FEWD - WEEK 2

WILL MYERS

Freelance Front End Developer

SLIDES

http://www.slideshare.net/wilkom/fewd-week2-slides

AGENDA

- Review
- Box Model
- Nested Selectors
- HTML Template
- Lab Time
  - How To Start

YOUR WEEKLY FEWD GITHUB REPOSITORY

- Use the '+' button in the top-left of GitHub Desktop (*Create* tab)
- Create a new repository called *'FEWD_Week2'*
- Choose the [home]/FEWD folder for the local path
- Open this repo folder in your editor
- Commit the changes and publish the *FEWD_Week2* repository to github.com

YOUR WEEKLY WORKING FILES FROM ME

To get the *week2_working_files* you should just be able to select the *ga-fewd-files* repository in GitHub Desktop and press 'Sync'. This should pull in this weeks folder from github.com.

If you any difficulties you should just re-clone the *ga-fewd-files* repository.

REVIEW

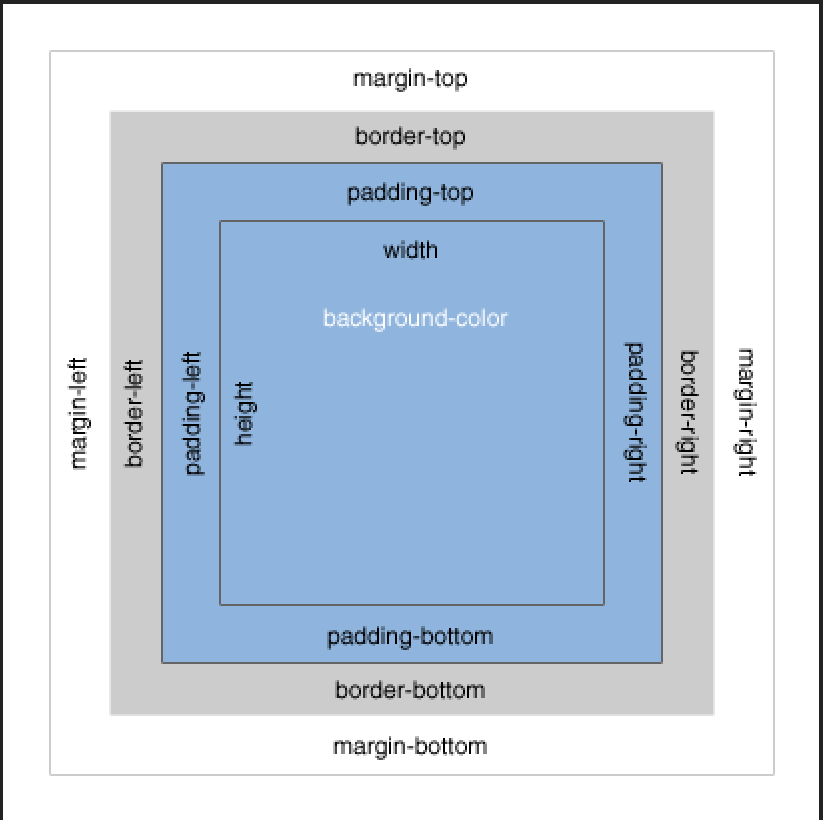
Let's have a look at some of your work from the last week!

BOX MODEL

Every element in web design is a box.

The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content.

BOX MODEL



margin-top
border-top
padding-top
width
background-color
margin-left
border-left
padding-left
height
padding-right
border-right
margin-right
padding-bottom
border-bottom
margin-bottom

MARGINS, PADDING AND BORDERS

- `margin`: sets the outer transparent rectangular border of an element
- `border`: sets the visible rectangular border style of the element
- `padding`: sets the inner transparent rectangular border of an element (is colored by a `background-color`)

`margin` area is transparent, `padding` area inherits background-color, `border` has its own style and color properties.

MARGINS AND PADDING

The values for `margin` and `padding` declarations can be set with shorthand:

- `margin: top right bottom left;` (clockwise)
- `margin: top-and-bottom left-and-right;`

Each side can also be set individually with a specific declaration:

- `padding-left: 10px;`
- `padding-top: 20px;`
- `etc`

BORDERS

Borders have their own style declarations:

- `border-width` (number px)
- `border-style` (string - e.g. solid, dotted, dashed)
- `border-color` (string or hex value)

The common shorthand syntax to set a border is:

```
width style color
```

```
border: 4px solid red;
```

BORDERS

Border style properties: `none` (low priority), `hidden` (high priority), `dotted`, `dashed`, `solid`, `double`, `groove`, `ridge`, `inset`, `outset`

Don't forget `border-radius`

`border-radius:50%;` makes a square into a circle

BOX MODEL

You can see a representation of the box model in Chrome dev tools (Cmd + Alt + I), in the 'Elements' tab.

