EP1000

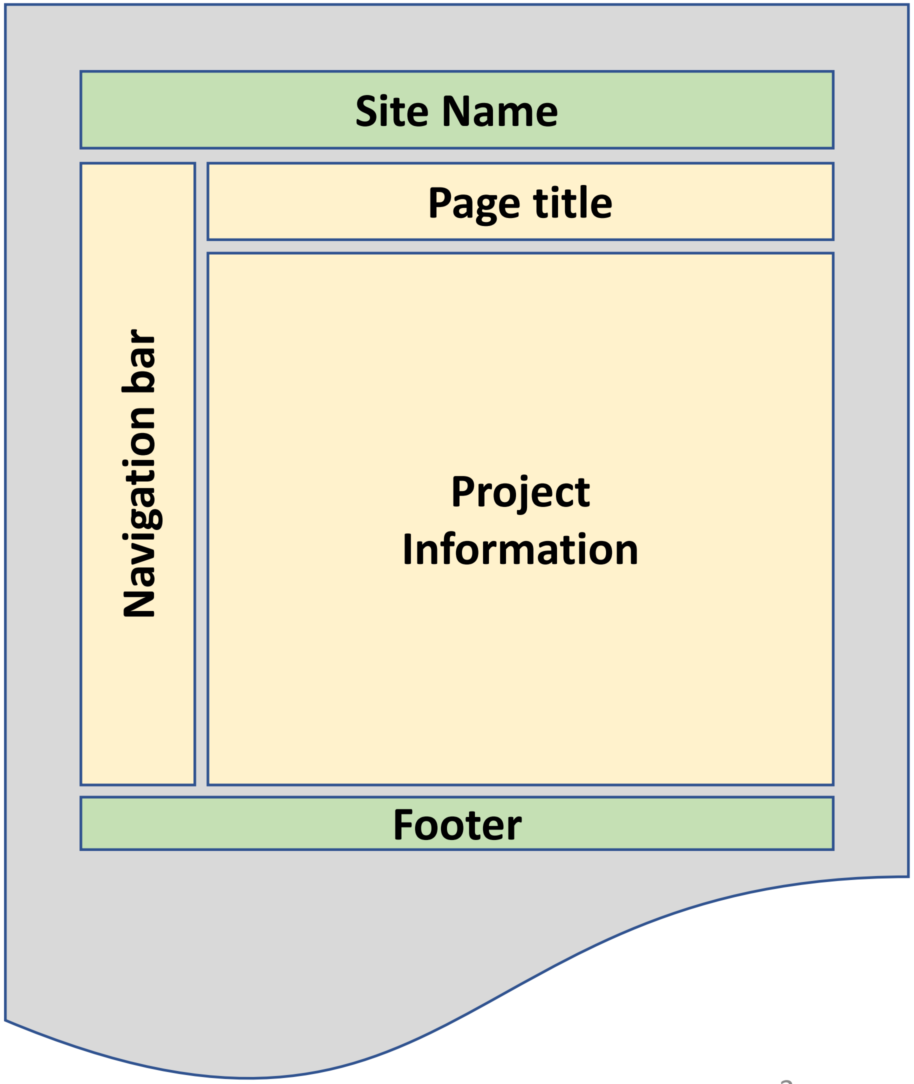
Markdown

Site Templates

- Typical site organisation
  - Main page
    - Describes the context of the site, an introduction of what is to come, why the site was built
  - About page
    - Provides information about the author, how to contact him/her
    - Provides background to the authors
  - Project pages
    - A separate section made up of project pages
    - Each page describes a task or a context of help
  - Final Project page
    - A major project construction
    - Communization of the work done in sections

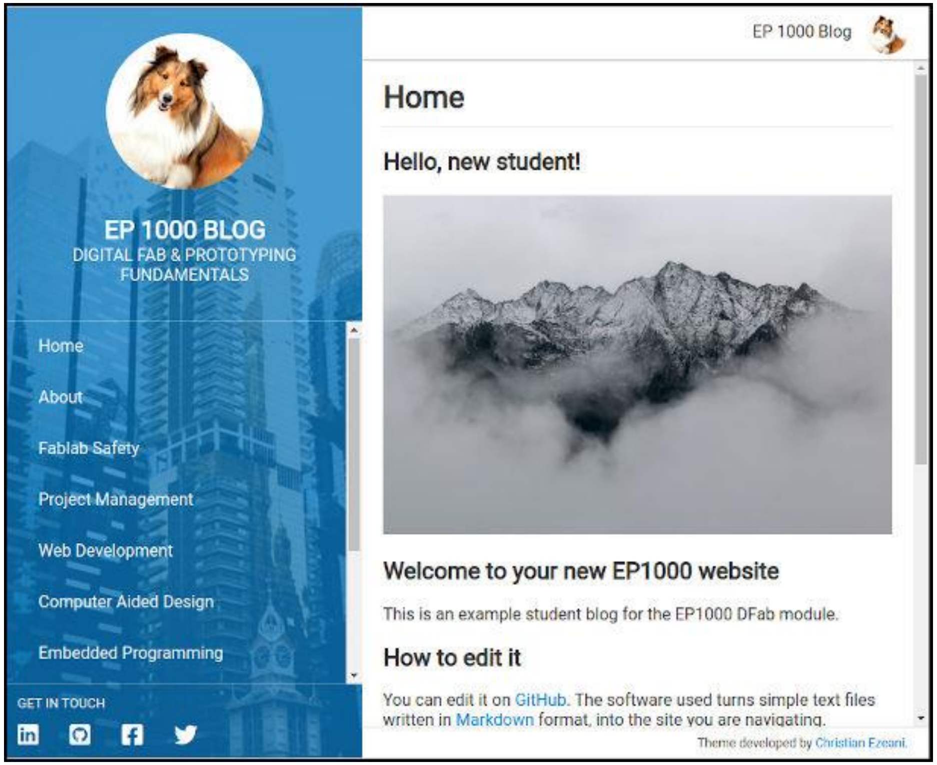
Page Templates

- In order to maintain uniformity, each page should be built from the page template
- Contains
  - Navigational information
  - Page title, page information,
  - References, links
  - Footer section for informational bits and links
- Consider
  - What can be duplicated?
  - Which section should change?
  - How can we automate the adding of pages?



