

Static site generators *e.g.* Jekyll

Gaspard Jankowiak

April 2018

`gaspard.janko.fr/en/misc`

WHAT WE WILL DO

What does "having a website" mean?

An example of static site generator: Jekyll

- Source structure

- The configuration file

- The front matter

- Formatted content: Markdown

- Integrating data and content into `.html`: Liquid

- Running Jekyll

Hands on example

OVERVIEW

What does "having a website" mean?

An example of static site generator: Jekyll


Hands on example

WHAT DOES "HAVING A WEBSITE" MEAN?

Content

```

Charlie Chaplin
16.04.1889



charliechaplin.com

Actor, director, composer,
screenwriter, producer, editor

"I was hardly aware of a crisis
because we lived in a continual
crisis; and, being a boy, I
dismissed our troubles with
gracious forgetfulness."

Films:
The Kid, 1921
A Woman of Paris, 1923
The Gold Rush, 1925
The Circus, 1928
City Lights, 1931
Modern Times, 1936
The Great Dictator, 1940
...

```

+ Structure

```

Charlie Chaplin
16.04.1889



charliechaplin.com

Occupation:
· Actor
· Director
· Composer
· Screenwriter
· Producer
· Editor

I was hardly aware of a crisis because
we lived in a continual crisis; and, being
a boy, I dismissed our troubles with
gracious forgetfulness.

Films:

```

Title	Year
The Kid	1921
A Woman of Paris	1923
The Gold Rush	1925
The Circus	1928
City Lights	1931
Modern Times	1936
The Great Dictator	1940


+ Layout and styling

```

Charlie Chaplin
16.04.1889

[Homepage] →
charliechaplin.com

Occupation:
· Actor
· Director
· Composer
· Screenwriter
· Producer
· Editor



I was hardly aware of a crisis because
we lived in a continual crisis; and,
being a boy, I dismissed our troubles
with gracious forgetfulness.

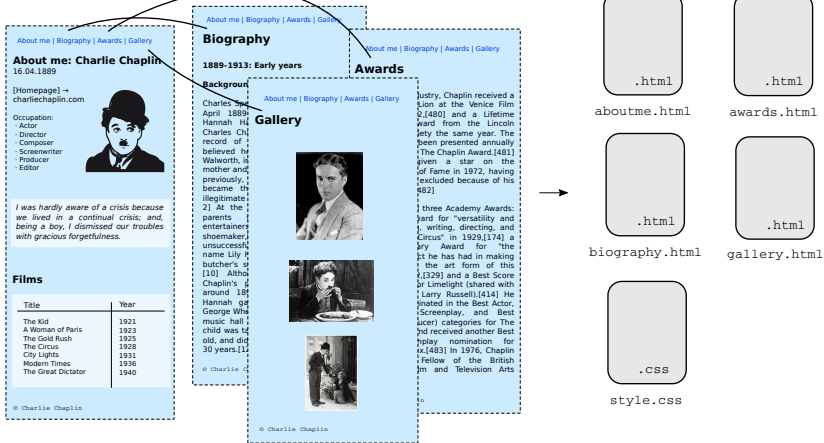
Films:

```

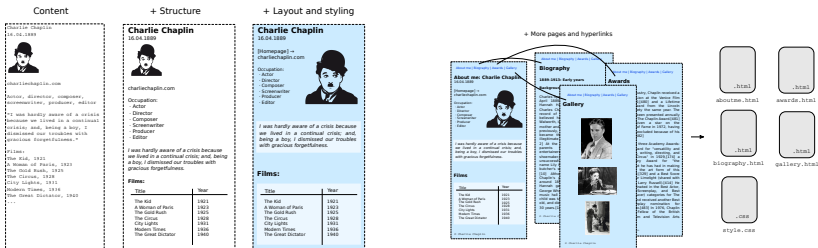
Title	Year
The Kid	1921
A Woman of Paris	1923
The Gold Rush	1925
The Circus	1928
City Lights	1931
Modern Times	1936
The Great Dictator	1940

WHAT DOES "HAVING A WEBSITE" MEAN?