BOX MODEL

Width = width + padding-left + padding-right + border-left + border-right

Height = height + padding-top + padding-bottom + border-top + border-bottom

Padding, border & margin will be outside of the box

```
.box {
    width: 350px;
    border: 10px solid black;
};
```

RESULT (rendered in the browser)

```
.box {width: 370px;}
```

BOX MODEL

It is possible to change what is included in a box model sizing using the box-sizing property.

You can change the value from the default `content-box` to `border-box`. This will *include* the `padding` and `border` in the calculated width rather than adding it on.

```
.box {
    width: 350px;
    border: 10px solid black;
    box-sizing: border-box;
}
```

This will lead to a box rendered in the browser of `width: 350px`

TAGS & BOXES

Copy the *tags_boxes* folder in *week2_working_files* into your *FEWD_Week2* folder and open it in Atom.

CSS POSITIONING

You can also position elements with exact values using the `position` property and `top`, `left`, `bottom`, `right` properties.

`position` has the following values:

- `static`: default positioning in the document flow
- `absolute`: positioned relative to its first non-static ancestor element
- `fixed`: positioned relative to the browser window
- `relative`: positioned relative to it's normal default position

http://codepen.io/wilkom/pen/xwmPeL

NESTED SELECTORS

So far we have looked at **element** selectors, **class** selectors and **id** selectors. If you want to be more precise in your selecting then you can use the **nested selector**.

There is more than one way of selecting nested elements, you can read more here.

The main nested selector is the `descendant` selector.

```
li a {
  text-decoration: none;
}
```

This will only select anchor tags that are descendants of a list-item tag.

</>

CODE ALONG

NESTED SELECTORS

Copy the *nested_selectors* folder in *week2_working_files* into your *FEWD_Week2* folder and open it in Atom.

NORMALISING AND RESETTING CSS

Every browser has a slightly different *default* style sheet which is applied to HTML elements before your own linked CSS styles are applied.

When you want to make sure that a web page looks exactly the same across lots of different browsers, there are two common techniques you can use with css **before** you apply your own styles:

- Reset the browser CSS by linking *reset.css* first
- Normalise the browser CSS by linking *normalize.css* first

RESETTING CSS

Resetting CSS will remove **all** a browser's default styles. You can then add back in only the styles you want, exactly as you want them. This technique gives your browser an extra workload when rendering the page, as it has to remove a lot of styles, but only older browsers are noticeably affected.

You can mitigate against this by only resetting and then styling the elements that you will actually use. NB there are multiple *reset.css* starting points available on the web.

NORMALIZING CSS

Unlike resetting all styles (or whichever ones you use), *normalize.css* works with existing browser styles. It "makes browsers render all elements more consistently and in line with modern standards".

Normalize CSS is updated regularly and resolves specific issues with HTML5 elements and mobile browsers as well as desktop browsers.

NORMALISING AND RESETTING CSS

Some CSS reset links:

- Eric Meyer's CSS Reset
- HTML5-Reset (with other tools)
- Yahoo UI Reset

Some CSS normalize links:

- Nicolas Gallagher's normalize.css
- HTML5 Boilerplate (with other tools)

NORMALISING AND RESETTING CSS

Which to choose?

You should either *normalise* or *reset* on a case by case basis.

Or you can just do some simple normalizing/resetting in your own stylesheet:

```css
* {
    margin: 0;
    padding: 0;
}
```

MODERNIZR

Modernizr is a tool that detects the capabilities of the end-user's browser and then dynamically sets classes on the `<html>` root-level element to indicate whether some browser feature is available or not.

You can then use nested selectors in your CSS to handle whether a feature is available or not. This technique is known as **progressive enhancement**.

https://modernizr.com/docs/#what-is-modernizr

RELAXR LANDING PAGE

Let's have a preliminary look at today's assignment and think about how to start this project. Specifically we should look at the design and style guide, as well as the README.

Copy the *assignment* folder in *week2_working_files* into your *FEWD_Week2* folder, commit these changes and then publish to github.com.

RELAXR LANDING PAGE

You can now read the README rendered in the root of the *assigment* folder on github.com.

You can also click into the *starter_code* folder and click on the *design_guide.md*.

You can see the design "flat" (image) at *starter_code/images/relaxr_landing.jpg*.

You will want to use a background image. Copy the *background_image_examples* folder in *week2_working_files* into your *FEWD_Week2* folder and open it.

DRAW ME A DOM

Let's draw a DOM tree on the whiteboard for how we think we should structure the Relaxr landing page.

