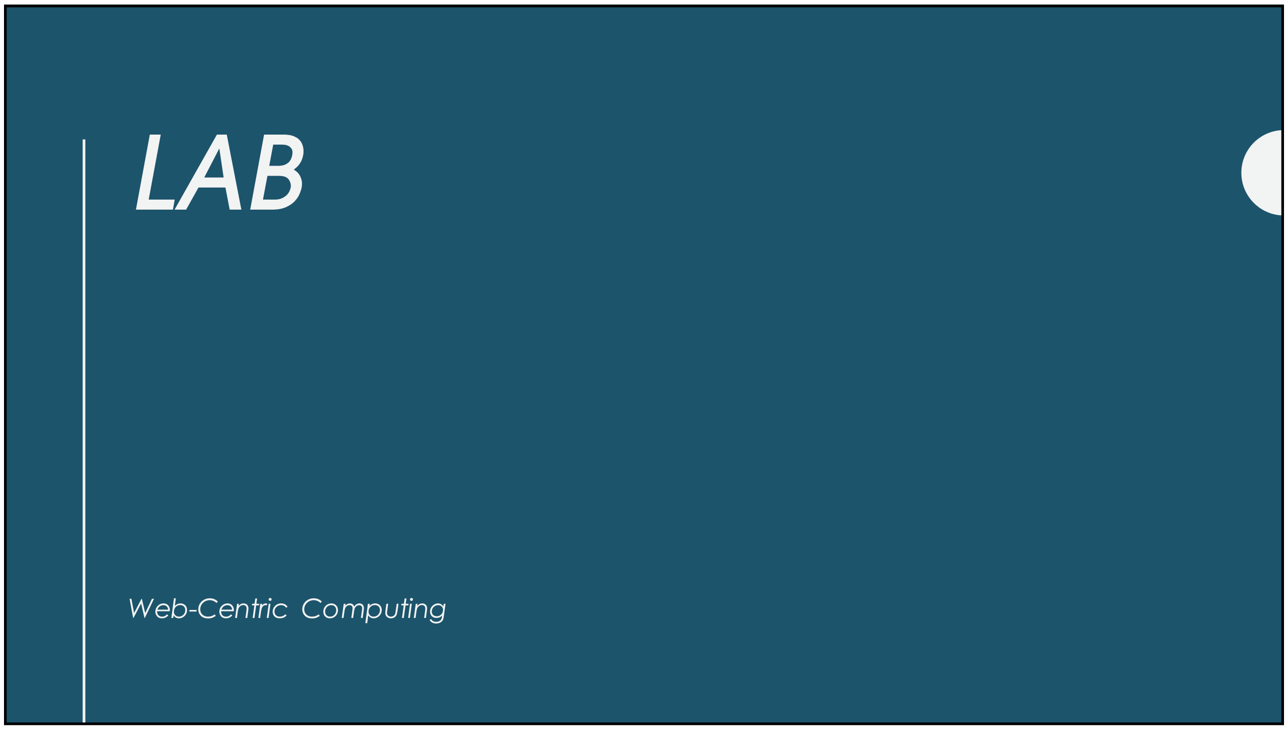
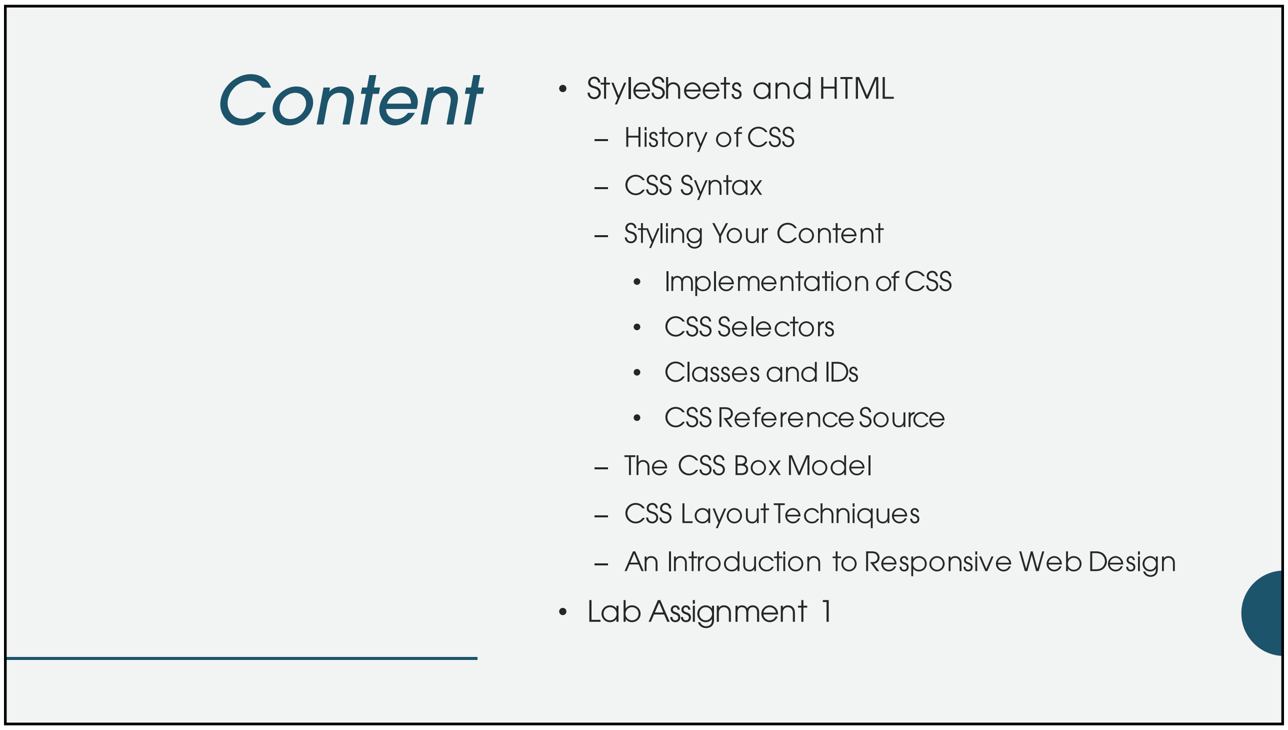


LAB

Web-Centric Computing



Content

- StyleSheets and HTML
  - History of CSS
  - CSS Syntax
  - Styling Your Content
    - Implementation of CSS
    - CSS Selectors
    - Classes and IDs
    - CSS Reference Source
  - The CSS Box Model
  - CSS Layout Techniques
  - An Introduction to Responsive Web Design
- Lab Assignment 1

An introduction

STYLESHEETS AND HTML

Stylesheets and HTML

History of CSS

- It was first added to HTML in 1996 as part of **HTML 4.0**
- Benefits:
  - Simplifies HTML code
  - Pages load much quicker
  - Easier and quicker to make changes to web pages
  - Simplifies site maintenance
  - Accessibility
  - Usability

Stylesheets and HTML

CSS Releases

- CSS 1:
  - Released in 1996, focuses on the **styling of text**
- CSS 2:
  - Released in 1998, focuses on the **page layout and positioning elements, and media descriptors**
- CSS 2.1:
  - Released in 2011, its main focus was to speed up the release of CSS 3 by including a few properties from that release
- CSS 3:
  - Provided a new modular approach to how properties were released, thus ensuring browser support for necessary modules

