Web Site Workshop

Tracy Stepien

tls52@pitt.edu
http://www.pitt.edu/~tls52/

University of Pittsburgh Student Chapter of SIAM
March 24, 2012
Outline

1 Introduction

2 HTML
   • Header
   • Body

3 CSS
   • Colors
   • Text and Fonts
   • Links and Lists
   • Box Model

4 Layout
   • Grouping Elements
   • Navigation
Who am I?

- Built my first web site in 1999
- Graduated from University at Buffalo with a B.S. in Mathematics and a minor in Music Performance
  - Served as the web master for the Golden Key International Honor Society (2006–2008) and Impulse Dance Force (2006–2007) student organizations
  - Took a graduate level course in information architecture
- Currently a 4th year Mathematics Ph.D. student at Pitt
  - Served as the web master for the University of Pittsburgh Student Chapter of SIAM (2008–2011)
Web standards consist of recommendations and standards for aspects of the Internet published by various standards organizations. The main standards organization is the World Wide Web Consortium (http://www.w3.org/), which was founded by Tim Berners-Lee at MIT.

“Browser Wars” have been a large cause of web site designers’ headaches. Companies want users to only use their browser and no others. Standards were/are sometimes not followed and sites might look different in different browsers.

If a web site complies with web standards, it means that it has valid HTML, CSS, Javascript, etc.

Web Standards allows sites
- to be created and debugged faster
- to be both forward and backward compatible
- use less bandwidth
- require less time to maintain

More information: http://www.webstandards.org/
Adhering to Web Standards makes web sites **accessible**.

- greater visibility in web searches
- all current browsers able to display your web site the way it was meant to be viewed
- all browsers able to display the content of your web site
- smartphone browsers, voice browsers, and Braille browsers able to understand your web site

**Note:** If you post documents that cannot be displayed easily with HTML (e.g., C.V.) to your web site, it is highly suggested to convert to PDF.
Information architecture refers to the organization of the information and content on a web site.

Users want to find information quickly and without a lot of trouble ("findability" and "usability"), so do not create a maze (literally or figuratively) in which users need to guide themselves through.

A tool that can be used to help determine your organization is benchmarking.

- Look at other web sites that have a similar purpose as yours, and compare how its information is organized.

Usability Checklist:
Suggested Free Software

**Web site editor**
- Amaya ([http://www.w3.org/Amaya/](http://www.w3.org/Amaya/)) – Windows, Mac OS X, and Linux
- already installed on your machine: Notepad (Windows), TextEdit (Mac OS X), gedit (Linux with GNOME)

**Image editor**

**File transfer**
- Fetch ([http://software.pitt.edu/](http://software.pitt.edu/)) – Mac OS X
- gFTP ([http://www.gftp.org/](http://www.gftp.org/)) – Linux

**SSH client**
- X11 – installed in Applications/Utilities on Mac OS X
- Terminal – installed on Linux
Create a Pitt Web Directory

Pitt faculty, students, and staff can host personal web sites; that is what we will create today. Your web site can be viewed at http://www.pitt.edu/~username/

To get started, we need to create a Pitt Web Directory. (This only needs to be done once.)

Open a terminal and type the following:\1,\2:

```
>> ssh username@unixs.cssd.pitt.edu
>> cd public
>> mkdir html
```  

\1For PuTTY users: you will type unixs.cssd.pitt.edu in “Hostname” (make sure “SSH” is selected), then you will be prompted for your username and password. You will type the second two lines in the terminal.

\2The first time you SSH into a server from a different computer, you will need to add the host key to your cache (so when prompted, select “yes”).
An HTML file titled index.html must be located in your public/html folder so that your web site can be viewed.

However, since today we are going to develop our site, but we may not necessarily want to publish it yet, let’s create a “dev” folder.

In your terminal type the following:

```
>> cd html
>> mkdir dev
```

We will put all the files we work on today in this folder – you can view them at http://www.pitt.edu/∼username/dev/
Useful UNIX Commands

- `ls` (view the files in the current directory)
- `ls -l` (view the files in the current directory with permissions)
- `pwd` (find out what your current working directory is)
- `chmod 755 filename` (change the permissions of filename to everyone can view the file, but only the owner may edit it)
- `cd folder` (change the directory to a different folder – note that a folder above requires: `../`)
- `pico filename` or `emacs filename` (edit a text file with Pico or Emacs)
- `exit` (sign out)
HTML

**HTML** = Hypertext Markup Language

- markup language (not programming language) for describing web pages
- HTML documents contain HTML tags and plain text
  - HTML tags usually come in pairs – there is an opening tag and closing tag
The index.html Page

Open a web site editor.

