

ADVANCED WEB TECHNOLOGIES

SET09103

TOPIC #04

FRONTENDS, UI, & UX

Dr Simon Wells

s.wells@napier.ac.uk

<http://www.simonwells.org>

TL/DR

- Front-end development is an active & innovative area of web technologies
- At some point, systems must be exposed to an end user of some sort (for the moment, usually a human)

OVERVIEW

- **Part#1:** Front-Ends, UI, & UX.
- **Part#2:** Flask Examples from the project book.

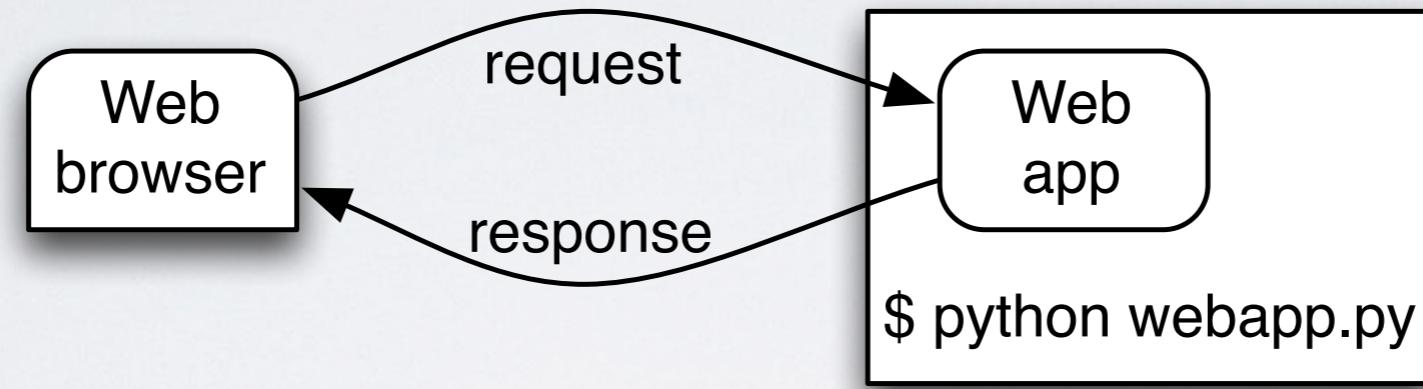


PART #1: BACKGROUND

FOCI

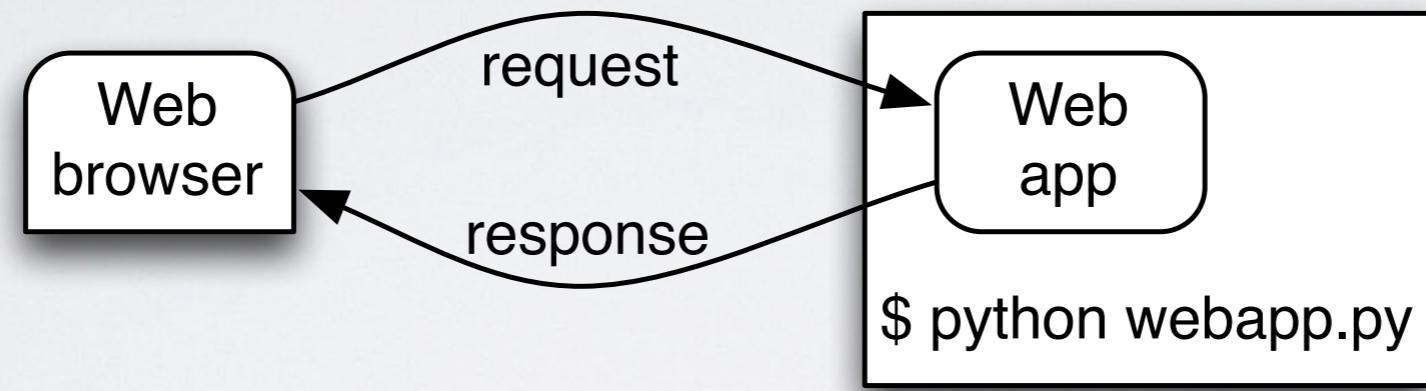
- Two foci for this week:
 - How to make use of HTML, CSS, & JS from with Flask.
 - How to design better user interfaces (using those tools (especially if we're not confident in UI/UX)).

OUR CURRENT ARCHITECTURE



- Client - makes request.
- Server - provides response:
 - **So far:** String containing plain text or HTML..
 - **Today:** HTML files linking to other static files (CSS, JS, images[jpg,png,gif], ...).
 - **Later:** JSON/XML - important aspect of APIs for machine to consume.
- This week only:
 - Let's consider the front-end -the bit for **people** to consume (left hand side of the diagram).
 - Do enough on the right hand side (in Flask) to make the left hand side a richer & more rewarding experience.