Stylesheets and HTML

CSS Syntax

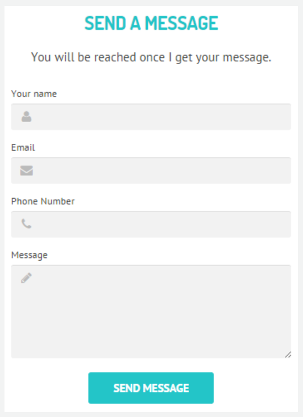
- To write a simple **CSS rule**:
  - Select the element to apply a style to (i.e., **selector**)
  - Specify the **property** to style
  - Set a **value** for that **property**
  - CSS uses **property: value;** pairs

```
selector  {
      property:  value;
      property:  value;
      property:  value;
        }
```

Stylesheets and HTML

Styling Your Content

- CSS helps you tell your browser how the HTML content, a user sees, should be displayed (i.e., how to **render** HTML tags)



SEND A MESSAGE

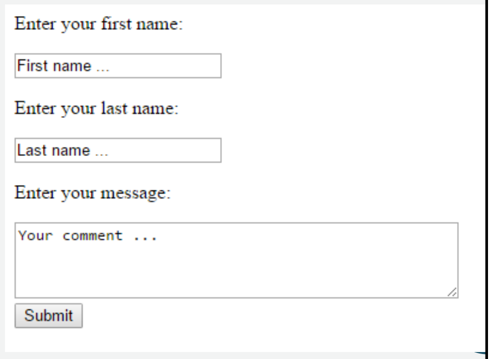
You will be reached once I get your message.

Your name

Email

Phone Number

Message



Enter your first name:

First name ...

Enter your last name:

Last name ...

Enter your message:

Your comment ...

Stylesheets and HTML

Styling Your Content

- StyleSheets can be contained
  - Within an HTML element (i.e., **Inline CSS**)
  - Within the <head> element (i.e., **Embedded CSS**),
  - OR can be kept completely separate in a CSS file (i.e., **External CSS**)

Stylesheets and HTML

Styling Your Content

- Inline CSS

```
<p style="color:red;">This is a paragraph.</p>
<p style="color: blue;text-decoration:underline">This is a paragraph.</p>
<p style="width:300px;text-align:center">This is a paragraph.</p>
```

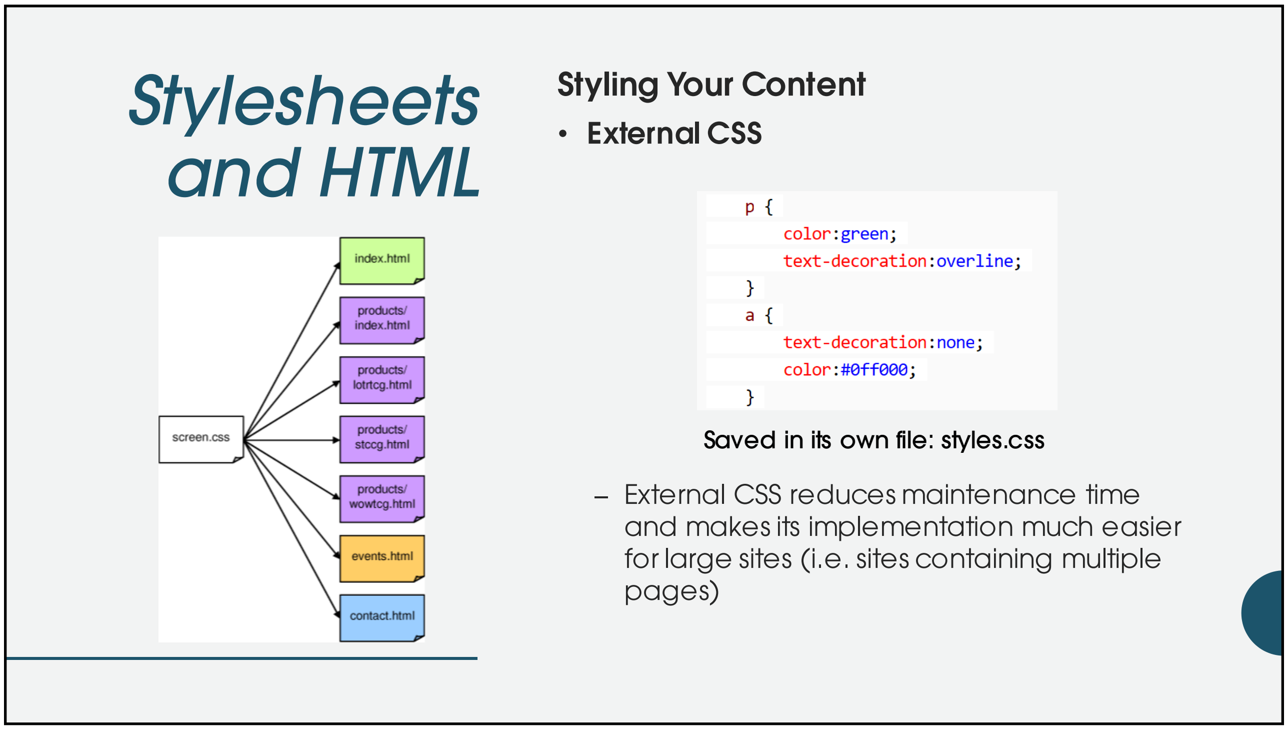
Stylesheets and HTML

Styling Your Content

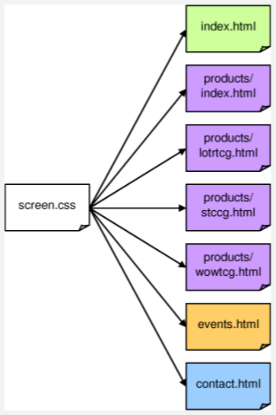
- Embedded CSS

```
<style>
    p {
        color:green;
        text-decoration:overline;
    }
    a {
        text-decoration:none;
        color:#0ff000;
    }
</style>
```

– We could also use **@import** inside the **<style>** element
  e.g., `@import url(css/style.css);`



