Divs and Spans

Two new XHTML Elements <div> and <span>

- Provides supporting structures
- Structures are able to be styled
- Shortcuts available to make specifying properties easier

<div>
- Lets you divide your webpage into logical sections or groupings
- Adds structure to a webpage by dividing into logical sections
- Logical sections are a group of related elements on the page
- Logical sections could represent the main content areas
- Logical sections will be enclosed within a <div> element

Adding id to <div>
- Adding an id to <div> enables you to label what the grouping means
- Using the id attribute provides a unique label for the <div>
- Give the cats <div> and id of cats and the dogs <div> an id of dogs
  <div id="cats"> </div>
  <div id="dogs"> </div>

Adding style to div

<#cats {
  background-image: url(cats.jpg);
}</div>
<#dogs {
  background-image: url(dogs.jpg);
}</div>

Creating two rules, one for each <div>
<div>

• Each <div> is selected by an id selector
• Setting a background on the <div> will show through on the elements within that <div>
• Elements in the <div> will also inherit properties from the <div> just as any child element

Benefits to adding more structure to pages

• Allows you to further expose the underlying logical structure of your pages
  – helps viewers to better understand them
  – Helps in maintaining them
• Structure gives you a way to apply style to a section

Adding a header and footer using div

    <div id="header"> </div>
    <div id="cats"> </div>
    <div id="dogs"> </div>
    <div id="footer"/>

Adding structure onto structure

    <div id="header"> </div>
    <div id="pets">
      <div id="cats"> </div>
      <div id="dogs"/>
    </div>
    <div id="footer"/>

Adding structure onto structure

• <div> acts like a container that you can put elements into to keep them all together
• Use <div> to add structure where it has real purpose
• Always keep structure as simple as possible
• Adding to much structure can overly complicate with no real benefit

Adding id to <div>

• Adding an id to <div> enables you to label what the grouping means
• Using the id attribute provides a unique label for the <div>
• Give the cats <div> and id of cats and the dogs <div> an id of dogs
• Add a pets <div> around the cats and dogs <div> and give the pets <div> an id of pets
Adding structure onto structure

```html
<body>
  <div id="header">
    <h1>Cats and Dogs header info</h1>
  </div>
  <div id="pets">
    <div id="cats">
      <h2>Cats section</h2>
      <p>Cats details</p>
    </div>
    <div id="dogs">
      <h2>Dogs section</h2>
      <p>Dogs details</p>
    </div>
  </div>
  <div id="footer">
    <p>Footer info</p>
  </div>
</body>
```

Another example using elixirs...

```html
<div id="elixirs">
  <h2>Weekly Elixir Specials</h2>
  <p><img src="images/yellow.gif" alt="Lemon Breeze Elixir" /></p>
  <h3>Lemon Breeze</h3>
  <p>The ultimate healthy drink, combines herbal botanicals...</p>
  <p><img src="images/chai.gif" alt="Chai Chiller Elixir" /></p>
  <h3>Chai Chiller</h3>
  <p>Not your traditional chai...</p>
</div>
```

Adding a border around the elixirs <div> element

- First, modify your CSS file selecting the elixirs <div> element using its id
- Add a thin solid border with a aquamarine hex color

```css
#elixirs {
  border-width: thin;
  border-style: solid;
  border-color: #007e7e;
}
```

Adding more style

We want to add the following to our elixer <div>

- Make the width narrower
- Add padding
- Add a background image
- Adjust text alignment
- Add text line heights
- Add heading colors

```css
#elixirs {
  border-width: thin;
  border-style: solid;
  border-color: #007e7e;
  width: 200px;
}
```
width: property

- Specifies the width of the element’s content area only
- Setting width on the elixirs <div>
- Content in the elixirs <div> will be 200 pixels wide
- Any the elements nested in the <div> will fit inside this width

```css
elixirs {
  border-width: thin;
  border-style: solid;
  border-color: #007e7e;
  width: 200px;
}
```

