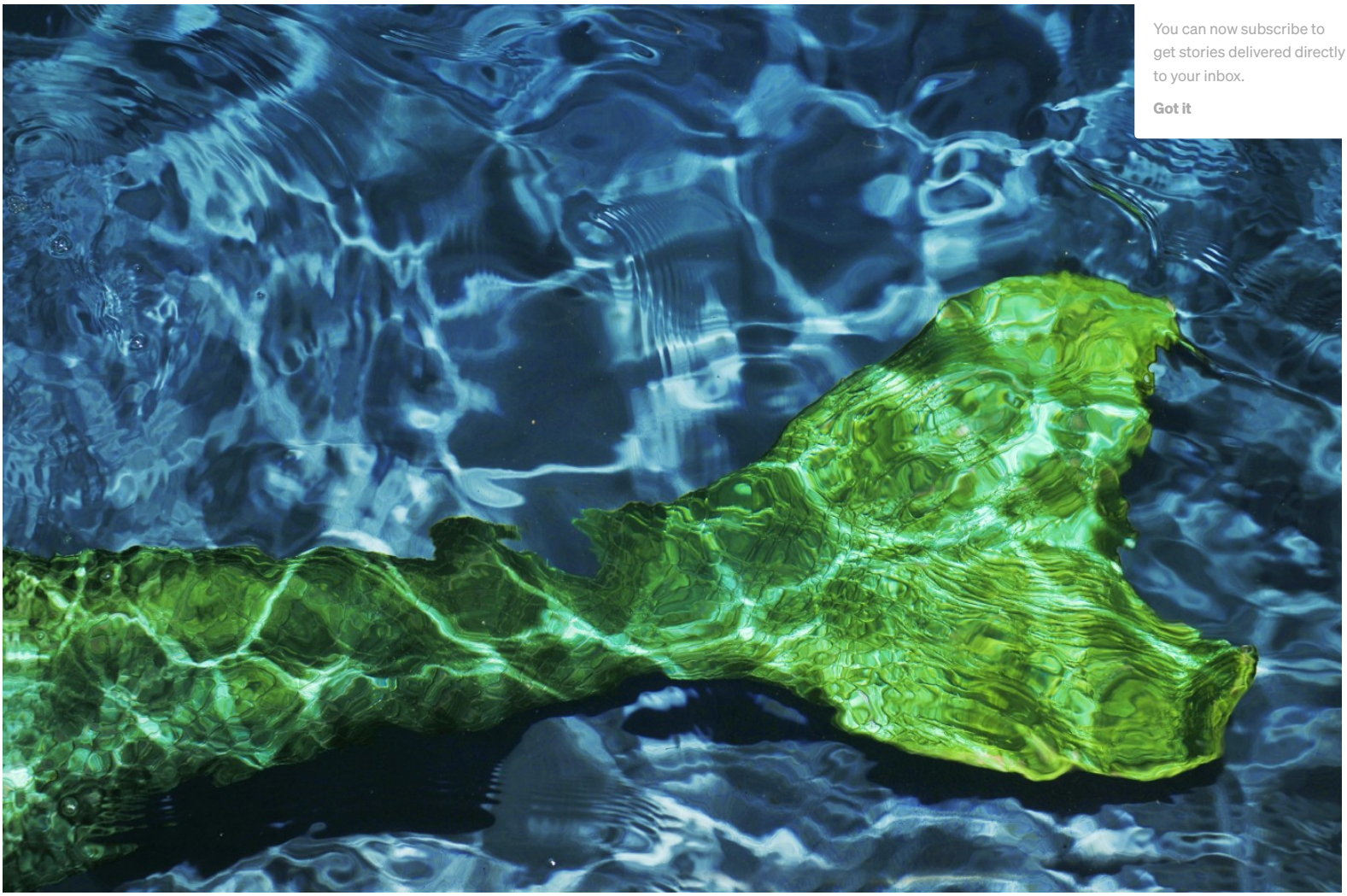
Published in Towards Data Science

Alexandra Souly Follow
May 31, 2021 · 8 min read ★ · Listen

Mermaid: Create diagrams quickly and effortlessly

If you've ever tried to explain anything complicated — be it an algorithm, a code base structure or a project plan — you probably know how useful good diagrams are.