FRONT-END



- From human perspective:
 - The User Interface (UI) & User Experience (UX)
 - Common approaches:
 - HTML+CSS+Javascript - very common. The three core, front-end, web technologies
 - Build it yourself or use a pre-packaged UI library (e.g. bootstrap)
 - Javascript Framework + JSON - increasingly common

HTML

- HyperText Markup Language
- Maintained by the Web Hypertext Application Technology Working Group (WHATWG)
- HTML Document contains HTML **elements**
- Elements delineated by **tags** - things within '<' & '>'
- Tree structure - everything between <html></html> tags

```
<!DOCTYPE html>
<html>
  <head>
    <title>This is a title</title>
  </head>
  <body>
    <p>Hello world!</p>
  </body>
</html>
```

CSS

- Cascading Style Sheets
- Designed to separate how things should look (responsibility of CSS) from describing what things are (responsibility of HTML)

- Used to do styling inline:

- `<h1>Hello Napier</h1>`

```
h1 {  
    color: red;  
}
```

- Now have linked css document (from HTML head section):

- `<link href="path/to/file.css" rel="stylesheet">`
- `<h1>Hello Napier</h1>`

JAVASCRIPT

- A high-level, interpreted programming language

console.log("Hello World!");

- Increasingly used to provide interactivity & dynamism in the client
- (Increasingly used on server-side: unifies technologies between server & client. Also as a general purpose language.)

SERVING HTML FROM FLASK

- In your project folder (the same location as your `whatever.py` file):
 - Add a **templates** folder.
 - Put HTML in the templates folder.
 - Tell Flask to return the HTML by using the `render_template()` function.

e.g. `return render_template('index.html')`



SERVING STATIC RESOURCES FROM FLASK

- Only HTML files are treated specially and put in the *templates* folder. Everything else is a static resource and goes in the *static* folder.
- In your project folder (the same location as your *whatever.py* file):
 - Add a **static** folder.
 - Put CSS, JS, images, ..., file in the static folder.
 - Tell your HTML where to find the static resources that it uses by using the *url_for* function

e.g. `url_for('static', filename='some-image-file.png')`

UI TOOLKITS

- Lots of bootstrap style UI toolkits available:
 - ...
- Some are barebones. Others are batteries included. Some are more opinionated & others less so.
- Try some out. See what works for you (different projects might necessitate different solutions)
- These are usually some combination of HTML, CSS, JS, and other static resources.
 - They obey the same rules on templates and static folders.
 - Can be convoluted to set up.

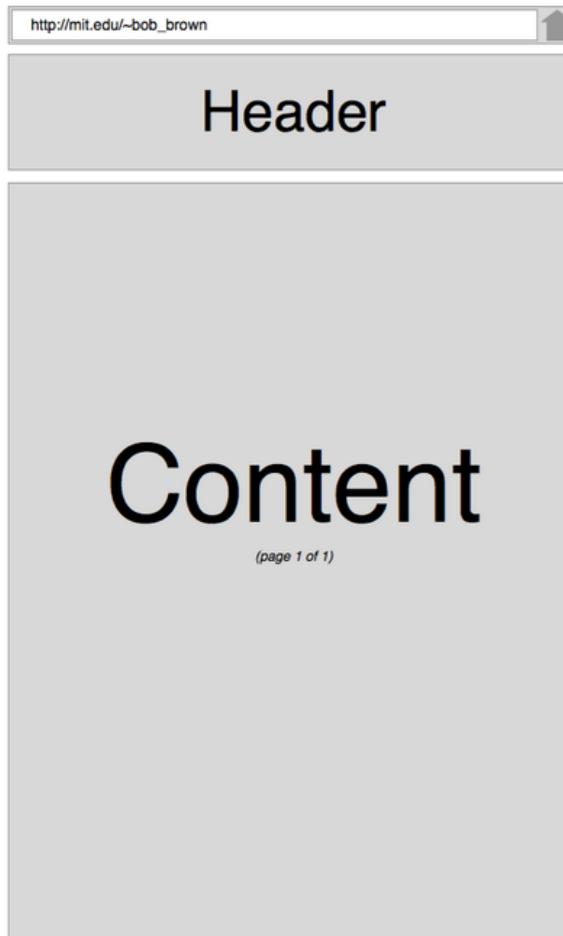
BOOTSTRAP

- Quick way to theme prototypes so that:
 - They don't look very ugly
 - They are *reasonably* consistent
 - Not reinventing the wheel every time you prototype - Get something built, make it look not awful
- <http://getbootstrap.com/>
 - Includes HTML fragments (basic layouts), CSS (bootstrap default look & feel), JS (responsiveness & UX)
 - Perfectly fine to use bootstrap or other toolkits if you are not confident in your ability to produce a pleasing UI.