Determining the entire width

- The width of the entire element would have to include content area + padding + border + margin
- All added together represents the width of the entire element
- If you have no padding, border, or margin but have defined the width of the <div> as 200, then the width of the entire box would also be 200 pixels

Remember the box model

![Box Model Diagram]

Adding padding, margin, align, background

Specifying additional elixir attributes

```css
elixirs {
  border-width: thin;
  border-style: solid;
  border-color: #007e7e;
  width: 200px;
  padding-right: 20px;
  padding-bottom: 20px;
  padding-left: 20px;
  margin-left: 20px;
  text-align: center;
  background-image: url(images/drink.gif);
  background-repeat: repeat-x;
}
```

Adding padding, margin, align, background

- Adding a padding left, right, bottom 20 pixels
- Adding a left margin of 20 pixels
- Aligning all in the center
- Adding a background image, repeats only horizontally

```css
text-align: center;
```

- Text-align property will also affect the alignment of the images
- Doesn’t only align text
- Aligns all inline content in a block element
- By setting the property on the <div> block element all inline content is nicely centered
- Text-align works on any kind of inline element
- Should be set on block elements only
- Has no effect if used directly used on inline elements like <img>
text-align: center;

- All the text inside the `<div>` element is in nested block elements
- Block elements inherit the text-align property from the `<div>`
- Headings and paragraphs since they are block elements inherit the text-align value of "center" and then align their own content to center
- Not all properties are necessarily inherited

Modifying headings only within the elixir `<div>`

Decendant selectors

Specifying a selector for example “select an `<h2>` element, but only if it's inside an elixirs `<div>`”

A way to select descendants

Selecting elements that descend from certain elements

Specifying a selector…

```html
div h2 {
  color: black;
}
```

...select an `<h2>` element, but only if it's inside an elixirs `<div>`

A way to select descendants

Select elements that descend from certain elements

- Div parent element
- h2 descendant name
- Rule says to select any `<h2>` that is a descendant of a `<div>`
- Rule selects the h2 within the `<div>` elixir

```html
div h2 {
  color: black;
}
```

A way to select descendants

Problems with specifying by `<div>`

Applies styling to all `<div>` with a h2 element

```html
div h2 {
  color: black;
}
```

A way to select descendants

Adding an id to the `<div>`

```html
#elixirs h2 {
  color: black;
}
```
**Ways to select descendants specifically**

Adding an id to the `<div>`

- Selects any `<h2>` that is a descendant of an element with the id "elixirs"
- Parent element is the element with the id elixirs
- Any descendant of elixirs that is `<h2>`

```
#elixirs h2 {
  color: black;
}
```

**More ways to select descendants**

To specify an `<h2>` that is the child of a `<blockquote>` within elixirs

Selects and `<h2>` element that descend from `<blockquote>` that descend from an element with an id of "elixirs"

```
#elixirs blockquote h2 {
  color: black;
}
```

**Example**

- Setting the `<h2>` heading to black within the "elixirs"
- Setting the `<h3>` headings to red within the "elixirs"
- Using the descendant selectors to target just the `<h2>` and `<h3>` elements in the elixirs `<div>`

```
#elixirs h2 { 
  color: black;
}

#elixirs h3 { 
  color: red;
}
```

**Example**

- Adds black and red headings in the elixirs section
- Doesn’t effect color being used for `<h2>` headings in the rest of the main page

```
#elixirs h2 { 
  color: black;
}

#elixirs h3 { 
  color: red;
}
```

**Setting the line height**

- Adding the line-height property on the entire `<div>` will allow it be inherited by all elements
- Line height is 1 times its inherited font size

```
#elixirs {
  Line-height: 1em;
}
```

- If you want to modify line-height more specifically than use a number

```
#elixirs {
  Line-height: 1;
}
```

**Setting the line height**

- Using a number
- Adding a line height of 1 to the elixirs `<div>` changes the line height of each element in it
- Each element within the elixirs `<div>` to have a line-height of 1 times its own font-size
- Rather than the font-size of the elixirs `<div>`

```
#elixirs {
  line-height: 1;
}
```
**Shortcuts for setting the different property values on the background of an element**