Stylesheets and HTML



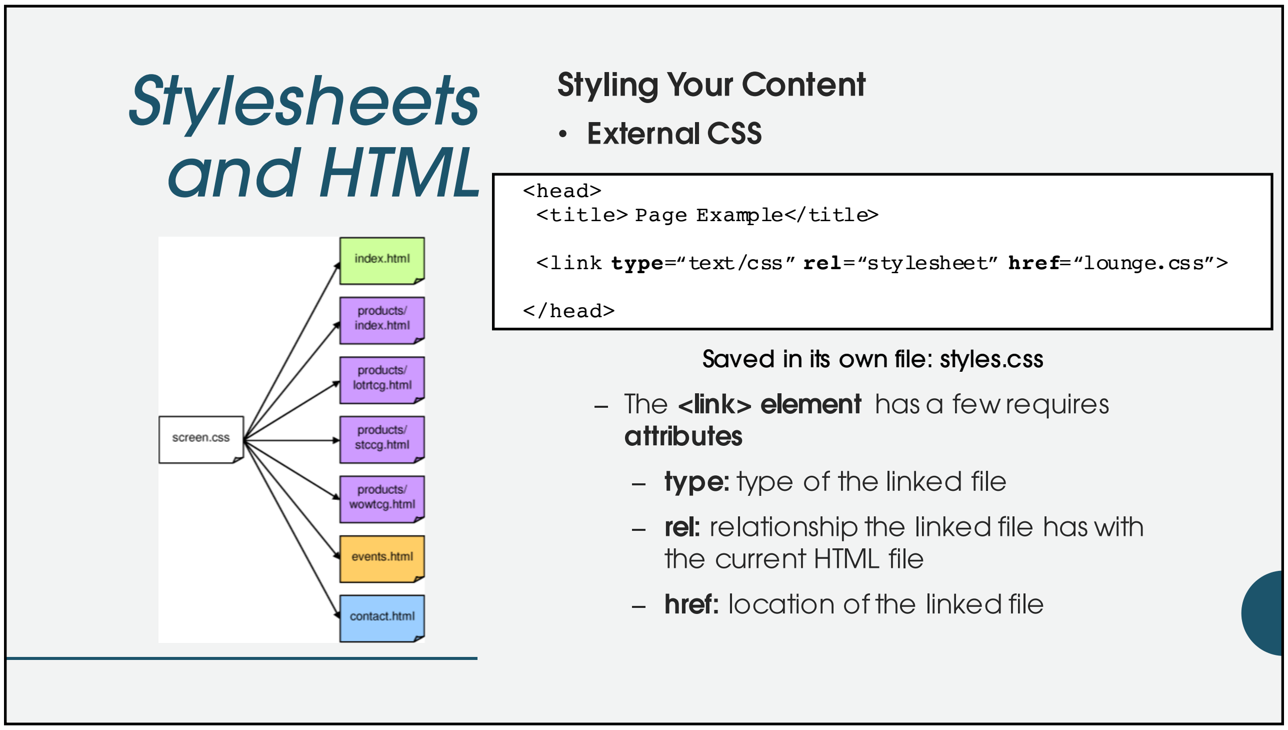
Styling Your Content

- **External CSS**

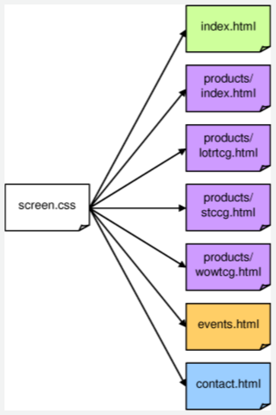
```
p {
    color:green;
    text-decoration:overline;
}
a {
    text-decoration:none;
    color:#0ff000;
}
```

Saved in its own file: styles.css

– External CSS reduces maintenance time and makes its implementation much easier for large sites (i.e. sites containing multiple pages)



Stylesheets and HTML



Styling Your Content

- **External CSS**

```
<head>
 <title> Page Example</title>

 <link type="text/css" rel="stylesheet" href="lounge.css">

</head>
```

Saved in its own file: styles.css

– The **<link> element** has a few requires **attributes**

  – **type:** type of the linked file

  – **rel:** relationship the linked file has with the current HTML file

  – **href:** location of the linked file

Implementation

STYLESHEETS AND HTML

Stylesheets and HTML

Styling Your Content

- **CSS Selectors**
  - The **Universal Selector** selects all the elements in a page

```
* {
    color:  green;
    font-size:  20px;
    line-height:  25px;
}
```

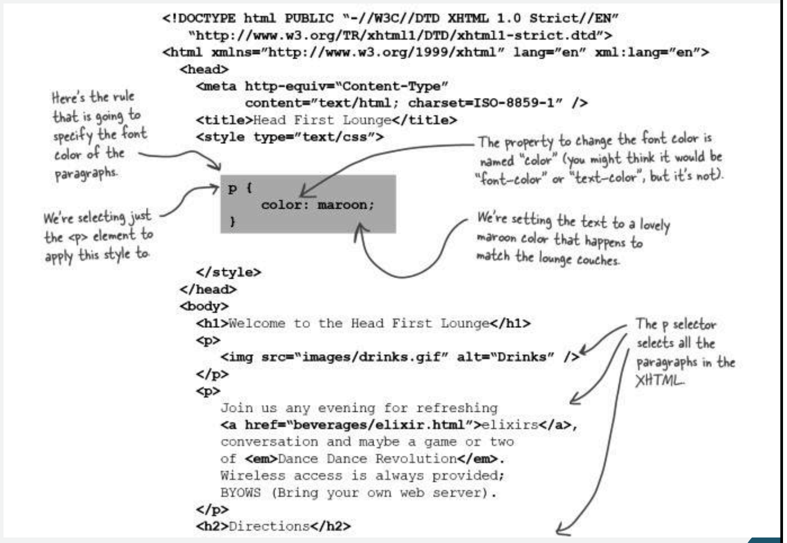
Stylesheets and HTML

Styling Your Content

- CSS Selectors
  - The **Element Type selector** must match one or more HTML elements of the same name

```
/* Rule applies to all unordered lists */

ul {
    color: green;
    font-size: 20px;
    line-height: 25px;
}
```

Stylesheets and HTML



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
  <head>
    <meta http-equiv="Content-Type"
          content="text/html; charset=ISO-8859-1" />
    <title>Head First Lounge</title>
    <style type="text/css">

      p {
          color: maroon;
      }

    </style>
  </head>
  <body>
    <h1>Welcome to the Head First Lounge</h1>
    <p>
      <img src="images/drinks.gif" alt="Drinks" />
    </p>
    <p>
      Join us any evening for refreshing
      <a href="beverages/elixir.html">elixirs</a>,
      conversation and maybe a game or two
      of <em>Dance Dance Revolution</em>.
      Wireless access is always provided;
      BYOWS (Bring your own web server).
    </p>
    <h2>Directions</h2>
```

Here's the rule that is going to specify the font color of the paragraphs.

We're selecting just the <p> element to apply this style to.

The property to change the font color is named "color" (you might think it would be "font-color" or "text-color", but it's not).

We're setting the text to a lovely maroon color that happens to match the lounge couches.

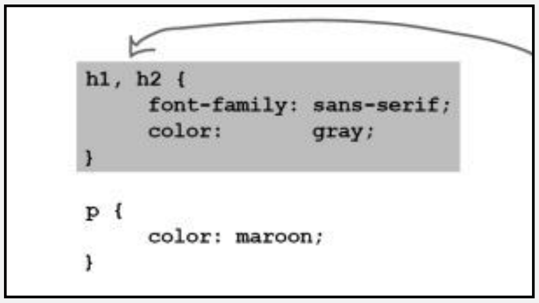
The p selector selects all the paragraphs in the XHTML.

E. Watrall and J. Siarto, Head First Web Design, Sebastopol: O'Reilly Media, 2009.

Stylesheets and HTML

Styling Your Content

- **Grouping CSS Selectors**