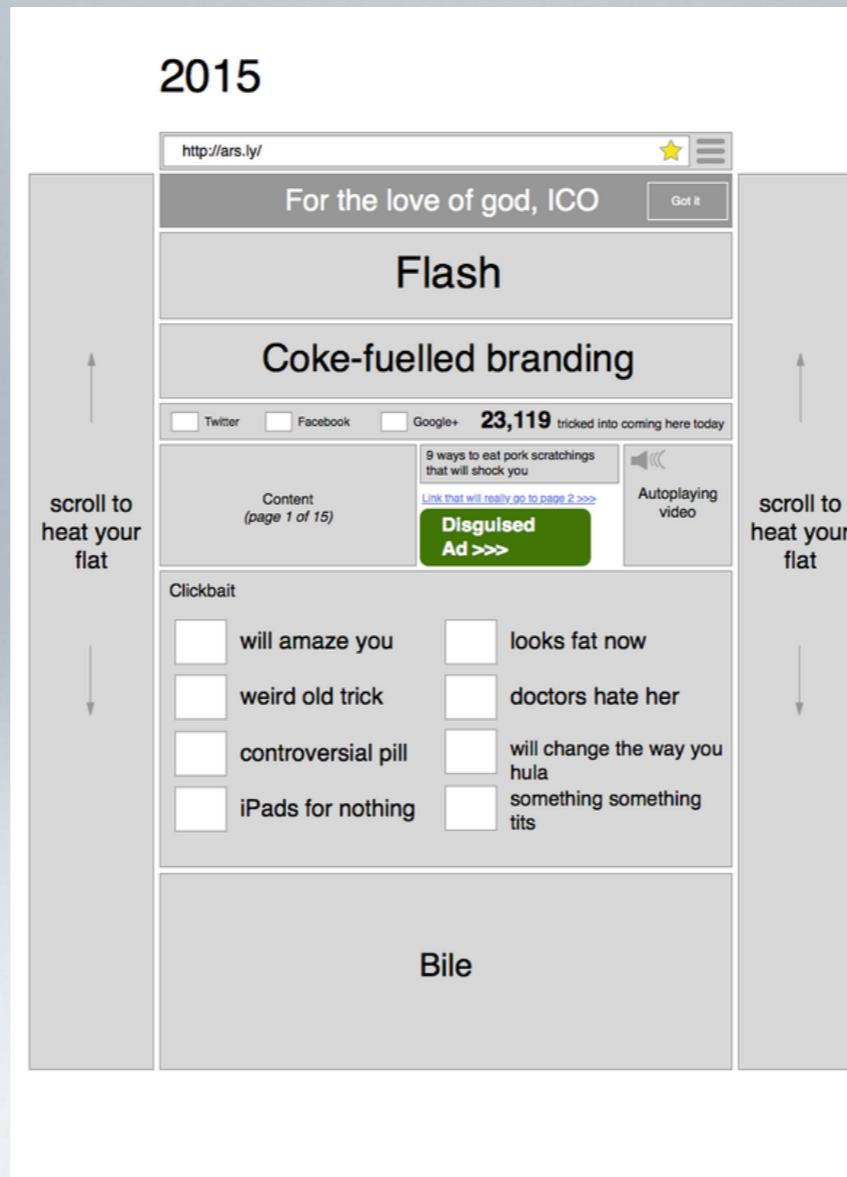
DATA DRIVEN UI

- HTML+CSS+JS still works really well - default implementation mode for web pages.
Browsers optimised for these technologies
- However: Sites increasingly data driven (UI generated on the fly, i.e. fulfilling a template with data to complete HTML page)
- Why not just return JSON (or any other data format) to the browser and take the load of generating pages off the server?
- Each client now assumes responsibility for generating the UI locally
- Like bootstrap a lot of competing toolkits that will turn your JSON data into web-pages, e.g.
 - React - ...