Padding old way

```css
padding-top: 0px;
padding-right: 20px;
padding-bottom: 30px;
padding-left: 10px;
```

**Shortcuts for setting the different property values on the background of an element**

Padding new way

```css
padding: 0px 20px 30px 10px;
```

**Shortcuts for setting the different property values on the background of an element**

Padding shorthand for all sides 20 px

```css
padding: 20px;
```

**Shortcuts for setting the different property values on the background of an element**

Margins old way

```css
margin-top: 0px;
margin-right: 20px;
margin-bottom: 30px;
margin-left: 10px;
```

**Shortcuts for setting the different property values on the background of an element**

Margins new way

```css
margin: 0px 20px 30px 10px;
```

**Shortcuts for setting the different property values on the background of an element**

Margins shorthand for all sides 20 px

```css
margin: 20px;
```
Shortcuts for setting the different property values on the background of an element

Margins new way
top and bottom 0px and left and right 20px

```css
margin: 0px 20px;
```

Shortcuts for setting the different property values on the background of an element

Border old way

```css
border-width: thin;
border-style: solid;
border-color: #007e7e;
```

Border new way

```css
border: thin solid #007e7e;
```

Additional border shorthand examples

```css
border: solid thin;
border: solid thin #007e7e;
border: #007e7e solid;
border: #007e7e solid thin;
border: solid;
```

Background shorthand examples

Background old way

```css
background-color: white;
background-image: url(images/drink.gif);
background-repeat: repeat-x;
```

Background new way

```css
background-color: white url(images/drink.gif) repeat-x;
```
Background shorthand examples

Font old way

Font-size: small;
Font-family: verdana, helvetica, sans-serif;
Line-height: 1.6em;

Font new way

Font: small/1.6em verdana, helvetica, sans-serif;

Adding <span>

Song and Artist shown below need separate styling

<ul>
  <li>At Last, Etta James</li>
  <li>Georgia On My Mind, Ray Charles</li>
  <li>Soul Man, The Blues Brothers</li>
</ul>

Adding <span>

Song and Artist now have a class

<ul>
  <li><span class="song">At Last</span>, <span class="artist">Etta James</span></li>
  <li><span class="song">Georgia On My Mind</span>, <span class="artist">Ray Charles</span></li>
  <li><span class="song">Soul Man</span>, <span class="artist">The Blues Brothers</span></li>
</ul>

The style of an <a> element changes depending on its state

States

- link (never clicked on, unvisited)
- visited (visited/clicked on)
- hover (mouse over but not clicked on)
Adding Style to the `<a>` element

Add the following to the stylesheet

```css
a:link {
    color: green;
}

a:visited {
    color: red;
}

a:hover {
    color: yellow;
}
```

Pseudo-class

Acts like a class but it isn't a real class
- You can style pseudo-classes not necessary to type the classes into XHTML code
- Example of a pseudo class:
  ```css
  a:link
  a:visited
  a:hover
  ```
  Browser interprets each of these as a pseudo-class

Adding a descendant selector combined with a pseudo-class

Add the following to the stylesheet

```css
#elixirs a:link {
    color: green;
}

#elixirs a:visited {
    color: red;
}

#elixirs a:hover {
    background: blue
    color: yellow;
}
```

Descendant selector with a pseudo-class

- First selector selects any unvisited `<a>` element nested in an element with the id “elixirs”, styles only links inside elixirs, sets color of text to green
- Second selector selects visited `<a>` element… sets color of text to red
- Third selector selects hover `<a>` element… sets background to blue and color of text to yellow

```
#elixirs a:link { color: green; }
#elixirs a:visited { color: red; }
#elixirs a:hover { background: blue
    color: yellow; }
```

The Cascade in CSS

- Refers to the way styles cascade down from multiple style sheets
- The most specific styling is applied to each element
- Basically three possible style sheet definitions
  1. Styles defined by the author or creator
  2. Styles defined by the person viewing or reading
  3. Styles defined by the browser
Cascade in CSS

- Browser determines which to display by the priority, author has the highest followed by viewer and then browser