```
h1, h2 {
      font-family: sans-serif;
      color:       gray;
}

p {
      color: maroon;
}
```

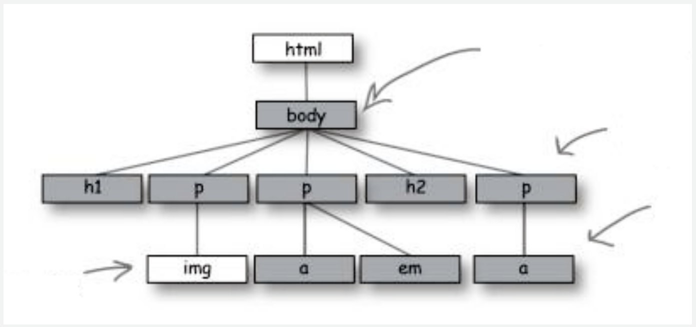
  - To increase efficiency, we can group identical rules for different elements

E. Watrall and J. Siarto, Head First Web Design, Sebastopol: O'Reilly Media, 2009.

Stylesheets and HTML

Styling Your Content

- **The Cascading in CSS**



  - If we write a rule for the **body** element, then it will affect all of the elements contained within it (i.e., parent and child)

E. Watrall and J. Siarto, Head First Web Design, Sebastopol: O'Reilly Media, 2009.

Stylesheets and HTML

Styling Your Content

- **CSS Selectors**
  - The **Descendant Combinator** allows you to combine more than one selector to increase **specificity**

```
/* Rule applies to all elements with .box
   nested inside #container */

#container .box {
    float: left;
    padding-bottom: 15px;
}
```

  - In this case the nested element **does not** have to be an immediate child (i.e., it could be wrapped by another element)

L. Lazaris, Jump Start CSS. Australia: Sitepoint Pty. Ltd., 2013.

Stylesheets and HTML

Styling Your Content

- **CSS Selectors**
  - The **Child Combinator** allows you to combine more than one selector, similarly to the Descendant Combinator

```
/* Rule applies to the first element with
.box nested inside #container */

#container > .box {
    float: left;
    padding-bottom: 15px;
}
```

  - In this case the nested element **does** have to be an **immediate child**

L. Lazaris, Jump Start CSS. Australia: Sitepoint Pty. Ltd., 2013.

Stylesheets and HTML

Styling Your Content

- **CSS Selectors**
  - The **General Sibling Combinator** allows you to match elements based on sibling relationships.

```
<h2>Title</h2>
<p>Paragraph  example.</p>
<p>Paragraph  example.</p>
<p>Paragraph  example.</p>
<div  class="box">
<p>Paragraph  example.</p>
</div>
```

```
/* Rule  applies  to  only  sibling  elements,
i.e.,  besides  each  other  in  the  HTML  */

h2 ~ p {
    padding-bottom:  15px;
}
```

  - In this case the nested element **does** have to be an **immediate child**

L. Lazaris, Jump Start CSS. Australia: Sitepoint Pty. Ltd., 2013.

Stylesheets and HTML

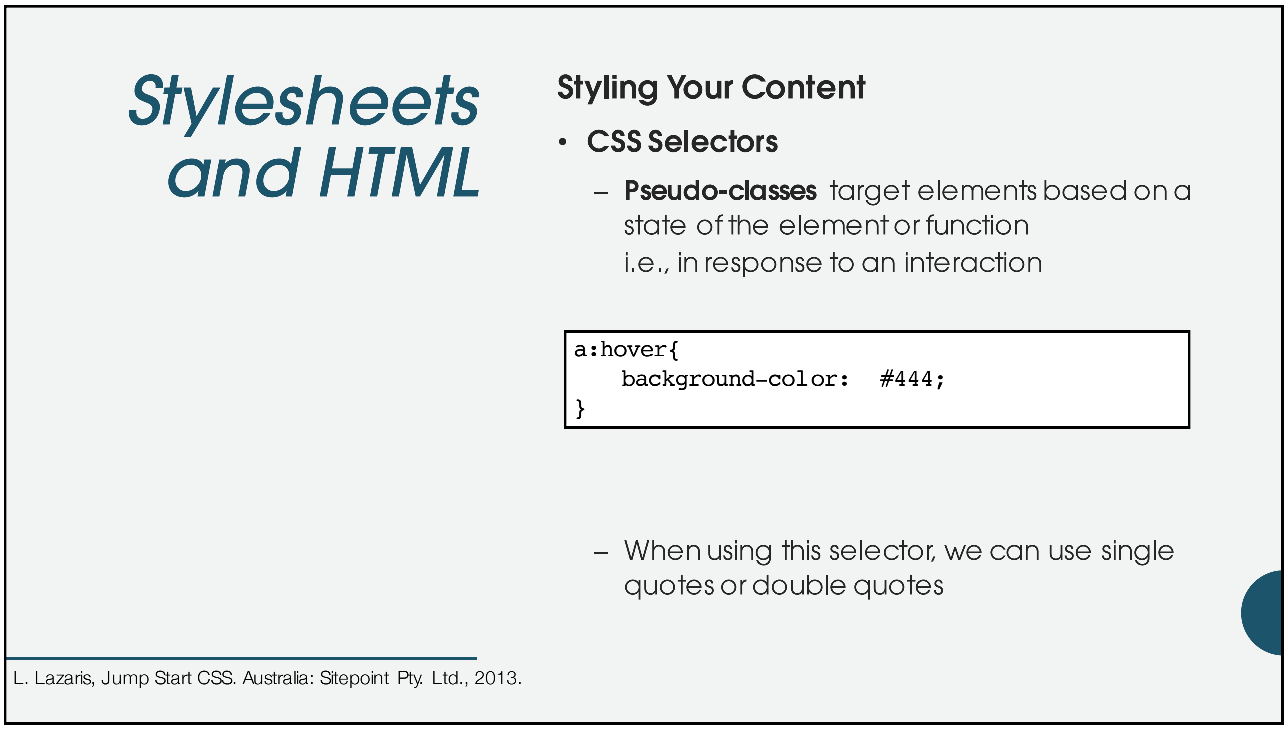
Styling Your Content

- **CSS Selectors**
  - The **Adjacent Sibling Combinator,** similar to the General Sibling Combinator, but the target element must be an **immediate sibling**

```
<h2>Title</h2>
<p>Paragraph  example.</p>
<p>Paragraph  example.</p>
<p>Paragraph  example.</p>
<div  class="box">
<p>Paragraph  example.</p>
</div>
```