Create a new HTML file – we’ll call it index.html

Let’s start from scratch with the required HTML tags and build up from there:

```html
<!DOCTYPE>
<html>
<head>
    <title></title>
</head>
<body>
</body>
</html>
```
<!DOCTYPE declaration

- must be the very first thing in your file
- is not an HTML tag; it is an instruction to the web browser about what version of HTML the page is written in
- refers to a Document Type Definition (DTD), which specifies the rules for the markup language so that the browsers render the content correctly
- see all choices at http://www.w3schools.com/tags/tag_doctype.asp

We will use XHTML 1.0 Transitional

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html> tag
- tells the browser that this is an HTML document
- also known as the root element
- container for all other HTML elements
- attributes
  - xmlns – specifies the namespace to use (required in XHTML, but invalid in HTML)
  - lang – specifies a language code for the content in an element
  - xml:lang – specifies a language code for the content in an element, in XHTML documents

Header: `<head>`

**<head> element**

- container for all the head elements
- must include a title for the document (what shows up in the title bar at the top of your browser), and can include scripts, styles, metadata, and more

```html
<head>
    <title>Insert Your Name Here</title>
</head>
```
<meta> tags

- provides metadata (information) about the HTML document for browsers, search engines, and other web services
- will not be displayed on the page, but will be machine parsable
- must be properly closed

attributes

- content: gives the value associated with the http-equiv or name attribute
- http-equiv: some values are content-language and content-type (see all options at http://www.w3schools.com/tags/att_meta_http_equiv.asp)
- name: some values are description, keywords, and robots (see all options at http://www.w3schools.com/tags/att_meta_name.asp)
<meta> tags

```html
<head>
  <meta http-equiv="content-type" content="text/html; charset="UTF-8" />
  <meta http-equiv="content-language" content="en-us" />
  <meta name="description" content="Personal web site of Insert Your Name Here" />
  <meta name="keywords" content="insert your name here, insert other keywords" />
  <meta name="robots" content="noimageindex" />
  <title>Insert Your Name Here</title>
</head>
```

More information about the robots value:

- [https://developers.google.com/webmasters/control-crawl-index/docs/robots_meta_tag](https://developers.google.com/webmasters/control-crawl-index/docs/robots_meta_tag)
Now that we have our document all set up, let’s start adding content!

- **Headings**
  
  \(<h1></h1>, <h2></h2>, <h3></h3>, <h4></h4>, <h5></h5>, <h6></h6>

- **Paragraph text**
  
  \(<p></p>\)

- **Stylizing paragraph text**
  
  emphasize (italicize): \(<em></em>\)
  
  strong/highlight (bold): \(<strong></strong>\)

- **Line break** (must be properly closed)
  
  \(<br />\)
Links
web site: <a href="http://www.pitt.edu/">Pitt</a>
e-mail: <a href="mailto:username@pitt.edu">e-mail</a>
within a page:
  <a name="elementname">anchored text</a>
  <a href="#elementname">link to anchored text</a>

Unordered List
<ul>
  <li></li>
</ul>

Ordered List
<ol>
  <li></li>
</ol>

3Worried about spam? Use an encoder: http://www.wbwip.com/wbw/emailencoder.html
**Table**

```
<table>
<tr>
  <th>Header 1</th>
  <th>Header 2</th>
</tr>
<tr>
  <td>Entry 1</td>
  <td>Entry 2</td>
</tr>
</table>
```

**Comments**

`<!--this text would not be displayed in the browser-->`
Images
the file must be uploaded into your Pitt Web Directory too
<img src="Image URL" height="size in px or %" width="size in px or %" alt="alternate text for the image" />

Pythagorean Cat

Will find hypotenuse
Let’s upload what we have so far!

Open a file transfer program.

Make sure the following items are entered (if they are displayed in your program):

- **Hostname**: unixs.cssd.pitt.edu
- **Port number**: 22
- **Username**: your Pitt username
- **Password**: your Pitt password
- **File protocol/Connect using**: SFTP

Navigate to the public/html/dev folder, and upload your index.html file and any images.

Go to [http://www.pitt.edu/~username/dev/](http://www.pitt.edu/~username/dev/) and look at your web site!
CSS = Cascading Style Sheet

- define how to display HTML elements
  - add colors, change fonts, choose the type of bullets for a list, change the layout, etc.
- can be inserted via an external style sheet, internal style sheet, or inline

We will create one external style sheet that will globally control how your entire web site looks.