+ More pages and hyperlinks



WHAT DOES "HAVING A WEBSITE" MEAN?



Once the `.html`/`.css` source is ready, it can be served to visitors.

WORKING DIRECTLY WITH .HTML IS PAINFUL

- ▶ Hard to read, tags clutter the content
- ▶ Structure not obvious. `<table>`, `<th>`, `<tr>`, `<td>` anyone ?
- ▶ Redundancy as soon as `{pages} > 1`: *simple* changes to common parts require editing *all* files. For example: layout, styling and navigation.

Pages can be drawn directly (Click-o-dromes like Google Sites), but lack of control, trap you, and cannot host on UniVie servers.

Pages can be generated by the server at each request (dynamic setup), but complex + dangerous (database, one more language) and trap you (less), and cannot host at UniVie.

When user interaction is not needed, we can generate pages *statically* using simple text files.

⇒ Static site generators

WORKING DIRECTLY WITH .HTML IS PAINFUL

- ▶ Hard to read, tags clutter the content
- ▶ Structure not obvious. `<table>`, `<th>`, `<tr>`, `<td>` anyone ?
- ▶ Redundancy as soon as $\#\{\text{pages}\} > 1$: *simple* changes to common parts require editing *all* files. For example: layout, styling and navigation.

Pages can be drawn directly (Click-o-dromes like Google Sites), but lack of control, trap you, and cannot host on UniVie servers.

Pages can be generated by the server at each request (dynamic setup), but complex + dangerous (database, one more language) and trap you (less), and cannot host at UniVie.

When user interaction is not needed, we can generate pages *statically* using simple text files.

⇒ Static site generators

STATIC SITE GENERATORS

- ▶ Content is described using formatted text (~Wiki)

```
# Charlie Chaplin
```

```
16.04.1889
```

```
![portrait] (photo.jpg)
```

```
[Homepage] (http://charliechaplin.com)
```

- ▶ Layout is described using templating

```
<!doctype html>
```

```
  <head><title>{{ page.title }}</title></head>
```

```
  <body><div id="main">
```

```
    {{ content }}
```

```
  </div></body>
```

```
</html>
```

- ▶ A small program compiles everything into .html files, ready to be uploaded.

STATIC SITE GENERATORS

.HTML RESULT (JEKYL)

```
<!doctype html>
<html>
  <head><title></title></head>
  <body><div id="main">
    <h1 id="charlie-chaplin">Charlie Chaplin</h1>
<p>16.04.1889

<a href="http://charliechaplin.com">Homepage</a></p>
  </div></body>
</body>
<html>
```

PICK YOURS: STATICGEN.COM

StaticGen

Top Open-Source Static Site Generators

Need a CMS to go with a generator below? Check out the brand new headlessCMS.org for your top options.

[SHARE](#)

All languages ▾
Sorted by stars ▾
About StaticGen
The Rules
Need a Static CMS?

Jekyll

jekyllrb.com

★	👤	📄
34037	7506	148
0	0	0

A simple, blog-aware, static site generator.

Language: Ruby
Templates: Liquid
License: MIT

[Deploy to netlify](#)

Hugo

gohugo.io/

★	👤	📄
25055	3074	234
0	0	0

A Fast and Flexible Static Site Generator.

Language: Go
Templates: Go Templates
License: APL 2.0

[Deploy to netlify](#)

Next

learnnextjs.com/

★	👤	📄
24508	2310	227
0	0	0

A framework for statically-exported React apps

Language: JavaScript
Templates: JavaScript
License: MIT

Get started with 1 click

Introducing the "Deploy to Netlify" button on select generators to help you deploy new sites from generator-specific templates with a single click. Get a site up in minutes, with custom domain, HTTPS and continuous delivery completely free of charge.

Want your own Deploy to Netlify button? [Learn more here](#)

Hexo

Gatsby

GitBook

Nuxt

OVERVIEW

What does "having a website" mean?

