LEARN WEB

ale66

BEYOND HTML

SATURDAY SPECIAL ARRANGEMENTS

Around 10:30:

- end-of-class coffee at the *Babila Hostel-Bistro*

WEB PAGES, TODAY

Writing direct HTML today could actually make you look strange, at least:

There are increasing levels of adoption of standard *platforms:*

sets of ancillary `.css` and `.js` files which are included in the page and determine the overall visuals and how the page interacts.

Bootstrap 5 is a platform.

1. make your own files, include CSS and javascript

Total creative control

need to learn and practice HTML, CSS and JS.

2. Include a Bootstrap 5 template.

Can you follow the Bootstrap 5 Cheatsheet?

3. include a Wordpress theme.

Less control but a uniform, smooth appearence.

4. include a Javascript *framework*

pages are created by JS functions which parse cookies and users' moves and select the appropriate elements.

Examples: Angular (Google), Webix...

able to collect and send back for analysis the telemetry of user movements, e.g., how long time is dedicated to reading each post.

5. work in a programming language to create HTML files dynamically.

Example: the Flask module for Python

Create HTML files dynamically, by writing Python functions which return HTML code

Take module `C4CSS: Coding for the Computational Social Sciences (6CFU)` to learn Python.

Mixing HTML and Python code in the same file:

```
1  <!doctype html>
2  <title>Hello from Flask</title>
3  {% if name %}
4    <h1>Hello {{ name }}!</h1>
5  {% else %}
6    <h1>Hello, World!</h1>
7  {% endif %}
```

github.com/ale66/learn-web

ADVANCED MARK-UP

DECORATION AND INJECTION

Marking-up your text to program its visualisation is *decoration*

Examples: MS Word, HTML5

Whereas HTML5 can be inspected and modified, in MS Word your text is *embedded* into the file

INJECTION

Adding mark-up commands to your text with software is *injection*

An HTML5 with a side `script.js` is an example of injection.

As G. Orwell would put it: decoration good, injection better!

Today injection is often carried out via pandoc.

THE PANDOC SOFTWARE

Pandoc = all documents.

Peels the decorations out, gives the content to the needed **injector**

On Win, install it via Chocolatey:

```
1  choco install pandoc
```

On Apple, use Homebrew:

```
1  brew install pandoc
```

WORK WITH PANDOC

Magic commands to transform files:

```
1  pandoc --from=txt mytext.txt -t html -o mypage.html
```

```
1  pandoc --standalone --from markdown --to html5 myfile.md -o myoutputfile.hmtl
```

more examples here.

Try it into the online sandbox!

Example

```
1  > pandoc -t markdown -o out.md .\grantPPA_PaccanaroRuediger-textonlyexample.txt
```

PRESENTATION VS. WEB PAGE

Today this distinction is blurring thanks to Javascript-based presentations that become web pages.

- presentation: synchronous, restricted

- Web page: asynch., universal.

Yet, we see the rise of manifold pages.

REMARKJS

MD cheatsheet

Familiarise with the Remarkjs FrameworK

https://remarkjs.com/#1

The '#1' is the page/slide number:

https://my-schematic-remark-presentation.andreamariadt.repl.co/#1

See the source of the intro slides to Remarkjs

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      ...
5    </head>
6    <body>
7      <textarea id="source">
8        name: inverse
9        layout: true
10       class: center, middle, inverse
11
12       ---
13
14       # remark
15
16       [ri-mahrk]
17
18       .footnote[Go directly to [project site](https://github.com/gnab/remark)]
```

Notice non-MD directives, e.g., `class: center, middle`

FROM MD TO SLIDES

Pandoc can generate a HTML file with slides

Use MD annotations with two extra conventions

By default, `#` for section title and `##` for slide title

a slide separator is needed: it redefines the meaning of `---` in MD

Sometimes `-----` is used.

Idea: keep the Markdown in a separated file, keep the HTML and Javascript boilerplate in a general `index.html` file

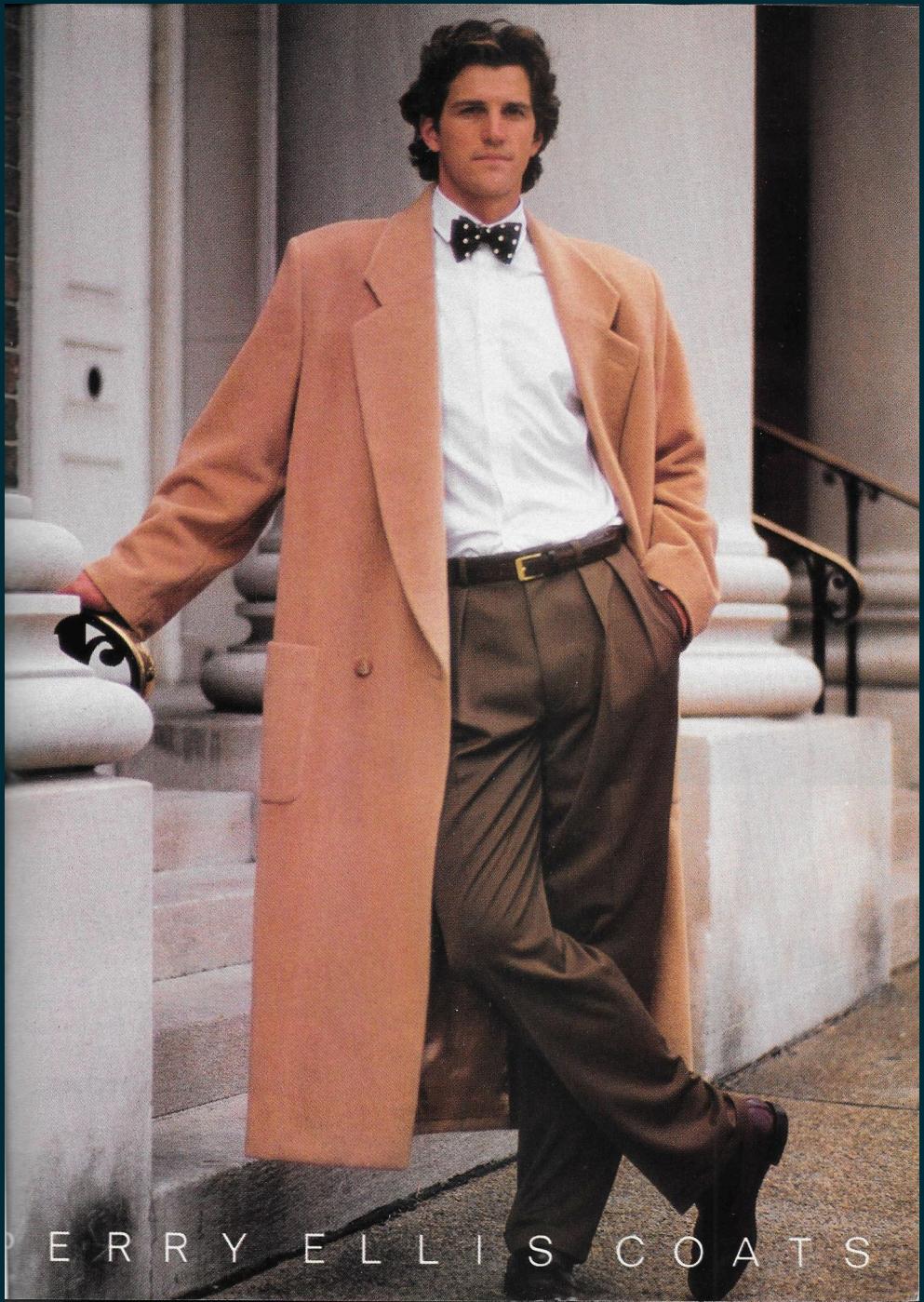# REVEALJS

A project now available as a stand-alone slides.com

A refined set of Javascript embeddings is available from the main site

Let's focus on MD embeddings

WHAT ABOUT POWERPOINT?

The first release was in 1987 [source: Wikipedia], i.e., well before the *internet revolution*.

For reference, this was fashion in 1987

PERRY ELLIS COATS

github.com/ale66/learn-web

Today,

1. everything is hyper-linked and hardly any document is self-contained

2. Cross-publishing invariably means changing styles

3. yet, content is often ambivalent synch/asynch

JS-based presentations have the concept of *linking* (and re-processing) at the core.

Powerpoint can only hope to import it in...