LECTURE 30

PLANNING OUR WEB APP

MCS 275 Spring 2023

David Dumas

LECTURE 30: PLANNING OUR WEB APP

Reminders and announcements:

- Please install **Flask**, e.g. with

  ```
  python3 -m pip install Flask
  ```

  in preparation for using it in class soon.
- Homework 11 due tomorrow.
- Project 3 solution posted.
- Project 3 grading and Project 4 description likely to be finished tomorrow (Tue Apr 4).

TODAY

Create the rough design for a web application we'll write in Python in the next few lectures.

Continue study of HTML and CSS by making some mockups of the user interfaces.

TOY MODELS

I will prioritize getting a working product that uses things we've discussed in the course.

That will mean skipping some natural features (e.g. authentication) and avoiding some technologies because they'd take too long to introduce (e.g. JS), or because they'd hide the use of things we're learning (e.g. SQLAlchemy).

WORK ORDER SYSTEM

A **work order** (WO) is a request that a certain job be done (e.g. "fix flickering computer projector in lecture room" or "replace damaged lock on tiger cage").

After a WO is created, it needs to be assigned to a person who will do the job.

Later, the person who the WO was assigned to may mark it complete.

The "ticket systems" used to track support requests are similar.

A system to track WOs would have several user types:

- **Submitters**
  - Create new WOs
- **Workers**
  - View available and assigned WOs
  - Take on assignments
  - Mark an assigned WO complete
- **Administrators**
  - Modify/delete WOs in other ways

SIMPLIFICATION

As we won't build any user account system, we'll just create separate pages (or *views*) in our web application that serve the needs of these different user types.

App name?

Let's try a Business Name Generator.

UI MOCKUPS

Let's make some HTML+CSS documents that look like the applications we're planning to build.

Later, we'll use these as the starting point for the HTML the app will generate.

CSS

A language that styles HTML. Basic syntax is

```
SELECTOR {
    PROPERTY1: VALUE1;
    PROPERTY2: VALUE2;
}
```

Elements that match the SELECTOR will have their styling changed by adjusting values of the given properties (things like color, font size, border).

CSS has a complex system of units for values like distances and sizes.

```
body {
    font-family: "Verdana", sans-serif;
    background: #C0C0C0;
}
p {
    border: 1px solid #000000;
}
```

The body element of the document (hence everything) will use the font called "Verdana", or if that isn't available, some sans serif font.

The body will be displayed with a light gray background.

Each paragraph will be surrounded by a 1-pixel black border.

HEX COLORS

The color code

#789ABC

means

- Amount of red = `0x78` = `120` out of `255`
- Amount of green = `0x9A` = `154` out of `255`
- Amount of blue = `0xBC` = `188` out of `255`

Google has a color picker that can be helpful.

CSS PROPERTIES

There are lots, e.g. `color`, `background`, `font-family`, `font-size`, `width`, `height`, `border`, `margin`, `margin-top`, `margin-left`, `padding`, `padding-left`, ...

CSS properties reference at w3schools

APPLYING A STYLESHEET

You can embed a stylesheet (block of CSS) directly in HTML by placing it inside `<style>` within `<head>`.

Or you can use the HTML `<link>` tag to specify the CSS can be found at another URL (e.g. is in another file).

```
<link rel="stylesheet" href="mysite.css">
```

Like `<style>`, `<link>` should go in the header.

The latter approach makes stylesheets reusable.

ID

An HTML element in a document can be given a unique identifier with the `id` attribute.

```
<p id="selector-intro">Let's talk about CSS selectors...</p>
```

If two elements have same `id`, your HTML is invalid.

You can link to an element within a document by id e.g.

```
We discuss <a href="#selector-intro">selectors</a> below.
```

CLASS

You can create **classes** (categories) for elements in a HTML document. Mainly used so items in a category (e.g. important, outdated, footnote) can be styled differently (e.g. red, faded, small text).

Specify the category of an element using the `class` attribute.

```
<p class="urgent-warning">Space wasps approaching.</p>
```

No need to declare classes in advance, nor to refer to every class in your CSS.

CLASS AND ID SELECTORS

Apply class- or id-specific styles in CSS:

```
/* Text in paragraphs is orange */
p { color: #FFA500; }

/* Paragraphs of class "urgent-warning" have big, red
   text. This overrides the previous line, as the more
   specific selector gets precedence.  */
p.urgent-warning { color: #FF0000; font-size: 120%; }

/* The paragraph with id "selector-intro" is bold */
p#selector-intro { font-weight: bold; }
```

CHILD AND DESCENDANT SELECTORS

```
div ul {
    /* style for any ul that has
       a div ancestor */
}
div > ul {
    /* style for any ul that has
       a div as its parent */
}
ul ul {
    /* style for ul inside another ul */
}
```

MANY OTHER SELECTORS

`first-child, only-child`, selection by adjacency, values of attributes, ...

w3schools CSS selectors reference

PYTHON'S BUILT-IN HTTP SERVER

```
python3 -m http.server
```

Opens a web server that serves files in the current directory and its subdirectories.

Visit `http://localhost:8000/` in a browser (or substitute other port number shown in startup message) to see `index.html`.

Firewall rules typically prevent incoming connections from the internet (and maybe the local network too). That's good! Or

```
python3 -m http.server --bind 127.0.0.1
```

will make sure it **only** listens for connections from the same machine.

REFERENCES

- UIC course IT 202 teaches web design properly
  - (whereas for us it is a vehicle to demonstrate Flask)
- HTML tutorial from W3Schools (all in-browser)
- jsfiddle
- Countless web design books in the O'Reilly technical library (free to anyone with a UIC email address).

REVISION HISTORY

- 2022-04-06 Last year's lecture on this topic finalized.
- 2023-04-03 Updated for 2023.