How would you *not* want to use a tool called Mermaid 🧜? Photo by Annette Batista Day

However, with lots of us working remotely, it is more difficult to share spontaneous drawings when explanations fall short. While it was easy to have a whiteboard session in the office, this is a lot more awkward online. Trying to draw wonky boxes with your mouse in a drawing app, or dragging clunky arrows, all while your co-workers are watching your shared screen in a call is not fun, to say the least.

But there is a way to generate all your diagrams quickly and painlessly with code only, so *you'll never have to draw a box again*!

Mermaid is a tool that lets you create complicated diagrams in Markdown — it works with simple commands and an intuitive syntax. You don't have

Mermaid arranges the diagram sensibly for you, so you won't have to worry about manually rearranging all your boxes if you forgot a middle step in a flowchart.

In this article, I will tell you about the different ways to use Mermaid, and show you some examples of what it's capable of. It is certainly my favourite tool to create diagrams, I hope it becomes yours too!

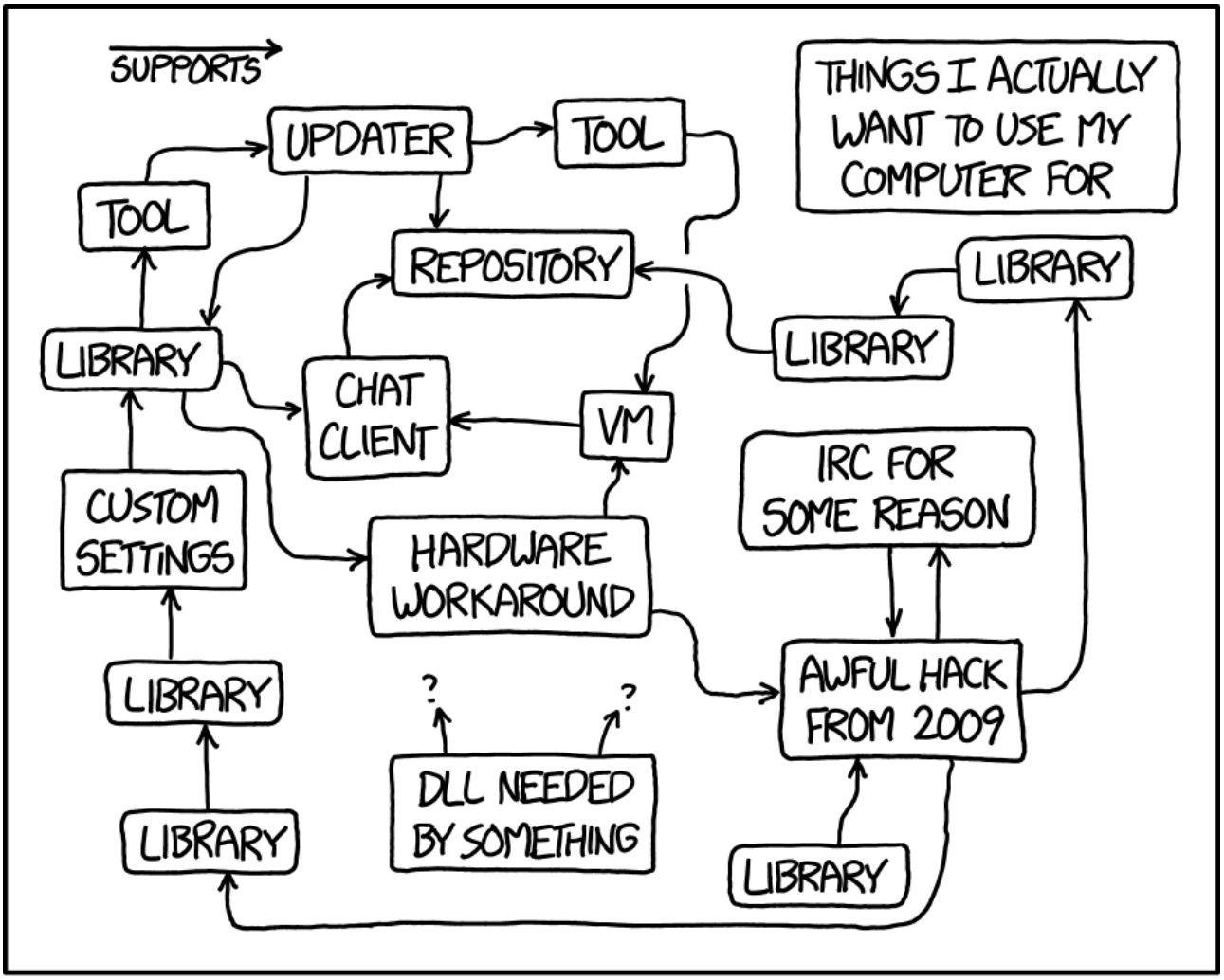
But why do we need diagrams?