1995



2015



*What characterises
Web page design
in 2025?*

UI INNOVATION

- Innovation is hard (to get right)
- Coming up with truly novel interfaces (that actually work well) is incredibly difficult
- Need to understand your data, what you want to do it with, & what you want your user to understand from it.
- Scatterplots/tables+filtering+zooming can make a useful way to explore a complex landscape:
 - <https://www.productchart.com>
 - <https://diskprices.com> & <https://battprices.com/>
- Visualisations appropriate to your data, e.g. <https://uk.wikibudgets.org/w/united-kingdom-budget-2015>
- There are *many* poor ways to innovate:
 - Rotating Cube Interface
 - Volume Sliders

DESIGNING & CODING

- Not aiming to build the worlds most beautiful sites (although we can if we want)
- But should aim to provide the best user experience we can manage
- Designers work really hard to ensure that every element is perfect and beautiful
 - Design is not an afterthought but an integral part of the process
- NB. Design has a huge effect on perception of your site **but** there is no “right way” to do design (not yet?)

LOOK AT EXAMPLES

- Professionals (of all kinds) critically evaluate other people's work to see what they can learn
- When was the last time you looked closely at a site that you think is visually pleasing and noted
 - how they achieved the things you appreciate
 - determined how to get a similar effect in our own sites?
- Why not? You probably do that with code. Look at examples (e.g. from Stack Overflow) and glue them together (at least until we develop sufficient knowledge and experience to work out own solutions).

DESIGN FOR HACKERS: OVERVIEW

- Use of frameworks/toolkits
- Prototyping:
 - Mockups\Wireframes
 - Placeholder text (Greeking)
- Colour
- Typography
- Consistency through documentation & planning

PROTOTYPING & MOCKUPS

- Many folk design sites on paper - just draw out your design. Indicate where the various parts will be. Then implement your design
- Or on a whiteboard (post-it based approaches can be useful)
- Others use general image/graphics programs - Simon uses Omnigraffle (MacOS only) - Object based tools that enable you to move layers and group things together can be really useful
- Dedicated mockup tools - Many choices. Quite a number of online tools, some offline tools.

ONLINE WIRE-FRAMING & MOCKUPS TOOLS

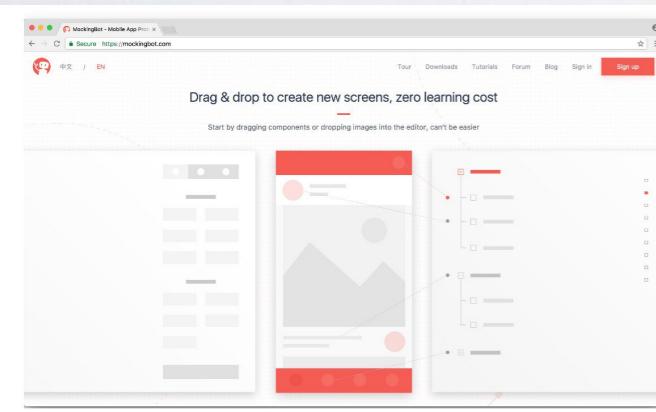
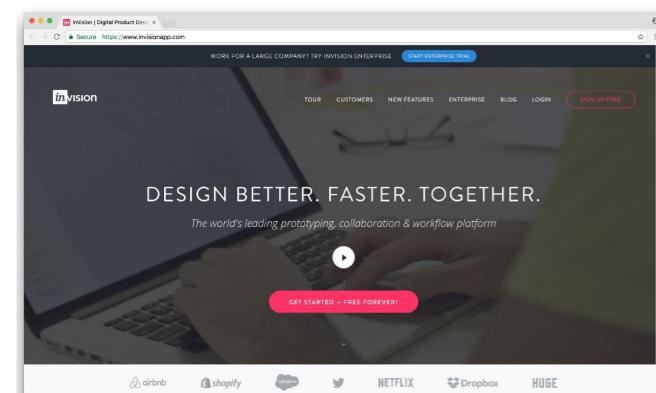
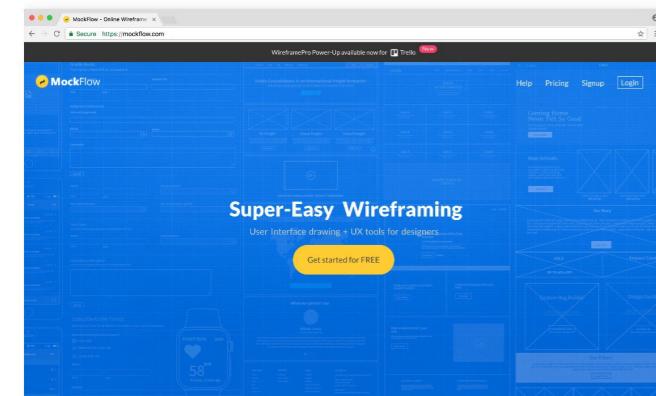
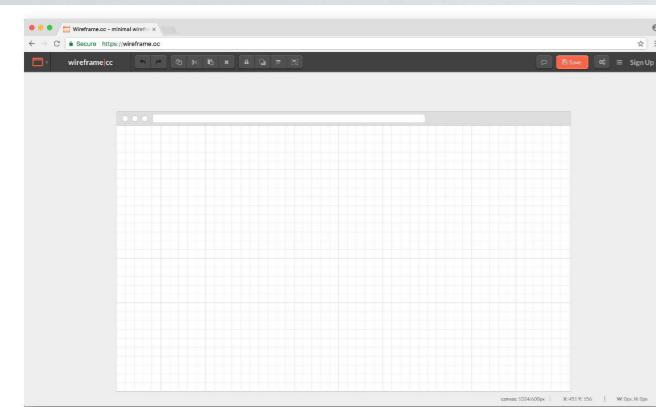
- There are many. Here are a few, some have free access, others not:

- <https://wireframe.cc/>

- <http://mockflow.com/>

- <https://www.invisionapp.com/>

- <https://mockingbot.com/>



PLACEHOLDER TEXT (GREEKING)

- So far we've assumed that we have data and just need to build a website to show it
- Sometimes we don't have data or content during design/development phase.
 - Might not be written yet or might not exist - can be very effort intensive to put together realistic text so that layouts look like you want them to
 - In fact writing this kind of text is a professional job: "copy writer" not to be confused with "copywriter":D
- The solution is to not bother with realistic text until you really need it
 - You (as designer/developer) know what the content will eventually be so you don't need real text

HOW GREEKING WORKS

"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum."

- The key is that the placeholder text looks like real text
- It has the same rhythms & cadences as real language, & hence real copy
- Word lengths & frequencies are similar so we can assume that text of a certain length (word count) will fit the area we've designed for
- This has been standard practise in publishing for many years
- You know roughly how many words an article will be but don't have the actual article yet.

DESIGN GUIDELINES

- After the first initial prototype:
 - You will have something that works (functionality) & looks OK (bootstrap)
 - You will probably want to personalise it (or your client might have branding that needs to be applied)
 - Need to develop **design guidelines** for your site:
 - A cohesive description of the look & feel of your site that describes every element
 - Can be used to make sure you are consistent
 - To make sure others are consistent (just a few “additions” can turn a well designed site into a dogs dinner)
 - Extends the life of your project by making it maintainable
 - Can be a **document**, a **page** of your site, an entire **language**.

DESIGN GUIDELINES IN THE WILD

- Used by many organisations to design:
 - Websites
 - Mobile apps
 - Paper comms (posters, flyers, leaflets) - *still very relevant*
- With aim of maintaining **consistency, cohesion, & predictability** both between and within communications.



TFL DESIGN STANDARDS

- TfL (Transport for London) publishes a corporate design standard to ensure that all communications maintain their design ideals
- “This guide is designed to help us work as one brand, and to make it easier to produce high quality communications, experiences and propositions both internally and externally.”

CONTENTS

OVERVIEW	5	1. TYPOGRAPHY	11	4. UI ELEMENTS	35
DIGITAL EXPERIENCE PRINCIPLES	6	2. COLOUR PALETTE	15	4.1 Buttons & calls to action	36
DEFINING THE VISUAL DIRECTION	9	3. UI STRUCTURE	19	4.2 Tabs	38
CREATIVE DIRECTION: LOOK & FEEL	10	3.1 Responsive design	20	4.3 Line dividers	39
		3.2 Universal grid	22	4.4 Pagination	40
		3.3 Baseline grid	23	4.5 Forms	41
		3.4 Desktop grid	24	5. ICONOGRAPHY	42
		3.5 Tablet grid	25	6. IMAGE SIZE RATIO	44
		3.6 Mobile grid	26		
		3.7 Pattern behaviour	27		
		3.8 Padding	28		
		3.9 Global masthead	29		
		3.10 Breadcrumbs, headers and icons	31		
		3.11 Footer	34		