```
/* Rule  applies  to  only  immediate  sibling
elements  */

h2 + p {
    padding-bottom:  15px;
}
```

L. Lazaris, Jump Start CSS. Australia: Sitepoint Pty. Ltd., 2013.



Stylesheets and HTML

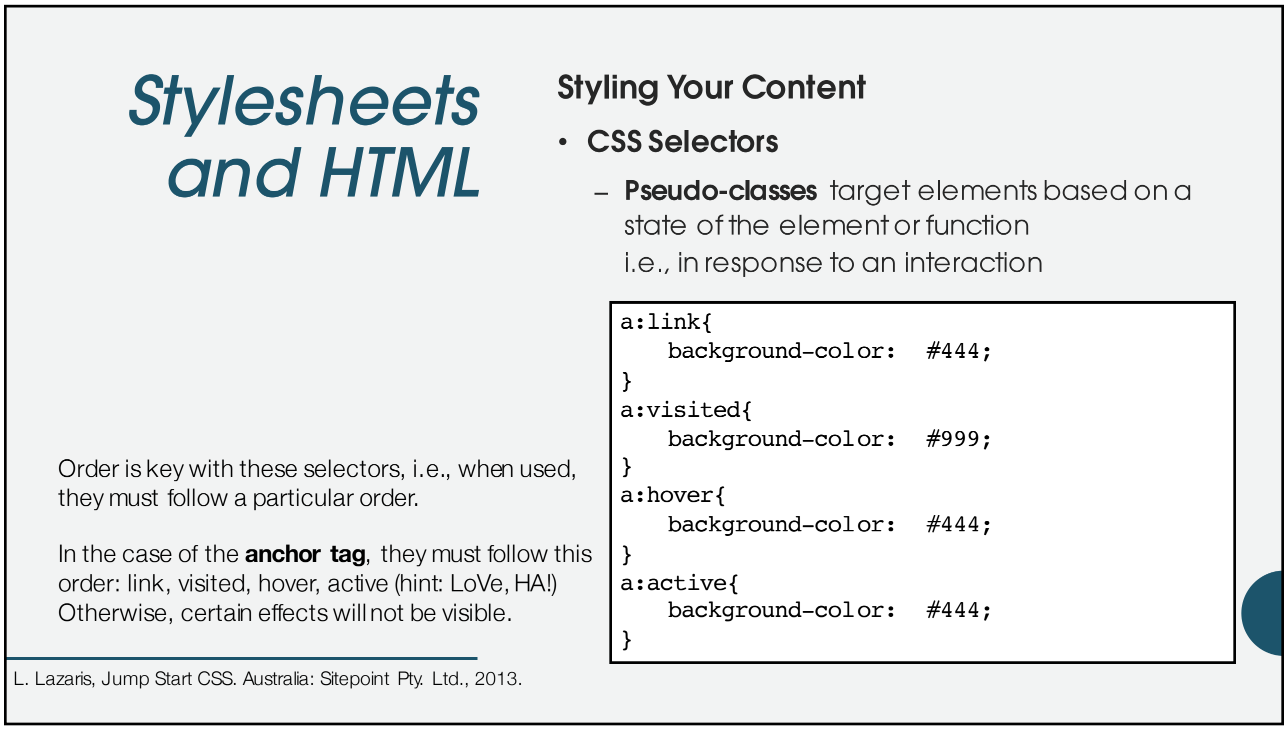
Styling Your Content

- **CSS Selectors**
  - **Pseudo-classes** target elements based on a state of the element or function
    i.e., in response to an interaction

```
a:hover{
    background-color:  #444;
}
```

  - When using this selector, we can use single quotes or double quotes

L. Lazaris, Jump Start CSS. Australia: Sitepoint Pty. Ltd., 2013.



Stylesheets and HTML

Styling Your Content

- **CSS Selectors**
  - **Pseudo-classes** target elements based on a state of the element or function
    i.e., in response to an interaction

Order is key with these selectors, i.e., when used, they must follow a particular order.

In the case of the **anchor tag**, they must follow this order: link, visited, hover, active (hint: LoVe, HA!) Otherwise, certain effects will not be visible.

```
a:link{
    background-color:  #444;
}
a:visited{
    background-color:  #999;
}
a:hover{
    background-color:  #444;
}
a:active{
    background-color:  #444;
}
```

L. Lazaris, Jump Start CSS. Australia: Sitepoint Pty. Ltd., 2013.

Stylesheets and HTML

Styling Your Content

- **CSS Selectors**
  - **Pseudo-elements** insert an imaginary

```
.container:before=“text”]  {
    content:  "";
    display:  block;
    width:  50px;
    height:  50px;
    background-color:   #141414;
}
```

  - i.e., an imagiary element is added before the **.container**
  - When using this selector, we can use single quotes or double quotes

L. Lazaris, Jump Start CSS. Australia: Sitepoint Pty. Ltd., 2013.

CSS Classes and IDs

STYLESHEETS AND HTML

Stylesheets and HTML

CSS Classes and IDs

- Classes
  - Can be applied to any element using the **class attribute**
  - Can be applied multiple times in a page
  - In a CSS file, we use the period to specify a class

The CSS

```
.prices {
    font-weight: bold;
  }
```

The HTML

```
<h2 class="prices"> prices </h2>

<p class="prices"> more prices </p>
```

Stylesheets and HTML

CSS Classes and IDs

- IDs
  - Can be applied to any element using the **id attribute**
  - Can be applied only once per page
  - In a CSS file, we use the number sign to specify an id

The CSS

```
#prices {
    font-weight: bold;
  }
```

The HTML

```
<h2 id="prices"> prices </h2>
```

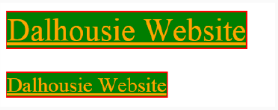
E. Watrall and J. Siarto, Head First Web Design, Sebastopol: O'Reilly Media, 2009.

Stylesheets and HTML

CSS Classes and IDs

- An example of using **Classes** and **ID**s to create custom CSS selectors

```
<style>
    .firstName {
        color:orange;
        background-color:green;
        width:300px;
        border:1px solid red;
    }
    #dalLink {
        font-size:x-large;
    }
</style>
<a href="www.dal.ca" class="firstName" id="dalLink">Dalhousie Website</a><br />
<a href="www.dal.ca" class="firstName">Dalhousie Website</a>
```



Dalhousie Website

Dalhousie Website

The Mozilla Developer Network Reference Source

CSS REFERENCE SOURCE

CSS Reference Source

Need more information on CSS?

- Check out the following links from the Mozilla Developer Network:
  - Getting Started with CSS https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Getting_started
  - CSS Reference https://developer.mozilla.org/en-US/docs/Web/CSS/Reference
  - A Collection of CSS Demos https://developer.mozilla.org/en-US/docs/Web/Demos_of_open_web_technologies

CSS Reference Source

Dealing with older browsers that don't support HTML5?

- You can try adding the following script link to the **head** section of a page

```
<!--[if lt IE 9]>
<script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>
<![endif]-->
```

Adding Some Content to Our Form

A CSS WARM-UP EXERCISE

CSS: A Front-End Warm-up

Exercise 1 (10mins)

- Add some content to the HTML form page you created in our last lab
  - Add a **header section** with a navigation bar with 3 links and a logo
  - In the **main content section,** where your form is, add a few paragraphs, you may use dummy text (i.e., lorem ipsum)
  - In the **footer section** add a small footer link section and copyright info

Note: *ensure you use W3C compliant HTML*

*Adding Some Instructions to the **<head> Section***

A CSS WARM-UP EXERCISE

CSS: A Front-End Warm-up

Exercise 2 (5mins)

- Add some instructions to your **<head> Section**
  - Add some **<meta name=" " content=" ">** tags to provide a description of your work (e.g., author name, description
  - Add a link to the JavaScript script needed to add HTML5 support for browsers that do not support it **(hint:** check the *CSS Reference Source* slides)
  - Download the **normalize.css Stylesheet** from BrightSpace and properly **link** it to your HTML file
  - Create a new file called **style.css** and properly **link** it to your HTML file

Note: *ensure you use W3C compliant HTML*

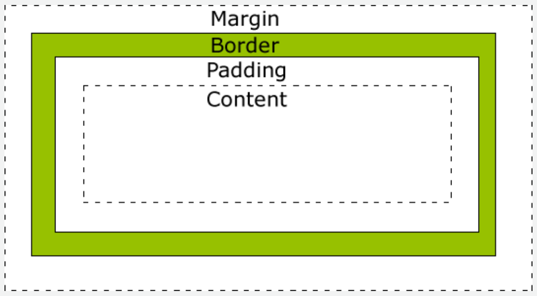
Layout Techniques

STYLESHEETS AND HTML

Stylesheets and HTML

The CSS Box Model

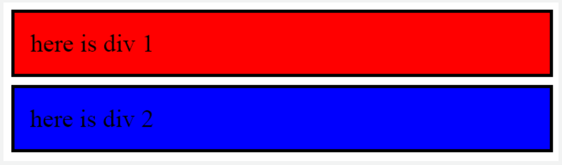
- Every HTML element, with the use of CSS, can create a Box



Stylesheets and HTML

The CSS Box Model

- For example, here is how we create some boxes