Human brains are much better at understanding images and patterns than text. We process a complicated structure faster if we see its visual representation, and it turns out we also **remember it better** — this is called the picture superiority effect. This is also intuitive — you can draw conclusions form a bar or pie chart much faster than reading a list of numbers.

Pictures also **grab the audience's attention**. If you give a lengthy, complicated explanation without a visual aid, most of your listeners will have zoned out after a minute, or already forgot the beginning. If you include a diagram, they can find their way back to your explanation.

They are also helpful to draw for yourself if you just want to understand something that requires more information than what your working memory can deal with. Drawing the high level code base architecture, the information flow in a components or a lower level OOP class designs can help you **spot design errors upfront**, when it's the cheapest to correct.

A very underrated use for diagrams is putting one in your **GitHub repository** README file, **displaying the code architecture**. Particularly for open source repos lots of people contribute to, it allows newcomers to familiarise themselves with the codebase quickly, instead of looking through what's in each directory. The same applies to new starters in your company — they will really appreciate having a diagram for every repository.



code based on a **Markdown-inspired syntax**. This makes the diagrams very easily **maintainable**. You will always have the code that generated the diagram, so you won't end up with a beautiful, but outdated image file you can't modify anymore.

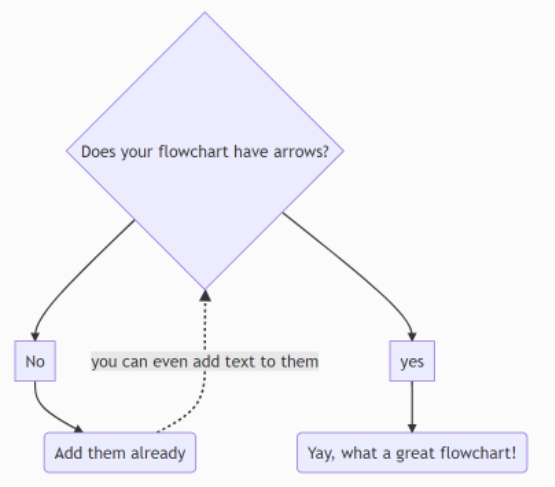
This also means that you can easily see the changes to a diagram, unlike with any other image-based method. If you keep your diagram under **Git version control** (which you can easily, as it's stored as text), a simple `git diff` will show you **what changed in a human-readable way**.

I've already mentioned that Mermaid **rearranges the boxes for you** if you decide to add more stuff to your diagram. This makes it possible to generate diagrams on the fly in a video call, much faster than hand-drawing them on a laptop. I have found that if you spend about half an hour on learning the syntax, you can completely replace any other graph drawing tool you use. Personally, I find drawing with a mouse really difficult, so for me Mermaid is superior to using Paint or OneNote, and with some practice it is much faster to use than draw.io for example.

What can Mermaid do?