NHS BRAND GUIDELINES

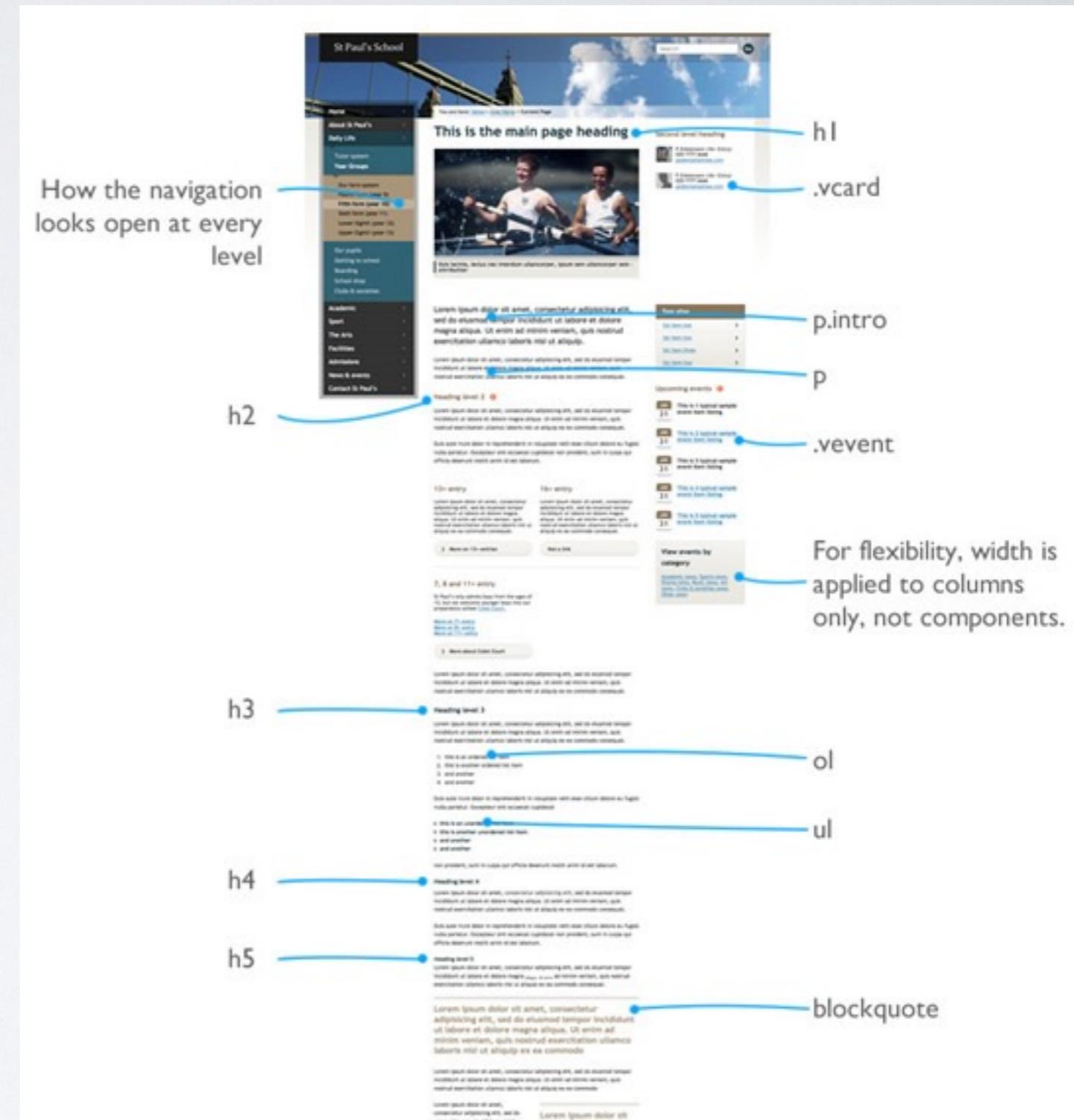
- Slightly different focus (England Wide) & covering local GP practices as well as hospitals and 'corporate' communication
- Aims to enable local doctors to develop, e.g. information leaflets, that are in keeping with wider NHS corporate branding standards
- There is an additional element of **trust** here. There is lots of health communication that is quackery at best & outright dangerous at worst. Important to establish **provenance**.
- <http://www.nhsidentity.nhs.uk/>

BBC GEL

- GEL - Global Experience Language
- Global - refers to the whole experience & the reach
 - Ensure all sites under the BBC brand are coherent
- <http://www.bbc.co.uk/gel>
- <http://gel.ed.ac.uk/>

DESIGN DOCUMENTS

- A single **deliverable** that comprises a single HTML page which contains an example of every element, style, & component for your site.
- Really nice because it enables you to see every design elements for the site in one place so you see combinations of elements at the design stage rather than the implementation stage.
- <https://paulrobertlloyd.com/styleguide/>
- <http://oli.jp/2011/style-guide/>