FEWD - LAYOUT

AGENDA

- Review
- Divs, Classes and IDs
- HTML5 Structural Elements
- Floats
- Lab Time

REVIEW

What would you like to review?

DIV REVIEW

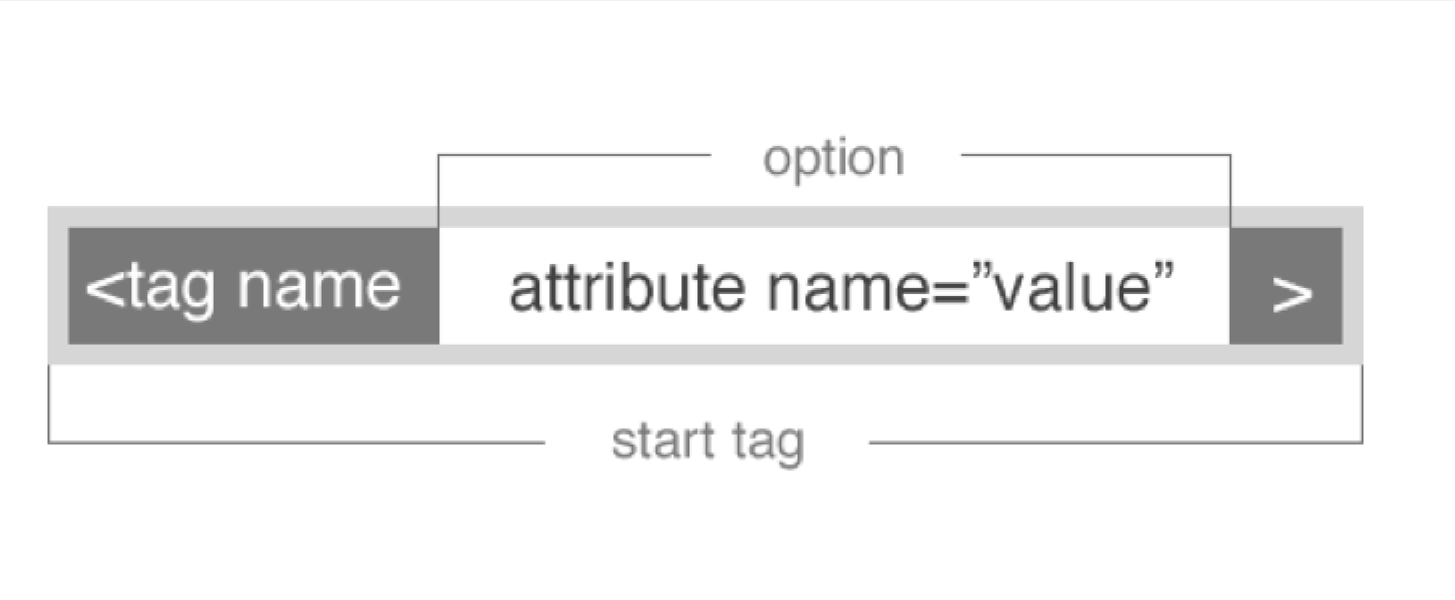
A generic container, `<div>`s are often nested in other `<div>`s

```
<div class="parent">
    <div class="child">Some child content</div>
</div>
```

Have a look at this example: http://codepen.io/wilkom/pen/OyrPzV

CLASS & ID REVIEW

With classes and ids we can target specific elements on a page, so we can manipulate it uniquely.



</> CODE ALONG

CLASS & ID

Copy the *error_message* folder in *week2_working_files* into your *FEWD_Week2* folder and open it in Atom.

