Andy Haskell
Posted on Nov 14, 2023

Write your tech talk slides rapidly with Marp

#devrel #career #codenewbie #publicspeaking

I've been doing public speaking in tech since 2015 when I presented my very first lightning talk at Boston Go, and since then, I've given talks at meetups, at work, and at a couple conferences.

But while I love speaking at events to teach tech concepts I've learned, there can be some friction getting the ideas from my brain to my slides, with thinking through slide layouts and polishing the slides so my talk flows well from slide to slide.

What's helped me with writing slides lately though, is a tool called Marp, which allows me to write my slide layout by typing just in my text editor **in Markdown**. Which for me means:

- 🏎️ Because Marp has sensible built-in layouts, I don't need to think about things like font size until I have a reason to; I can just type like I would for a blog post, and the slide will look good enough that I can practice with it right away, and tweak its layout later!
- 👀 Since I'm writing slides in my editor, if I want to switch to experimenting with the code that my talk is covering, that's much less of a context-switch for me than going between a drag-and-drop slide tool and a text editor.
  - Speaking of preventing context switching, there's also a Marp plugin for VSCode that you can use to see your slides side by side with the text you're writing, in real time!
- 🦕 Since my slides are made of text, that makes it easy to have different versions of my talk **versioned in Git**, so as I experiment with different arrangements of slides, I can switch versions as fast as you can say `git checkout`!
- 🎨 You can tweak your layout either for individual slides, or all slides, using CSS!

And I can say that Marp is conference-ready, since it's what I used for writing and iterating on my slide deck at GopherCon this year!

So I'd like to show you in this tutorial how to use Marp to write three types of slides: A title slide with an eye-catching image, an "about me" slide with an image on the left and text on the right, and a code snippet slide. I'll also show you how to add speaker notes that only the speaker can see while presenting, and how to use CSS to give your slide layout more customization.

Installing Marp

For getting Marp, you install its command-line interface with a package manager, like Homebrew, Scoop, or Yarn.

```
yarn global add marp-cli
```

With Marp installed, let's write our slide deck!

Writing our first title slide

First, let's give our talk a title slide. Paste this Markdown into a file named `talk.md`.

Andy Haskell

Software engineer at Salesforce (prev MIT), Google Developer Expert in Go, organizer at Boston Golang, resident #sloth enthusiast at everywhere

LOCATION
Boston

WORK
Software Engineer at Salesforce

JOINED
Feb 27, 2019

More from Andy Haskell

Conferencing tips as an autistic + ADHD conference-goer
#neurodiversity #codenewbie #career #devrel

How AuDHD traits have helped me get good at devrel
#neurodiversity #devrel #codenewbie #career

Using Testify Mock in web clients
#go #testing #webdev #codenewbie

```
---
marp: true
theme: uncover
paginate: true
---

# What is io.Reader in Go?

YOUR NAME - @YOUR_SOCIAL_MEDIA
```

We've got two pieces of Marp syntax written now: At the top, the section full of options inside the triple dashes is called the frontmatter, and it contains configuration for your presentation as a whole.

The rules in our frontmatter are:

- `marp: true`, which indicates that this is a Marp presentation
- `theme: uncover`, which sets your slides' theme to Uncover, one of the built-in themes from Marp.
- `paginate: true`, which causes your slides to include a page number.

There's other rules you can add in your frontmatter as well, such as `style` for additional custom CSS styling, or `backgroundImage` if you want your slides to all have a default background image, such as a patterned background to make your slides look more flashy.

Below the frontmatter, we have our first slide written in Markdown.

```
# What is io.Reader in Go?

**YOUR NAME - @YOUR_SOCIAL_MEDIA**
```

Just like in standard markdown, the text represents a header for the title of your talk, and regular text in bold for your name and social media account.

To see what this slide actually looks like in a presentation, run the Marp CLI we just installed:

```
marp -w talk.md
```

You will then get a prompt to go to a localhost URI. Navigate to there in your browser and you'll
see your slide deck. So far, the title slide looks like this:

We've got our header and name. This is a nice clean layout for a title slide, but what if we wanted a cool image to kick off the talk with? You can add that with some Markdown for an image.

Adding an image

Similar to in standard Markdown, in Marp you use the `![alt text](image URL)` format for rendering images. To try this out, we'll use a photo I got of the bright blue waters on a beautiful day at the beach.