Let's look at some examples Mermaid can render for us, and what the code that generates it looks like:

```
graph TD
    A{Does your flowchart have arrows?} --> B[No]
    A --> C[yes]
    B --> D(Add them already)
    C --> E(Yay, what a great flowchart!)
    D -.->|you can even add text to them| A
```



A flowchart in 6 lines of code

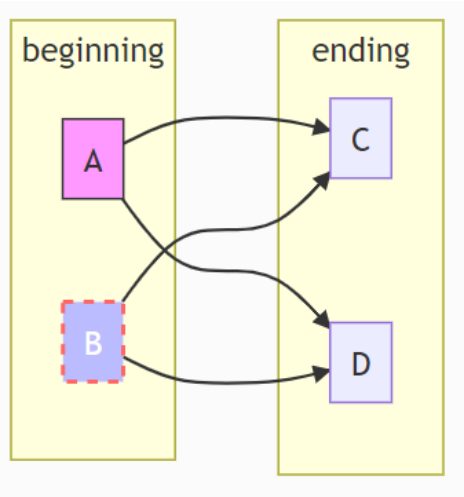
The start `graph TD` indicates the **orientation of the graph**: top-down as opposed to LR(left-right), RL(right-left) or BT(bottom-top).
You can specify nodes by a short identifier (A, B, C here) and indicate what shape and text it should have with the brackets following it. You can specify many shapes, including circle, rhombus or trapezoid. To specify the arrows from one box to another, you just type the two node IDs joined by an arrow, depending on what arrow type you want. You can add text to the arrows with the `|<text>|` before the destination node.

There is also a shorthand to express more complicated dependencies to save time typing out all the arrows, using `&` to bundle nodes together. To create subgraphs, you can just type `subgraph <name>` and `<end>` once you have finished declaring what's inside. Mermaid uses CSS style for customisation, you can change the style of your nodes if you want. Although that's a bit harder to just memorise, it should be okay if you are familiar with CSS :

```
graph LR

A & B--> C & D
style A fill:#f9f,stroke:#333,stroke-width:px
style B fill:#bbf,stroke:#f66,stroke-width:2px,color:#fff,stroke-dasharray: 5 5

subgraph beginning
A & B
```



Subgraphs, funky arrows and colours

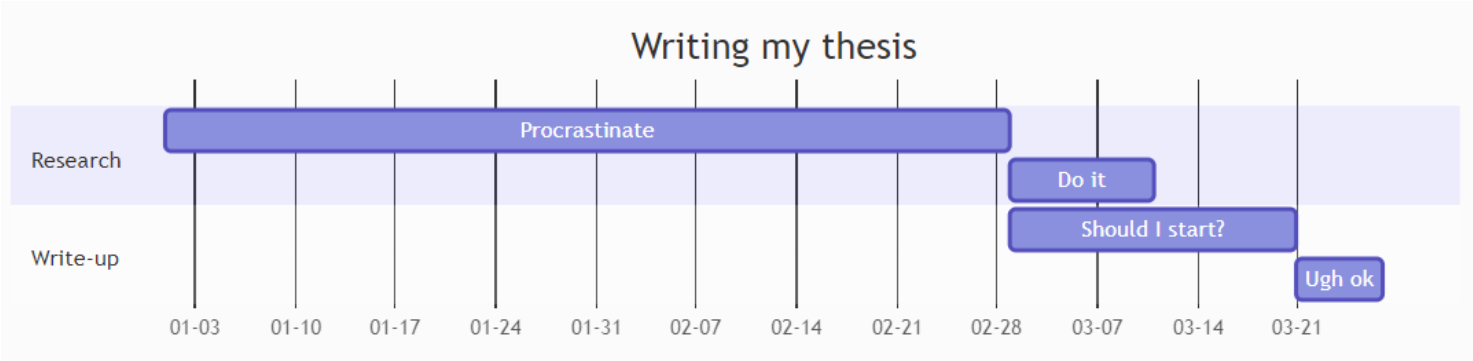
Another example, this one of a Gantt chart, commonly used for project management:

```
gantt
title Writing my thesis

dateFormat  MM-DD
axisFormat  %m-%d

section Research
Procrastinate           :a1, 01-01, 59d
Do it     :after a1  , 10d

section Write-up
Should I start?     :03-01 , 20d
Ugh ok      : 6d
```



A Gantt chart always helps project management

We had to specify the date formats, what our sections are, what the tasks are, when they start and how long they last. Straightforward, and no boilerplate statements needed.

To learn more about the syntax and different options like pie charts and status diagrams, check out the Mermaid syntax guide here.

How can I use Mermaid?

You can use Mermaid in lots of different ways:

- Online live editor

- Code editors
- Browser extension
- Command line

The full list of integrations can be found here.