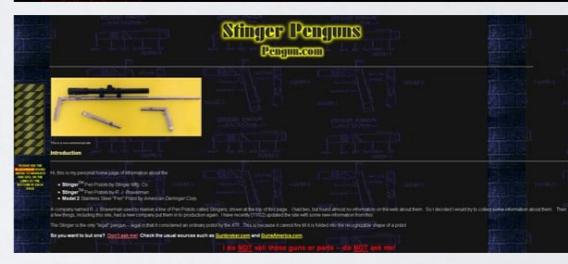
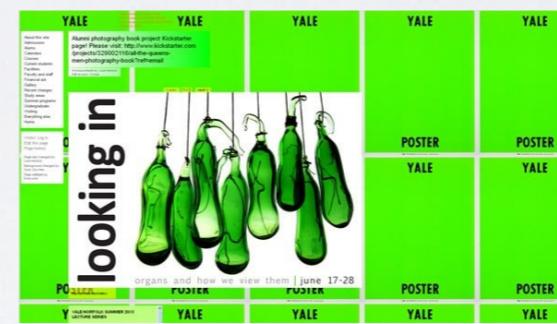
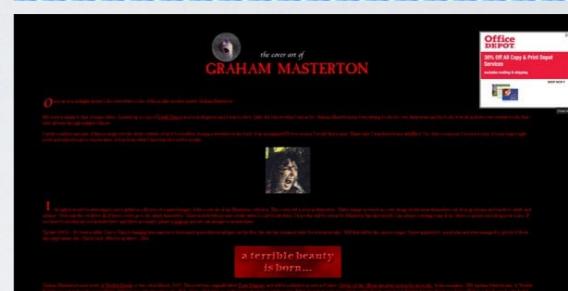
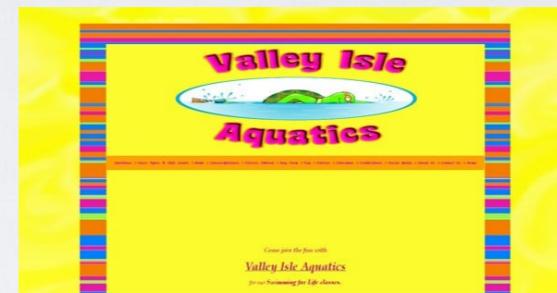
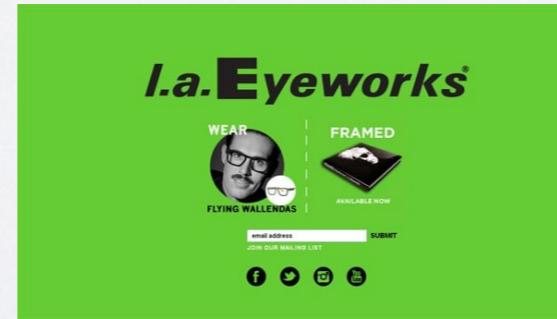


NOT JUST DESIGN

- Standards used everywhere to try to bring about consistency, cohesion, and predictability.
- Coding, Design, Development, Usage Standards:
 - Programming Languages:
 - Python PEPs
 - Java
 - C/C++
 - Technology Communities:
 - Drupal: <http://drupal.org/coding-standards>
 - Wordpress: http://codex.wordpress.org/WordPress_Coding_Standards
 - Apple HIG (human interface guidelines): <http://developer.apple.com/library/mac/#documentation/UserExperience/Conceptual/AppleHIGuidelines/Intro/Intro.html>
 - Mozilla,
 - Android (Material Design),
 - Twitter (Bootstrap)
 - ENU, School of Computing

POOR COLOUR SCHEMES

- Green, white, yellow on green background
- Predominant use of blue as background
- Light objects on light background
- Bright colour combinations
- Bright/Textured backgrounds with coloured text
- Vibrant colours against black background
- Too many colours



LICK HERE TO GN YVETTE'S QUEST-BOOK !!

Yvette's Fashion Models!! click here

Yvette's has gorgeousowns full figure!! [click here](#)

Jewelry, Pearls, Varovski crystals, necklaces, earrings, bracelets, Aurora Borealis stones, click here

Pageants
Mad to Yvette's LOCATION!
Please click here !! ~*

Мы любим Россия.

Yvette's sells
Kitty Chen Couture,
click here ~Sensational !!~
Fashion Label~

Nous aimons la France.
Tuxedo rentals
starting @
\$65.00 ~*

Welcome to YVETTE'S ...

Established
1980
sensational !! ~

PAGEANT
PROM
HOMECOMING
CRUISE
HOLIDAY
CELEBRATE

Lauren, Brooke & Brittany trying on Gowns @ Yvette's, click here

Yvette's Bridal Gowns

Yvette's is here for YOU !! ~*~*~*

Celebrate! Prom 2011 !! ~*~*~*

Designer Fabrique ~

Apple Pie , please click here ~

please click here to Advertise on Yvette's Site !! ~*~*~*~*~*~*

Easy Yvette's Navigation !! ~* please click here !! ~*~*~*~*~*

Yvette's Paintings for sale !!, please click here !! ~*

Yvette's phone number: (850)~871~3000

Yvette's Loves You !! ~* Yvette's Loves You !!~*

email Yvette's! [ENTER YVETTE'S](#)

click here!