Can we “borrow” examples?

- JW Sample Site
  - EP1000 example
- W3schools
- Static Site Templates
- Static Site Builders
  - HUGO
  - Jekyll (EP1000 example)
- Other examples
  - ReadTheDocs
  - GitBook
  - TiddlyWiki, MoinMoin, Doku-Wiki



Home

Hello, new student!

Welcome to your new EP1000 website

This is an example student blog for the EP1000 DFab module.

How to edit it

You can edit it on GitHub. The software used turns simple text files written in Markdown format, into the site you are navigating.

Theme developed by Christian Ezeani.

Active sites:
CMS, WordPress, Site Providers, ASP

Static Sites

- Sites that are cannot be changed/loaded from a database
- Why? No embedded code

Advantages:

- Full control
- HTML=content, CSS=layout
- Use templates
- Beautiful
- Can add some extra functionality
    - Javascript
    - Java
- Can be Responsive (Bootstrap)

Disadvantages:

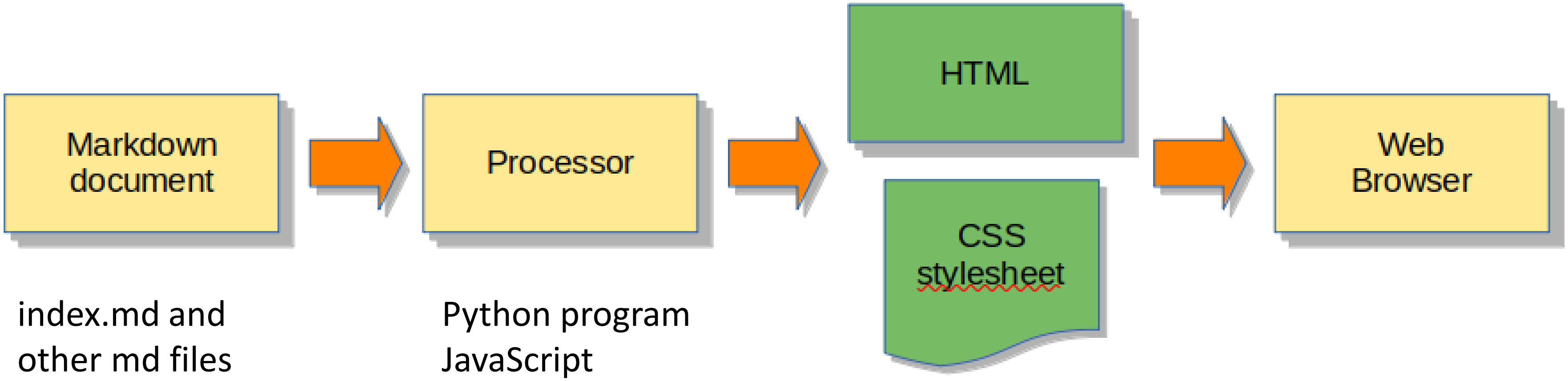
- Need to write in HTML and CSS
- Need to learn HTML, CSS
    - Javascript, Java
    - Bootstrap
- Templates may be confusing
- Too much control
- Need to understand the template

- Can't I just write in plain text ?
  (yeah, right, you're just one of the top 100 public intellectuals of US)

Markdown

- Light-weight markup language for creating formatted text using a plain-text editor.
- Markdown is widely used in blogging, instant messaging, online forums, collaborative software, documentation pages, and readme files.
- You can learn it in less than 15 minutes!
  Mastering Markdown
  Markdown Guide

Markdown Processing



Markdown document

Processor

HTML

CSS stylesheet

Web Browser

index.md and
other md files

Python program
JavaScript

GitHub automatically processes .md files when they are committed to the repository.

Converted files (HTML)
are stored in a temporary buffer, unliked from the md files.

Github pages serves them up.

Markdown Editors

- Online Markdown Editors
  - HackMD
  - StackEdit
  - Dillinger
- Plug-ins for common editors
  - Sublime Text
    - MarkdownEditing,
    - Using Markdown in Sublime Text 4
  - Atom
    - Set up & Use Atom as a Markdown Editor
  - Visual Studio Code
    - Markdown and VSC
- Any plain-text editor will work!
  - Markdown Live Preview

Github & Markdown

- Github automatically converts markdown files into HTML
- Site starts with index.md
- Problems:
  - Limited formatting (there are some available themes)
  - Boring….mainly text
  - Poor control of data flow, layout
  - Need a program to preview your work (Github may be slow)
- Solutions:
  - Easy to write documentation
  - Use HTML when you need control

Alternatives

- Strapdown
  - Original ides: strapdown.js
  - Javascript wrap-around for Markdown Pages
  - Can be online or on-site (move the JS files)
- ReMarkJS
  - Convert Markdown to presentation slides

EP1000

Markdown

End