```
<div class="div1">
    here is div 1
</div>
<div class="div2">
    here is div 2
</div>
```

```
.div1 {
    width: 320px;
    padding: 10px;
    border: 2px solid black;
    margin: 5px ;
    background-color:blue;
}
```

```
.div2 {
    width: 320px;
    padding: 10px;
    border: 2px solid black;
    margin: 5px ;
    background-color:blue;
}
```

StyleSheets and HTML

Layout Techniques

- HTML Elements can be **block** or **inline**
  - **Block** elements include **<div>, <p>, <section>, <ul>,** to name a few
  - **Block** elements are more structural and tend to be used for layout purposes
  - Unless given a **width**, block elements tend to span the entire width of a browser window

Developers can use the **display property** to force an element to be displayed as **block**

```
a {
   display: block;
}
```

StyleSheets and HTML

Layout Techniques

- HTML Elements can be **block** or **inline**
  - **Inline** elements include **<span>, <strong>, <em>, <a>,** to name a few
  - **Inline** elements are nested inside block elements and flow in the same context as the text, and tend to only hold text or other inline elements

Developers can use the **display property** to force an element to be displayed as **inline**

```
a {
   display: inline;
}
```

StyleSheets and HTML

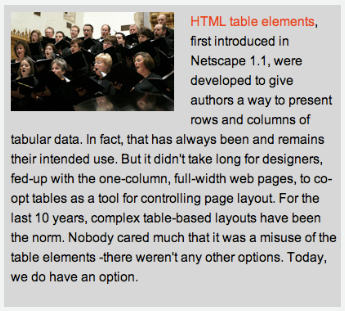
Layout Techniques

- HTML Elements can be **block** or **inline**
  - There may be times when we want an element to be both inline and block

```
.example {
   display: inline-block;
}
```

  - Using the **inline-block** value, allows us to have an element be subject to text-based CSS and flow with the text, as well as accept width, height, and margin values like a block element would

StyleSheets and HTML



HTML table elements, first introduced in Netscape 1.1, were developed to give authors a way to present rows and columns of tabular data. In fact, that has always been and remains their intended use. But it didn't take long for designers, fed-up with the one-column, full-width web pages, to co-opt tables as a tool for controlling page layout. For the last 10 years, complex table-based layouts have been the norm. Nobody cared much that it was a misuse of the table elements -there weren't any other options. Today, we do have an option.

Layout Techniques

- The CSS **Float Property**
  - Used to wrap text around images or multiple column layouts
  - When using the **float property**, the width property **must** be used

```
.example {
   float: left;
   width: 40%;
}
```

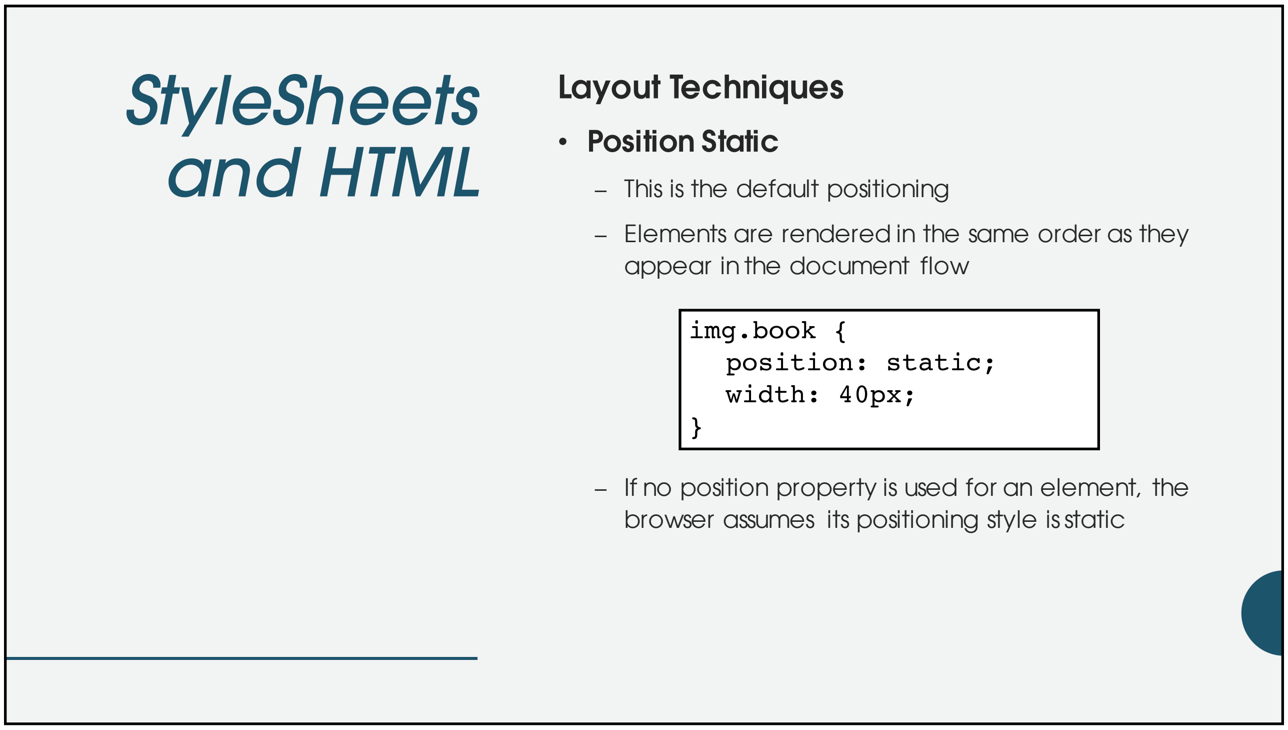
  - The **float property** can take three values: none, left, right
  - A float must be **cleared** for the rest of the document to follow, this technique is known as **clearfix**

