HTML & CSS Level 1: Week 4

June 18 - July 23, 2014

Instructor: Devon Persing

This week

- HTML data tables
- New HTML5 containers
- CSS review and abbreviations
- Layouts 101: CSS **`display`**, **`float`** and **`clear`** properties

HTML data tables

<> What's a `<table>` really for?

- Presenting data in a tabular format
- That's it
- That's all *
- For example:
  - Listings of people, addresses, etc.
  - Financial data
  - Product feature comparisons

*Also HTML emails :(

<> Basic table elements

- **`<table>`** wraps all **table** elements
- **`<tr>`** creates a **row** of table cells
- **`<th>`** creates a **table header** cell for a column *or* a row
- **`<td>`** creates a regular **table data** cell within a row

<> A basic table

```
<table>
   <tr>
      <th>Column 1 Header</th>
      <th>Column 2 Header</th>
   </tr>
   <tr>
      <td>Column 1 Data Cell</td>
      <td>Column 2 Data Cell</td>
   </tr>
</table>
```

<> `<th>` attributes

- For accessibility, it's good practice to use a scope attribute for table header cells:
  - **`scope="col"`** for table headers that describe a column
  - **`scope="row"`** for table headers that describe a row
- Creates an explicit connection for data cells that have multiple headers

<> A table with a header row

```
<table>
    <tr>
        <th scope="col">Column 1 Header</th>
        <th scope="col">Column 2 Header</th>
    </tr>
    <tr>
        <th scope="row">Row 2 Header</th>
        <td>Row 2, Column 2 Data Cell</td>
    </tr>
</table>
```

{ } Styling table elements

- **`background`**, **`border`**, **`margin`**, and **`padding`**, etc. styles can all be applied!
- **`border-spacing`**: space between cells

```
td { border-spacing: 10px 5px; }
```

- **`border-collapse`**: removes spacing and mashes cell borders together

```
td { border-collapse: collapse; }
```

<> Spanning multiple cells

- The **`colspan`** attribute allows a cell to span multiple columns
- The **`rowspan`** attribute allows a cell to span multiple rows
- In general, cells that span only one row or column are recommended

New HTML5 containers

Pre-HTML5 structure

- Previously, HTML only had semantic tags for content (ex.: **`<h1>`**, **`<p>`** and **`<td>`**)
- We faked it for containers by using **`<div>`** elements with semantic-sounding **`id`** attributes (ex.: **`id="header"`**)
- Designed to "futureproof" and create semantic structure for chunks of content

Pocket designed by Michela Tannoia from the Noun Project

<> Major HTML5 containers

- **`<header>`**: header of a container
- **`<nav>`**: navigation links
- **`<main>`**: primary content
- **`<section>`**: a group of related content
- **`<article>`**: what is says on the tin
- **`<aside>`**: supportive, non-primary stuff
- **`<footer>`**: footer of a container

Support for older browsers

- IE9 and other older browsers have no native support for shiny new HTML5 tags
- These browsers can be ~~tricked~~ *gently coaxed* into displaying and styling new HTML5 elements via Javascript
- The most popular method is the **HTML5 shim**: https://code.google.com/p/html5shim/

Installing the HTML5 shim

1. Download the shim zip file and unzip it
2. Find the **`html5shiv.js`** file, and move it to a **`js`** directory in your files
3. Inside the **`<head>`** element of all your pages, add:

```
<!--[if lt IE 9]>
   <script src="js/html5.shiv.js"></script>
<![endif]-->
```

Toolset designed by Calvin Goodman from the Noun Project

{ } While we're in our files...

- The **`<main>`** element is so new (~2013) that our reset styles (~2012) don't have it
- Let's add it to our **`styles.css`** file

<> `<header>`

- Wraps introductory content or navigation
- Can appear in multiple places on the page
- Use it for things like:
  - The global header of a site
  - The header of an article
  - The header of a long section within an article

<> `<nav>`

- Contains major navigation elements
- Can appear in multiple places on the page
- Use it for things like:
  - Global navigation links for a site
  - Pagination links
  - Anything used to get around within a site

<> `<main>`

- Wraps the main content of a page
- Is used **only once per page**
- Use it for things like:
  - A group of blog posts
  - An article with its own header
  - Whatever makes up the main focus of the page

<> `<section>`

- Wraps thematically related content, often with its own heading
- Can appear in multiple places on the page
- Use it for things like:
    - A group of related blog posts
    - A section within an article
    - A sidebar widget with its own header

<> `<article>`

- Wraps standalone texts
- A page might have one, several, or none
- Use it for things like:
  - Individual blog posts
  - Individual news articles
  - Individual comments on other articles

<> `<aside>`

- Wraps non-primary or tangential content
- A page might have one, several, or none
- Use it for things like:
  - A sidebar of related links
  - A pullquote from an article
  - Things that can stand alone but aren't the main content of a page

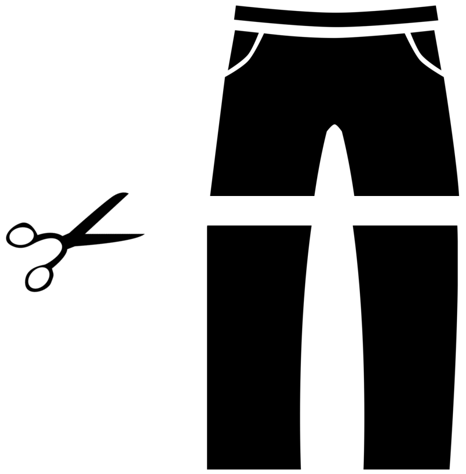
<> `<footer>`

- Wraps any closing information in a container
- A page might have one, several, or none
- Use it for things like:
  - The global footer of a site
  - The footer of an article

Notes on using HTML5 containers

- HTML5 is an experiment in process and documentation!
- Elements will come and go
- When in doubt, use an online resource like http://html5doctor.com/ (these will be more up to date than books)
- If you're not sure, use a **`<div>`**!