~Pink Links to Your DESIGNERS~

Follow the Pink Links to Yvette's !! ~*

Yvette's sells tuxedos !!

gown Panama City

850-871-3000

Invitations !!
[please click here ~*](#)

우리는 한국을 사랑한다.

Call to Artists !! ~* please click here !! ~*

我們愛中國。

Yvette's habla poquito espanola !! ~*

私達は日本を愛する。

The screenshot shows the homepage of the Bavarian Brathaus website. At the top left, there's a cartoon character of a leprechaun-like figure with a pipe. To the right, the word "Bavarian" is written in white script inside a red circle, and "Brathaus" is in blue inside a blue banner. The background features a brick wall texture. The main menu items are listed in large yellow text: "19 German Draft Beers", "Authentic German Restaurant", "Outdoor Biergarten", and "Let's Strudel Baby!". Below the main menu, there's a section for "Oktoberfest 2012" featuring "The Little German Band" on Friday, June 21st. On the left side, there are links for "Menus", "German Draft Beer (Fassbier)", "Menu", "Kinder Menu Dessert", "Vegetarian Fare", "Live Music", "Bavarian Brathaus Reviews 2010", "Reviews 2011", "Reviews 2012", "Address", "Hours", "Map", "Grilling BBB Bratwurst at home", and "Own a Bavarian Brathaus".

CHOOSING COLOURS

- Start thinking about how many colours you need during your design process,
 - e.g. background, foreground, heading text, emphasised text, body text, hyperlinks, boxes, semantic blocks, &c.
 - This will give you an idea of *how many* colours you need in your **palette** but not concerned with what the colours are yet
 - Colours can be re-used in multiple places - so aim to minimise the number of colours - A five colour basic palette is very common even for complex sites

PALETTE SELECTION TOOLS

- **Use a palette building tool to design your palette for you**
 - e.g. You select dominant or base colour for the scheme, the tool selects **complimentary, monochromatic, triad, analogous, compound, or shade** based groups of colours for you from the colour wheel on the basis of colour theory.
 - Use the colours from the tool consistently in implementing your design - don't deviate. If you need to deviate then it means there is another iteration of design required.
 - If you run out of colours then return to the palette tool and increase the size of the generated palette and reapply all of the adjusted colours to your design (*this is why separating out HTML [markup] and presentation [CSS] is a good thing - altering the CSS alters everywhere it is used*)

COLOUR TOOLS

- This is a selection of common online palette generation tools:
 - <https://coolors.co/>
 - <http://www.colourlovers.com/palettes>
 - <http://paletton.com/>
 - <http://www.color-hex.com/color-palettes/>
 - <https://color.adobe.com/create/color-wheel/>
- There are others. Find one that you like and which you can use effectively.

TYPOGRAPHY

- Traditionally arranging type to make text legible, readable, and appealing but also: style, arrangement, & appearance of letters, numbers, & symbols.
- Makes use of typefaces (fonts), point sizes, line lengths, letter-spacing, line-spacing
- Most browsers include a family of fonts by default that cover most use cases
- Choosing the right font can give your site character.
- As with colours, using them well is tricky.

DESIGN FOR CODERS

- Use palette selection Tools to generate a decent set of colours then use colour carefully & deliberately
- Mock-up your layout using greeked text
- Make good use of whitespace (not too cramped)
- Don't use too much whitespace (good information density)
- Use less furniture (rules, boxes, dots) around your data
- Spacing & alignment are hugely important (especially when comparing data), e.g.
 - Align digits with monospaced fonts
 - Align decimal points (particularly for currency)
- Narrow line heights keep data close together (but make sure it is cleanly separated)
- Reduce volume of data:
 - If have first and last name in separate columns can they be combined into "name" when displayed? Town, County, Postcode?
 - Remove redundant data & anything the user doesn't need to see
- Ask your users (but they can be wrong). Watch your users, but don't slavishly reimplement real-world procedures
- You'll never make the perfect UI for all people, so why not let your user export (CSV, JSON, &c.) your data so they can view it in a better (for them) tool?

SUMMARY

- Considered
 - a variety of factors that affect our designs (Colour, Typography)
 - a range of tools & techniques to help manage UX design:
 - prototyping, placeholder text, Colour palette generators, design guidelines

PART #2: PRACTICAL WORK

EXAMPLES

- Hello-static example.
- Static Resources example.
- Project Book: Simple Quiz with HTML files examples

WRAPPING UP

- Static resources & templates in Flask
- Design for Hackers