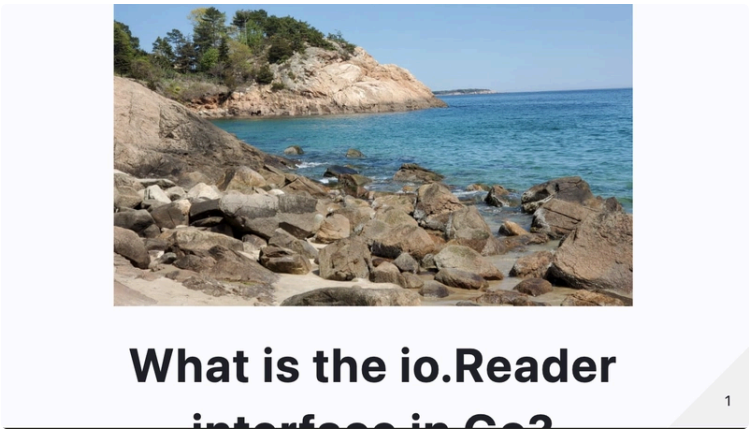
```
![Picture of the view at the beach; rocks are in the foreground and the w

# What is io.Reader in Go?

YOUR NAME - @YOUR_SOCIAL_MEDIA

---
```

Now look at talk.html, and



Woah that's a tall image! If we want this image to fit on the slide, we're going to need to resize it, and we would do so by adding a directive to the alt text, and also making the header's text smaller by converting it from an h1 header to an h3 header.

```
![w:800 h:480 Picture of the view at the beach; rocks are in the foregrou

### What is io.Reader in Go?
```

With the directives `w:800 h:480`, we're having Marp set the width and height of the image to 800 pixels and 480 pixels, respectively, fitting neatly in the slide.

Or if you really want to be flashy and make an image that takes up the whole background, you can use the `bg` directive!

```
![bg Picture of the view at the beach; rocks are in the foreground and the water that day was aqua](https://url/to/beach/image)
```

But you'll want to ensure the text contrasts with the colors on the image that it might overlap with. We'll look at one way to do that using drop-shadow text in the section on CSS.

Adding an "about me" slide

To write a new slide, you separate each slide in Marp by adding a line containing three dashes, like this:

```
![w:700 h:500 Picture of the view at the beach; rocks are in the foregrou

# What is io.Reader in Go?

YOUR NAME - @YOUR_SOCIAL_MEDIA

---

Everything from here down to
the next --- line will be
will be part of the Markdown of
the second slide of your talk,
and then you add another --- to
give your talk a third slide
```

We'll replace the placeholder block of text with an "about me" slide that follows a common layout for slides: An image taking up the left side of the slide, and text on the right.

And we'll be doing this using a parameter of the image `bg` directive we saw earlier, which makes the `bg` image only be the left side of the slide: `bg left`. For this example, let's have my dog Lola give this talk.

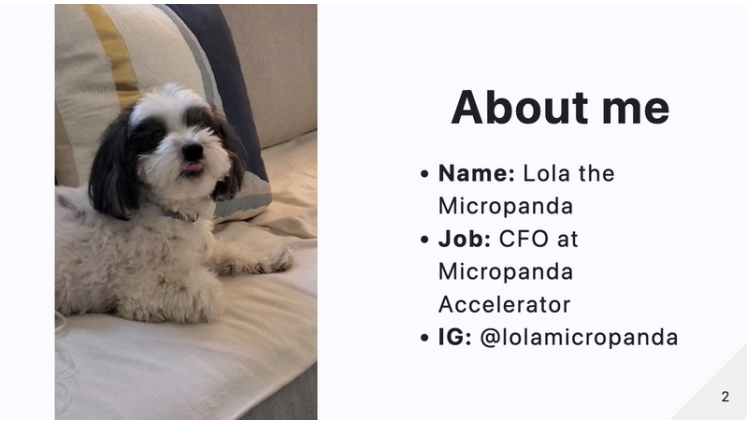
```
---

![bg left 70% My dog Lola, an adorable black and white Havanese dog who l

# About me

- **Name:** Lola the Micropanda
- **Job:** CFO at Micropanda Accelerator
- **IG:** @lolamicropanda
```

Open up your Marp slide deck and you'll see:



Lola's "about me" slide! Notice in addition to the `left` parameter on the `bg` directive, I also decided to add a `70%` parameter. This makes it so the image takes up the middle 70% of the left of the slide, rather than the whole left side of the slide, making the "about me" image appear as a stripe across the left of the slide.

By the way, you know the technique of presenting a bulleted list where you display the bullet points one by one as you present? Like "A bit about me [show bullet point], I'm Lola the Micropanda, [show another bullet point], I'm Chief Floof Officer of Micropanda Accelerator [show one more bullet point], and you can follow me on Instagram at @lolamicropanda"?

Marp supports that as well! If you change the bullet points from dashes to asterisks, like this

```
* **Name:** Lola the Micropanda
* **Job:** CFO at Micropanda Accelerator
* **IG:** @lolamicropanda
```

You get a fragmented list; go through the "about me" slide in `talk.html` and now when you present, your bullet points appear one by one!

Code snippets and speaker notes

Just like how Marp supports images, headers, regular text, and lists, you're also able to have slides in Marp that contain code snippets. Just like in standard Markdown, you make a code snippet with the triple backticks (replaced with a message in brackets for the code snippets ahead because I was having trouble rendering triple backticks inside a code snippet).

```
## io.Reader in Go: small interface with tons of power

[replace this box with triple backticks in your code]
type Reader interface {
    Read(p []byte) (n int, err error)
}
[replace this box with triple backticks in your code]

---
```

If you look at this slide in your `talk.html`, you'll see:

Your code on a slide, formatted to monospace! Though you might want syntax highlighting on your slides, and if you do, then there's a simple tweak you can do: before the leading backticks, type the programming language your code is in. My code is in Go, so I am leading with `go` at the end of the triple backticks.

```
- [replace this box with triple backticks in your code]
+ [replace this box with triple backticks in your code]go
  type Reader interface {
      Read(p []byte) (n int, err error)
  }
```

Now your slide will have syntax highlighting.