Open your web site editor and create a new CSS file – we’ll call it style.css
selector {property1: value1; property2: value2;}

- "selector" is any HTML tag you would like to stylize (e.g., p, h1, body, a)
- "property" is what you would like to change (e.g., size, color)
- "value" is what you would like to change it to (e.g., 12px, red)
- comments: /*this text is commented*/
CSS Syntax

id

- specify a style for a **single, unique** element
- define with a #
- do not start its name with a number!
- example:
  - in HTML file: `<p id="centerstuff"></p>`
  - in CSS file: `#centerstuff {text-align: center;}`

class

- specify a style for a **group** of elements
- define with a .
- do not start its name with a number!
- example:
  - in HTML file: `<p class="makered"></p>`
  - in CSS file: `.makered {color:red;}` (all elements of class makered)
    - or `p.makered {color:red;}` (all p elements of class makered)
Colors in CSS can be specified by Hexadecimal\textsuperscript{4}, RGB\textsuperscript{4}, RGBA\textsuperscript{5}, HSL\textsuperscript{5}, or HSLA\textsuperscript{5} colors or Predefined/Cross-browser color names\textsuperscript{4}.

As Web Standards is important to our method of web design, we will use Hexadecimal colors:

- it is specified with #RRGGBB, where the RR (red), GG (green), and BB (blue) hexadecimal integers specify the components of the color – all integers must be between 0 (low) and f (high).
- if both R’s, G’s, and B’s, are the same integer, it can be written as #RGB instead

\textsuperscript{4}supported by all major browsers
\textsuperscript{5}supported by IE9+, Firefox 3+, Chrome, Safari, and Opera 10+
The 216 “Web Safe colors” ensure that all computers will display the colors correctly when running a 256 color palette; however, most computers can display millions of different colors now.

Some image editors can be used to determine a color’s hexadecimal integers, or a free online resource is [http://www.w3schools.com/cssref/css_colorsfull.asp](http://www.w3schools.com/cssref/css_colorsfull.asp)
So let’s change the color of the text, links, and backgrounds by editing style.css!

```css
body {background-color: #ff9;}
p {color: #ccc; background-color: #90c;}
h1 {color: #39c;}
h2 {color: #399;}
```
What are some other properties we can change for text?

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>direction</td>
<td>text direction/writing direction</td>
</tr>
<tr>
<td></td>
<td>ltr, rtl</td>
</tr>
<tr>
<td>letter-spacing</td>
<td>increases/decreases the space between characters</td>
</tr>
<tr>
<td></td>
<td>use units; negative values allowed</td>
</tr>
<tr>
<td>line-height</td>
<td>line height</td>
</tr>
<tr>
<td></td>
<td>use numbers (e.g., “2” is double spacing), units, or %</td>
</tr>
<tr>
<td>text-align</td>
<td>horizontal alignment</td>
</tr>
<tr>
<td></td>
<td>left, right, center, justified</td>
</tr>
<tr>
<td>text-decoration</td>
<td>decoration added</td>
</tr>
<tr>
<td></td>
<td>underline, overline, line-through</td>
</tr>
<tr>
<td>text-indent</td>
<td>indentation of the first line in a text-block</td>
</tr>
<tr>
<td></td>
<td>use units or %</td>
</tr>
<tr>
<td>text-transform</td>
<td>capitalization</td>
</tr>
<tr>
<td></td>
<td>capitalize, uppercase, lowercase</td>
</tr>
<tr>
<td>vertical-align</td>
<td>vertical alignment</td>
</tr>
<tr>
<td></td>
<td>use units or %, sub, super, top, text-top, middle, bottom, text-bottom</td>
</tr>
<tr>
<td>word-spacing</td>
<td>increases/decreases the space between words</td>
</tr>
<tr>
<td></td>
<td>use units; negative values allowed</td>
</tr>
</tbody>
</table>
Now let’s change the font family, boldness, size, and the style of text.