```
.example:after {
   clear: both;
}
```

StyleSheets and HTML

Layout Techniques

- The CSS **Position Property**
  - Useful for aligning elements in a precise way
  - There are **four** positioning values, or types of positioning



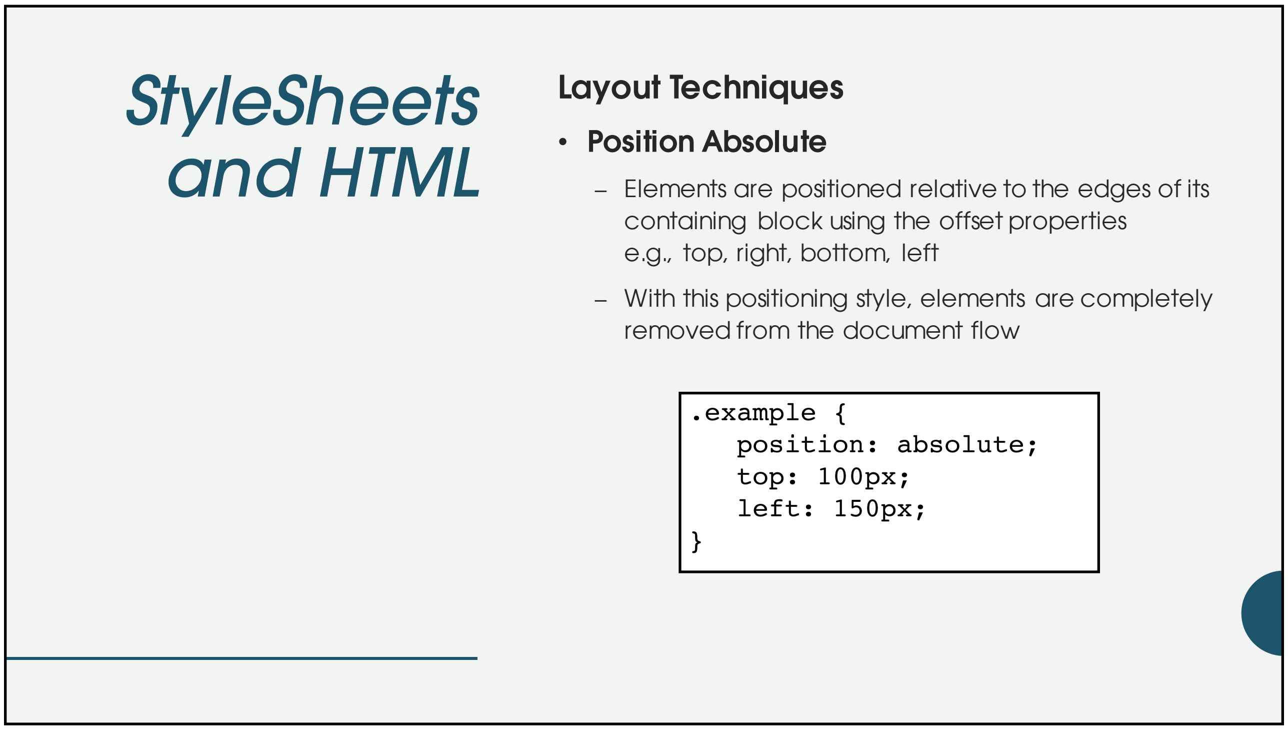
StyleSheets and HTML

Layout Techniques

- **Position Static**
  - This is the default positioning
  - Elements are rendered in the same order as they appear in the document flow

```
img.book {
   position: static;
   width: 40px;
}
```

  - If no position property is used for an element, the browser assumes its positioning style is static



StyleSheets and HTML

Layout Techniques

- **Position Absolute**
  - Elements are positioned relative to the edges of its containing block using the offset properties e.g., top, right, bottom, left
  - With this positioning style, elements are completely removed from the document flow

```
.example {
    position: absolute;
    top: 100px;
    left: 150px;
}
```

StyleSheets and HTML

Layout Techniques

- **Position Fixed**
  - Elements are positioned relative to the edges of the browser window using the offset properties e.g., top, right, bottom, left
  - With this positioning style, elements are completely removed from the document flow, and remained fixed on the browser window (i.e., always visible)

```
.example {
   position: fixed;
   top: 100px;
   left: 150px;
}
```

StyleSheets and HTML

Layout Techniques

- **Position Relative**
  - Elements are positioned relative to their initial position in the normal document flow using the offset properties
    e.g., top, right, bottom, left
  - With this positioning style, the original space of the relatively positioned element is preserved
  - Can be used to overlap elements
  - It is common practice to declare the position of a parent element as relative, and contain an absolutely position child element

```
.tweet:before{
   content:url(../images/twitter-icon.png);
    display: block;
    position: absolute;
    left: 15px;
    top: 4px;
}
```

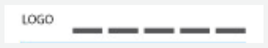
Adding a Layout to Our HTML Page

A CSS WARM-UP EXERCISE

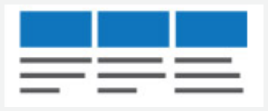
CSS: A Front-End Warm-up

Exercise 3 (15mins)

- Add some layout instructions to your
  - Use CSS layout techniques to define a similar layout to the one in the image below to your **<header> section**



  - Use CSS layout techniques to define a similar layout to the one in the image below to your **main content area**. Your HTML form should be displayed in the third column



Note: *ensure you use W3C compliant HTML*

http://www.webdesign.tn/meel/les-sites-web-non-adaptes-au-mobile-perdront-en-visibilite-dans-le-nouvel-algorithme-de-google/

Introduction

RESPONSIVE WEB DESIGN

Responsive Web Design

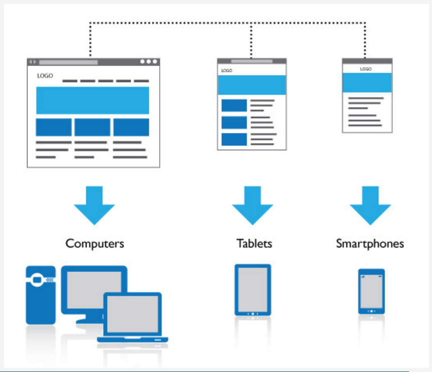
width
device-width

What is Responsive Design?

- It involves coding CSS in a way that ensures the content of a page (i.e., width) will adapt to the size of the browser window
- **Breaking points** are useful points of reference for when our layout will break
  - Mobile portrait: 320px
  - Mobile landscape: 480px
  - Small tablet: 600px
  - Tablet portrait: 768px
  - Tablet landscape/netbook/desktop: 1024px

Sharkie, C. and Fisher, A. Jump Start Responsive Web Design. Sitepoint: Australia, 2013.

Responsive Web Design



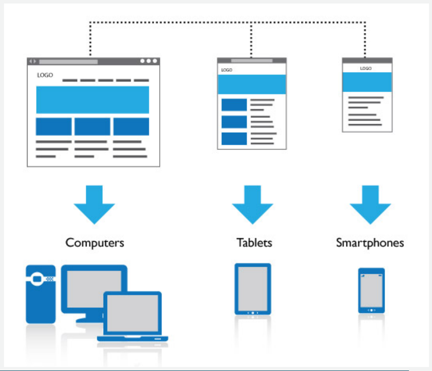
Responsive Web Design and Media Queries

- Responsive Web Design heavily relies on the use of media queries
- In this case, we use the **CSS media at-rule** (@media)