And if I was giving this talk for real, this would be a slide where speaker notes would really come in handy, since there's several things I'd want to point out about Go's `io.Reader` interface.

Not every speaker uses notes, but if you use speaker notes or want to try them, Marp lets you give any of your slides some notes that only you'll see when you

present using HTML-style `<!-- -->` comments. I personally find them useful for the flow of my talk since it gives me reminders of concepts I want to bring up, or as timing cues such as to go to the next slide mid-sentence. Here's an example

```
<!--
The io.Reader interface is a type you'll see all over the Go interface, w

* In the return value n is the number of bytes read to the buffer
* err can include EOF for reaching the end of the input
* The part I found surprising was we don't return the bytes we read, we c
-->

---
```

To use your speaker notes, open talk.html, mouse over the slides, and click the lectern icon at the right of the menu that opens. Now you're in **presenter view** just like in drag-and-drop tools. So you'll have one desktop window showing the current slide, which you share to the audience, and the other showing your speaker notes, the current slide, and a thumbnail of which slide is next.

Using CSS in your slides

The layout for Marp slides does have sensible defaults, but you are able to customize your slide's layout with the full power of CSS, by inserting a `<style scoped>` HTML tag into a single slide, or a `<style>` HTML tag to add CSS rules that apply to all slides.

First up, on our slide showing the code to Go's `io.Reader` interface, while I like having a small amount of information on each slide in order to reduce cogntive load, with so little code on the slide, the code could use a bigger font size to take up more of the screen. Here's how we'd do that:

```
---

<style scoped>
code {
  font-size: 40px;
}
</style>

## io.Reader in Go: small interface with tons of power

[replace this line with triple backticks in your code]go
type Reader interface {
    Read(p []byte) (n int, err error)
}
[replace this line with triple backticks in your code]

---
```

Look at this slide in `talk.html`, and you'll see the code now takes up more of the screen.

Next up, remember how I mentioned we can have our title slide be a background image with text on it using an image's `bg` direcctive, but we'd want to use CSS to ensure our text is legible?

Here's the CSS we'd add that for a title slide. Notice we're also changing the header back to an h1 since the image is now a background image so we don't need to conserve space with an h3 header.

```
<style scoped>
h1, p {
  color: #FFFFFF;
  font-weight: bold;
  text-shadow:
    5px 5px 3px #000000;
}
</style>

![bg Picture of the rocks on Singing Beach; rocks are in the foreground a

# What is the io.Reader interface in Go?

YOUR NAME - @YOUR-SOCIAL-MEDIA

---
```

The rules make all the text on that slide (a "heading 1" and a paragraph, or `h1` and `p` in CSS and HTML) white, bold-faced, and with black text-shadow down and to the right. Here's how that looks in `talk.html`

Awesome! As one last thing to look at, let's say we want to ensure the content of your slides doesn't cross into the bottom 20% of the slide. This is a scenario you

might see if a conference has live captioning and wants to place the live captions in the bottom 20% of the screen, which GopherCon did.

When I found that out, I was able to quickly throw in `<style scoped>` tags to get all my slides taking up 80% of the screen, but what could have gotten that done faster, was if I had instead used a `<style>` tag without the `scoped` to apply 20% padding to the bottom of all slides!

```
<style>
section {
  padding-bottom: 20%;
}
</style>
```

This works because in the HTML that the Marp tool produces, an individual slide is a `<section>` HTML element.

This does though raise up the content of all existing slides by 20%, so if you apply this to your slide layout after you've already gotten some slides written, you'll still want to look at your existing slides and make sure you don't need to further tweak any individual slides' layouts.

One other thing, if your page has too much content on it or too big an image, it's still possible for a slide to go past a height cutoff, so one trick I learned at the GopherCon rehearsal (can't remember who told me that but whoever that was thank you!) was to temporarily change the background image of my slides to a 5 x 5 grid, with a red horizontal line at 80% height.

You can apply that background image to all slides (except slides with their own background image, like our newly upgraded title slide) using `backgroundImage` in the frontmatter.

```
---
marp: true
theme: uncover
paginate: true
backgroundImage: url('https://url/to/cutoff/image.png')
---
```

With that background image, this is what the code slide looks like, well above the 80% mark!

io.Reader in Go: small interface with tons of power

```
type Reader interface {
    Read(p []byte) (n int, err error)
}
```

3

If you were interested in Marp, hopefully this helps get you started writing slides in Marp! I'd also be interested to hear about what other tools you've used for making the process of writing a tech talk more ergonomic, so if you've used any other tools, I'd love to hear about them in the comments!

Top comments (1)

Michael Tharrington • Nov 14 '23

Nice! Marp sounds like a really helpful tool... appreciate you writing this up, Andy.

Code of Conduct • Report abuse