CSS abbreviations



Pants designed by Pham Thi Dieu Linh from the Noun Project
Scissors designed by Kelly Ness from the Noun Project

{ } Abbreviated hex colors

```
color: #333333;
/* becomes */
color: #333;


color: #aa0099;
/* becomes */
color: #a09;
```

{ } Abbreviated font styles

```
font-style: italic;
font-variant: small-caps;
font-weight: bold;
font-size: 1em;
line-height: 1.5em;
font-family: Helvetica, sans-serif;

/* becomes */

font: italic small-caps bold 1em/1.5em
Arial, Helvetica, sans-serif;
```

{ } Personally...

- I'm not a huge fan of the font property
- It's hard to remember
- Values for font size and font family are **required**
- Usually you want to set a font-family for a bunch of types of text and font sizes for individual types of text
- But, you can use it!

{ } CSS `background` abbreviations

- **`background-color`** and **`background-image`** and can be combined into **`background`**
- Color is always listed first, then the image

```
body {
background: #eee url("../img/kitten.jpg");
}
```

Going into a neighbor directory

- If you're using images in styles, you might need to go into a neighbor folder
- To go up a level and down into another folder, we use this method:
  - Two dots (`..`) tells the browser to go up a level to the main directory
  - Backslash (`/`) says to get ready to go into a directory again

```
background-image: url("../img/kittens.jpg");
```

{ } More `background`

- You can also add your **`background-repeat`** and **`background-position`** styles:

```
/* a div with a light gray background, and a background
image that doesn't repeat and is positioned in the
bottom right */

div {
   background: #eee url("img/kitten.jpg")
no-repeat bottom right;
}
```

{ } margin & padding abbr.

```
/* clockwise: top, right, bottom, left */

margin-top: 20px;
margin-right: 40px;
margin-bottom: 1em;
margin-left: 35px;

/* becomes */

margin: 20px 40px 1em 35px;
```

{ } margin & padding abbr. cont.

```
/* top/bottom and left/right match? */

padding-top: 20px;
padding-right: 2em;
padding-bottom: 20px;
padding-left: 2em;

/* combine them! */

padding: 20px 2em;
```

{ } border abbreviations

```
border-top-width: 4px;
border-right-width: 2em;
border-bottom-width: 3px;
border-left-width: 5px;
border-style: solid;
border-color: #a00;

/* becomes */

border: 4px 2em 3px 5px solid #a00;
```

Intro to layouts

A brief history of web layouts

- Before CSS, we used **`<table>`** elements to make layouts :(
- With CSS we can use a variety of properties to arrange elements on the screen by adjusting the flow of the page
    - **Pros:** Any content can be displayed anywhere!
    - **Cons:** Any content can be displayed anywhere!

{ } Setup: Centering our page

- Most websites sit in the middle!
- To do this, give your **`<body>`** (or another container that wraps the whole page):
    - a **`width`** value
    - a left *and* right **`margin`** value of **`auto`**

```
body {
   width: 960px;
   margin: 0px auto;
}
```

{ } The `display` property

- Remember block, inline, and inline-block elements?
- You can roll your own with the **`display`** property
- The most common ones are:
  - **`display: block;`**
  - **`display: inline;`**
  - **`display: inline-block;`**

{ } Why use `display`?

- Make a link look like a button
- Add padding and margins to a "naturally" inline element like a **`<span>`**
- Make a list of navigation links horizontal
- Many other uses cases to keep style and content separate

{ } Floating down the river

- Our page is a flowing river of HTML elements
- Elements can be taken out of the river and made to float on it instead
- The page will flow around these elements, like a river flows around boats, tubes, etc.

{ } `float` property

- "Floating" elements is the easiest way to offset content like images, pullquotes, or other tidbits within the flow of a document
- The **`float`** property has three values:

```
float: left;
float: right;
float: none;
```

{ } The simplest column layout

1. Have a parent element (like **`<body>`**) and give it a width value:
2. Give elements you want to be columns **`width`** values that add up to the parent's **`width`** (child A + child B = parent)
3. Make your columns **`float`**
4. Voila!

{ } `clear` property

- HTML river showing up in weird places?
- The **`clear`** property fixes **`float`** and also has three values:

```
clear: left;
clear: right;
clear: both;
```

"Homework"

- Make and style an HTML data table
- Read up on and try out HTML5 containers
- Practice CSS abbreviations
- Practice with floats
- Optional reading:
    - *HTML5 for Web Designers*: Ch. 5
    - *HTML & CSS*: Ch. 15, 17

Next time

- iframes and embedded media
- Web fonts
- CSS pseudo-selectors and pseudo-elements
- Fancier fluid and fixed layouts
- Related topics and lingering questions

Questions? Comments?

- Visit dpersing.github.io/svc for:
  - Class slides
  - Code samples
  - Resources
- Email me: dep@dpersing.com
- Tweet at me: @devonpersing