An example of static site generator: Jekyll

- Source structure

- The configuration file

- The front matter

- Formatted content: Markdown

- Integrating data and content into `.html`: Liquid

- Running Jekyll

Hands on example



SOURCE STRUCTURE

- ▶ `_config.yaml` configuration file
- ▶ `_layouts/` directory of `.html` templates
- ▶ `_site/` output directory (can be changed)
- `_includes/` reusable bits of templates
- `_data/` structured data, like list of publications
- `_posts/` directory for pages with a fixed publication date (posts)
- `_sass/` directory for css on steroids

Files beginning with `_` will be used by Jekyll to build the site.

Files with a *front-matter* will be processed into `.html`.

All others are simply copied to `_site`



THE CONFIGURATION FILE

Used to defined configuration and site-wide variables

- ▶ The configuration file can be empty
- ▶ Variable `foo` defined in the front-matter are accessible with `site.foo`.

Most important options:

- ▶ `baseurl`: at UniVie something like
`baseurl: "/~jankowiak"`
- ▶ `markdown`: which Markdown processor to use (kramdown is good)

THE FRONT MATTER

aboutme.md:

```
layout: default
title: "About me"
---
```

```
# Charlie Chaplin
16.04.1889
![alt text](photo.jpg)
[Homepage](http://charliechaplin.com)
```

Variable `foo` defined in the front-matter are accessible with `page.foo`.

FORMATTED CONTENT: MARKDOWN

- ▶ Very similar to the Wiki language
- ▶ Block of text are wrapped into `<p>` tags
- ▶ Headings:
 - ▶ `# Title` → `<h1>Title</h1>`
 - ▶ `## Subtitle` → `<h2>Subtitle</h2>`, etc.
- ▶ Links: `[link text](http://link.to/someplace)`
- ▶ Images: `![alt text](url/image.jpg)`
- ▶ Lists:

```
* item foo
* item bar
```

→ `...`

```
1. item one
2. item two
```

→ `...`

- ▶ Also easy to do tables, code blocks, blockquotes, etc... :
`daringfireball.net/projects/markdown/syntax`
- ▶ Can include html tags if needed

TEMPLATES: LIQUID

- ▶ Idea: put variables into html, like php
- ▶ Uses a language based on Ruby (`if` finishes with `endif`, `for`, `while` finish with `end`)
- ▶ Allows to use variables from the page (`page.foo`) and from the config (`site.foo`)
- ▶ Code is between `{% ... %}` markers (no output in the `html`)
- ▶ Printing is between `{{ ... }}` markers

```
{% if user %}
  Hello {{ user.name }}!
{% endif %}
```
- ▶ The `{{ content }}` tag will render the Markdown of the current page
- ▶ Many fun "filters" `{{ "title" | capitalize }}` → Title
`{{ "/style.css" | relative_url }}` → /my-baseurl/style.css

WORKING WITH TEMPLATES AND DATA

EXAMPLE: NAVIGATION

[About me](#) | [Biography](#) | [Awards](#) | [Gallery](#)

About me: Charlie Chaplin
16.04.1889

Figure: Remember this?

```
data/navigation.yaml: layouts/default.html:
- text: "About me"      ...
  url: /aboutme        {% for link in site.data.navigation %}
- text: "Biography"    <a href="{{ link.url }}">
  url: /biography      {{ link.text }}</a>
- text: "Awards"       {% if forloop.last == false %}
  url: /awards         |
- text: "Gallery"      {% endif %}
  url: /gallery        {% end %}
                      ...
```

RUNNING JEKYL

- ▶ `jeekyll build`: Compile all the things!
- ▶ `jeekyll serve`: Same + start a small server to see the result locally
- ▶ `jeekyll serve --watch`: Same but recompile every time some file changes (apart from `_config.yaml`)

So basically, only one command.

OVERVIEW

What does "having a website" mean?

An example of static site generator: Jekyll

Hands on example

HANDS ON EXAMPLE

- ▶ Download <https://github.com/bk2dcradle/researcher>
- ▶ or find something you like on <http://jekyllthemes.org/>
- ▶ Look at the directory structure
- ▶ Generate and visualize the result
- ▶ Upload to the internet
- ▶ Configuration file
- ▶ Edit and play