```
@media (max-width: 1500px) {
    /* CSS Code Here */
}
@media (max-width: 1200px) {
    /* CSS Code Here */
}
@media (max-width: 900px) {
    /* CSS Code Here */
}
```

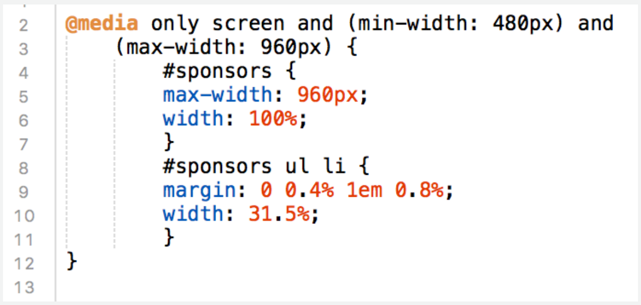
http://www.webdesign.tn/meel/les-sites-web-non-adaptes-au-mobile-perdront-en-visibilite-dans-le-nouvel-algorithme-de-google/

Responsive Web Design



Responsive Web Design and Media Queries

- Responsive Web Design heavily relies on the use of media queries
- In this case, we use the **CSS media at-rule** (@media)



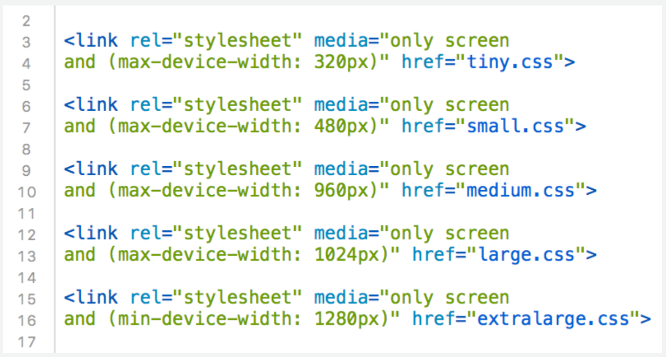
```
@media only screen and (min-width: 480px) and
    (max-width: 960px) {
        #sponsors {
        max-width: 960px;
        width: 100%;
        }
        #sponsors ul li {
        margin: 0 0.4% 1em 0.8%;
        width: 31.5%;
        }
}
```

http://www.webdesign.tn/meel/les-sites-web-non-adaptes-au-mobile-perdront-en-visibilite-dans-le-nouvel-algorithme-de-google/
Sharkie, C. and Fisher, A. Jump Start Responsive Web Design. Sitepoint: Australia, 2013.

Responsive Web Design

External StyleSheets & Media Queries

- Here are some examples of how media queries can be used to **filter** a **stylesheet**



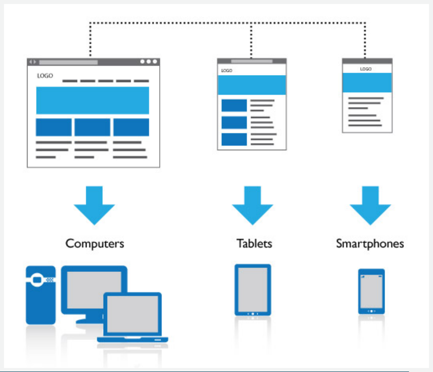
```
<link rel="stylesheet" media="only screen
and (max-device-width: 320px)" href="tiny.css">

<link rel="stylesheet" media="only screen
and (max-device-width: 480px)" href="small.css">

<link rel="stylesheet" media="only screen
and (max-device-width: 960px)" href="medium.css">

<link rel="stylesheet" media="only screen
and (max-device-width: 1024px)" href="large.css">

<link rel="stylesheet" media="only screen
and (min-device-width: 1280px)" href="extralarge.css">
```

http://www.webdesign.tn/meel/les-sites-web-non-adaptes-au-mobile-perdront-en-visibilite-dans-le-nouvel-algorithme-de-google/

Adding Some Responsiveness

RESPONSIVE WEB DESIGN EXERCISE

Responsive Web Design



Exercise 4 (30mins)

- Add some responsiveness to your Web Form page
  - Add a banner image to be displayed below the **<header> section**
  - Create a CSS layout similar to the image (left), targeting smartphones (i.e., 320px wide)
    In other words, your layout will go from 3 columns down to a 1 column layout
  - You must use relative measurements in your layout (i.e., em or percentages)
  - You may use dummy text and you have complete creative freedom

NOTE: You **MAY NOT** use an off-the-shelf mobile solution

http://www.webdesign.tn/meel/les-sites-web-non-adaptes-au-mobile-perdront-en-visibilite-dans-le-nouvel-algorithme-de-google/

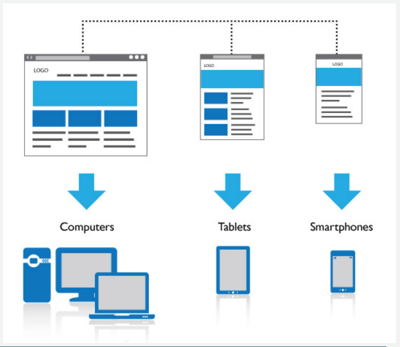
Done!

- Things for you to do:
  - Lab Assignment #1

Making our form responsive

LAB ASSIGNMENT # 1

Responsive Web Design

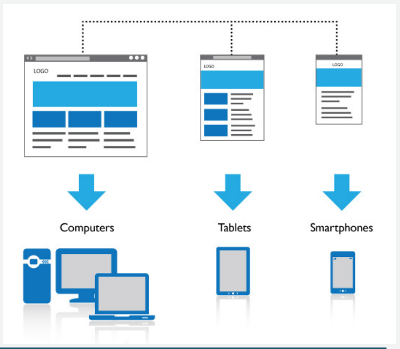


Using the Breakpoints for a Responsive Page

- Using the form page created in **Exercise 4**
  - Create a CSS layout with the following breakpoints: **320px**, **768px**, and **960px**
  - Your single-page site should go from a single-column to a three-column layout as the browser widths increase
  - You must use relative measurements in your layout (i.e., em or percentages)
  - You may use dummy text and you have complete creative freedom, and should strive to make an aesthetically pleasing result

NOTE: You **MAY NOT** use an off-the-shelf mobile solution

Responsive Web Design



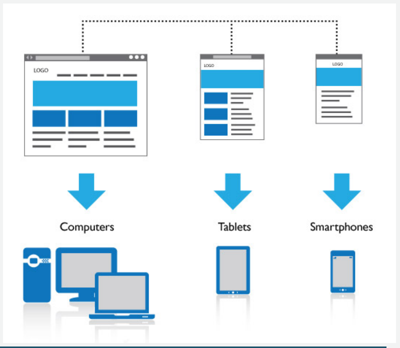
Using the Breakpoints for a Responsive Page

- Marking
  - Proper CSS layout with the following breakpoints: 320px **(1pt)**, 768px **(2pts)**, and 960px **(2pts)**
  - Your single-page site should go from a single-column to a three-column layout as the browser widths increase **(2pts)**
  - You must use relative measurements in your layout (i.e., em or percentages) **(2pts)**
  - Aesthetically pleasing **(1pt)**

TOTAL: 10pts

NOTE: You **MAY NOT** use an off-the-shelf mobile solution

Responsive Web Design



Using the Breakpoints for a Responsive Page

- Submitting your lab
  - Markers will go to the following URL for marking, please ensure you submitted your lab properly by visiting this URL yourself

http://web.cs.dal.ca/~yourcsid/csci3172/lab1/index.html

DUE: September 26th, 11:59pm.

NOTE: You **MAY NOT** use an off-the-shelf mobile solution