Online Live Editor

This is the simplest way to try out Mermaid, using their online tool. You can start with a template, and verify how your changes affect the final result. You can save your image directly, or just create a link to it. As a bonus, it also saves your recent changes with cookies, so you don't have to worry about continuing the next day. All the examples in this article were made with the live editor.

GitLab / GitHub

GitLab does Mermaid support by default in its Markdown files. You can just add your diagrams with the following syntax, and it will magically work:

```
```mermaid
graph TD;
A →B;
A →C;
B →D;
C →D;
```
```

If you are however a GitHub user, I will have to disappoint you — they have not officially integrated with Mermaid so far. There is a project to compile your Mermaid graph as a GitHub action, so you can still insert the image into a README file without doing anything and let the pipeline take care of it.

Websites

- If you have a **website** with Yarn as your package manager, adding Mermaid as a dependency is simple:

```
yarn add mermaid
```

- If you **don't have a bundler**, use the following:

```
<script src="https://cdn.jsdelivr.net/npm/mermaid/dist/mermaid.min.js"></script>
<script>mermaid.initialize({startOnLoad:true});</script>
```

This will make the Mermaid parser look for <div> tags with class Mermaid, and render them into diagrams. So you would use it as

```
<div class="mermaid">
graph TD;
A-->B;
A-->C;
B-->D;
C-->D;
</div>
```

- If you use **GitHub pages**, just add the following to your ***'header.html',*** or download the file and source from the relative path:

```
<script src="https://cdn.jsdelivr.net/npm/mermaid/dist/mermaid.min.js">
```

```
graph TD;
A-->B;
A-->C;
B-->D;
C-->D;
</div>
```

- Alternatively, if you have a **Jekyll site** that is not on GitHub pages, you can use the jekyll-mermaid plugin.
- If you use **WordPress**, there is a plugin for you too.

Code Editors

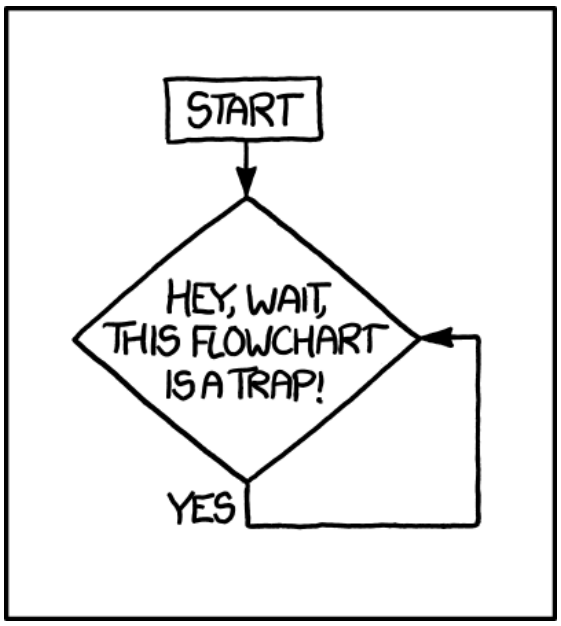
May editors support Mermaid, including Atom, VIM and VS Code. I personally use VS Code, and the extension Markdown Preview Mermaid Support is very helpful in rendering the graphs in the markdown previews.

Browser extensions

The browser extension processes all the Markdown syntax found in a Markdown document, which means you can still locally view your graphs even if for example GitHub won't render them to you. This extension works both on Chrome and Firefox. Of course, if you want other people to also see the diagram, they also have to download the extension.

Command line

Mermaid-CLI exists so that you can directly create diagrams out of input files using the CLI. It works from within a Docker container, you can see how to use it here.



Comic by xkcd

🤳 Now go and have a play with it 🤳

Mermaid is such a fun tool to use once you get used to it, and it can save so much time and effort when trying to create diagrams without paper. I hope you found this short guide useful and it can save some time for your next diagram!

You can find more about Mermaid here:

https://mermaid-js.github.io/

Related articles:

- Blog — Use Mermaid syntax to create diagrams
- Mermaid — Create Charts and Diagrams With Markdown-like Syntax | by Trevor Lasn | Better Programming

- Diagrams as Code: Mermaid (freshbrewed.science)