CLASS & ID

IDs are unique

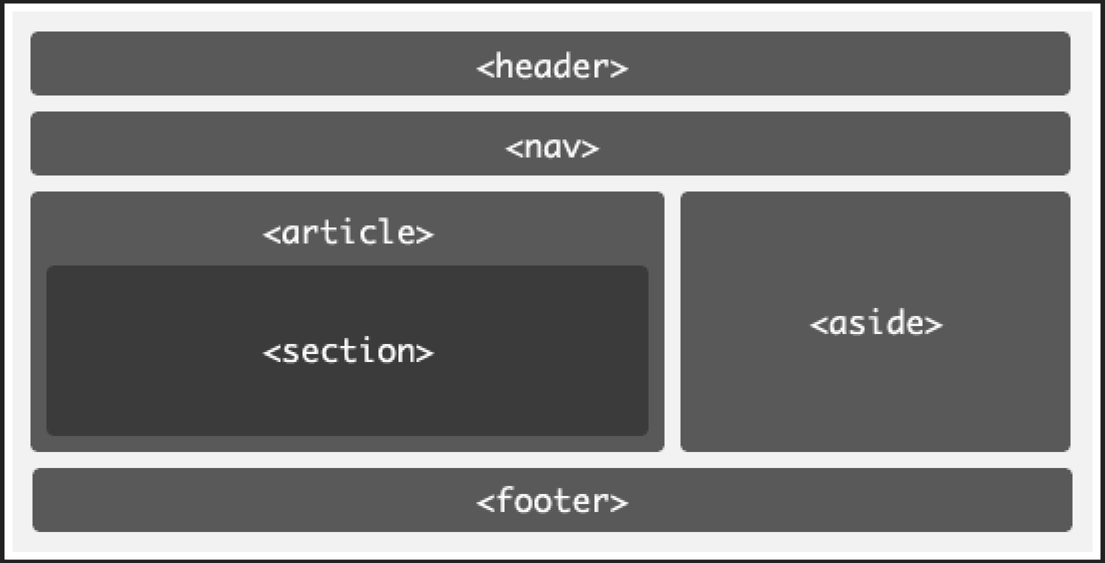
Classes are not unique

How to **select** classes in CSS

```
.className
```

```
#idName
```

HTML5 SIMPLE LAYOUT



```
<body>
     <header>...</header>
     <nav>...</nav>
     <article>
          <section>
               ...
          </section>
     </article>
     <aside>...</aside>
     <footer>...</footer>
</body>
```

HTML5 STRUCTURAL ELEMENTS REVIEW

Adding structure to HTML elements that are related to content layout.

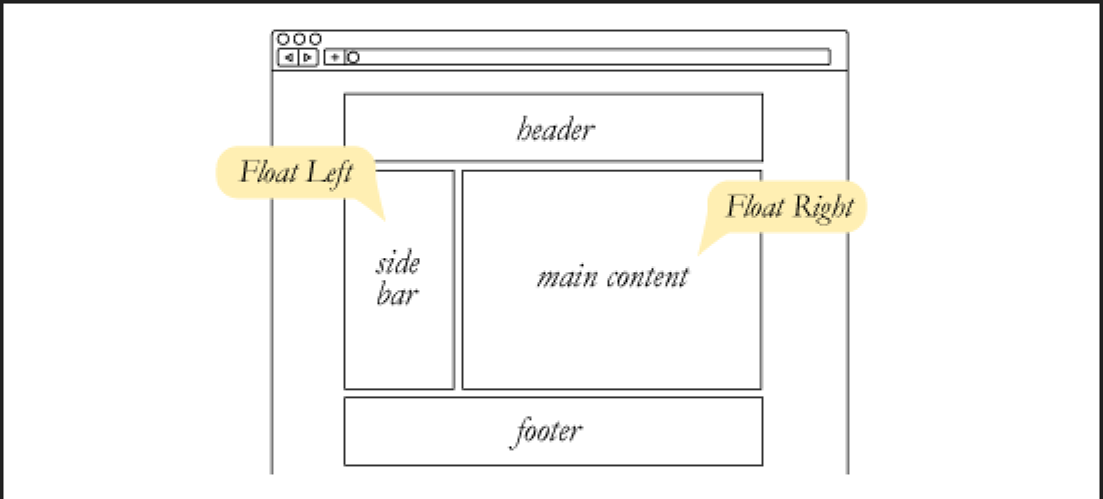
- header
- aside
- section
- footer

Copy the *simple_page_layouts* folder in *week2_working_files* into your *FEWD_Week2* folder and open it in Atom.

Use HTML5 structural elements instead of `<divs>`

FLOATS

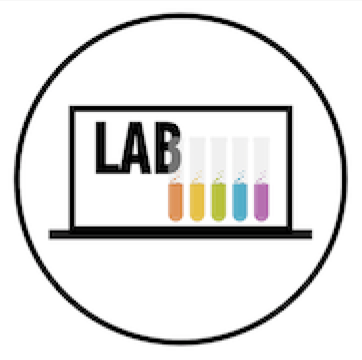
Float is originally intended for making content 'float alongside' other content. It ended up also being used to float containers alongside each other.



Floated layouts are now being replaced by `flexbox`, which we will look at next week.

FLOATS

- You can 'float' elements to the left or right of a parent container.
- Floats are still often used for page layouts - for example to have a sidebar
- You need to use the `clear` property in the style applied to the container of the floated elements. This stops the container *collapsing*, and the float affecting the rest of the page. By convention a style for clearing a float is commonly called a `clearfix`. You can also put a `clearfix` right after a floated element, on the same hierarchical level.



LAYOUT CHALLENGE

Copy the *layout_challenge* folder in *week2_working_files* into your *FEWD_Week2* folder and open it in Atom.