We can set all font properties using one property `font` or multiple properties `font-style`, `font-variant`, `font-weight`, `font-size`, `font-family`

If you use the `font` property, the order of the other properties is important:

```
body {font: font-style font-variant font-weight font-size font-family;}
h1 {font: font-style font-variant font-weight font-size font-family;}
p  {font: font-style font-variant font-weight font-size font-family;}
```
## Text Fonts

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>font-style</td>
<td>italicize</td>
</tr>
<tr>
<td></td>
<td>italic, oblique</td>
</tr>
<tr>
<td>font-variant</td>
<td>capitalization with original lowercase letters smaller</td>
</tr>
<tr>
<td></td>
<td>small-caps</td>
</tr>
<tr>
<td>font-weight</td>
<td>amount of boldness</td>
</tr>
<tr>
<td></td>
<td>use multiples of 100 (up to 900), bold, bolder, thicker, lighter</td>
</tr>
<tr>
<td>font-size</td>
<td>size of the font</td>
</tr>
<tr>
<td></td>
<td>use units or %, xx-small, x-small, small, medium, large, x-large, xx-large,</td>
</tr>
<tr>
<td></td>
<td>smaller, larger</td>
</tr>
</tbody>
</table>

### Font Size

To avoid a problem with resizing in some browsers, you can set a default font size in percent for `body` and then use the units em (1em=16px)

```css
body {font-size: 100%;}
h1 {font-size: 2.5em;}
h2 {font-size: 1.875em;}
p {font-size: 0.875em;}
```
font-family Property

For font-family, let’s review the difference between Sans-Serif and Serif fonts

On computer screens, sans-serif fonts are considered easier to read than serif fonts
The font you want to use may not be available on all browsers/operating systems, so a list of fonts, ending with a generic family, should be used as the value for font-family.

Sans-Serif Fonts
- Arial, Helvetica, sans-serif
- Arial Black, Gadget, sans-serif
- "Comic Sans MS", cursive, sans-serif
- Impact, Charcoal, sans-serif
- "Lucida Sans Unicode", "Lucida Grande", sans-serif
- Tahoma, Geneva, sans-serif
- "Trebuchet MS", Helvetica, sans-serif
- Verdana, Geneva, sans-serif

Serif Fonts
- Georgia, serif
- "Palatino Linotype", "Book Antiqua", Palatino, serif
- "Times New Roman", Times, serif

Monospace Fonts
- "Courier New", Courier, monospace
- "Lucida Console", Monaco, monospace
Stylizing Links

Four link states
- a:link – a normal, unvisited link
- a:visited – a link the user has visited
- a:hover – a link when the user mouses over it
- a:active – a link the moment it is clicked

Note these rules on the order
- a:hover must come after a:link and a:visited
- a:active must come after a:hover

Want to add an icon next to links to documents (e.g., PDF)?
If you have links to documents, it’s a good idea to let users know that. Either explicitly type the file extension or put an icon next to the link. Here’s example code for the latter. The padding and background selectors will be discussed later.

```css
a[href$='.pdf'] {padding: 5px 20px 5px 0; background: transparent url('Image URL') no-repeat center right;}
```
Stylizing Lists

You can set all of the list properties using one property `list-style` or multiple properties `list-style-type`, `list-style-position`, `list-style-image`. The order is important:

```css
ul {list-style: list-style-type list-style-position list-style-image;}
ol {list-style: list-style-type list-style-position list-style-image;}
```

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>list-style-type</td>
<td>list marker</td>
</tr>
<tr>
<td></td>
<td>ul: circle, disc, square, none</td>
</tr>
<tr>
<td></td>
<td>ol: decimal, lower-roman, upper-roman, lower-alpha, upper-alpha, lower-greek,</td>
</tr>
<tr>
<td></td>
<td>georgian, hebrew, none</td>
</tr>
<tr>
<td>list-style-position</td>
<td>markers appear inside or outside content flow</td>
</tr>
<tr>
<td></td>
<td>inside, outside</td>
</tr>
<tr>
<td>list-style-image</td>
<td>replace marker with an image</td>
</tr>
<tr>
<td></td>
<td>url('Image URL'), none</td>
</tr>
</tbody>
</table>

Note that this will not look the same in all browsers – see “Crossbrowser Solution” on [http://www.w3schools.com/css/css_list.asp](http://www.w3schools.com/css/css_list.asp) for an example of how to fix the problem.
Let’s upload what we have so far!

In order for our styling to show up, we must add the following line to within the `<head>` tag of every HTML file:

```html
<link rel="stylesheet" type="text/css" href="style.css" />
```

**Validate HTML and CSS**

When you view your web site, sometimes it displays differently than you thought it would. You can double-check that your code is correct through these validators:

- HTML: [http://validator.w3.org/](http://validator.w3.org/)
- CSS: [http://jigsaw.w3.org/css-validator/](http://jigsaw.w3.org/css-validator/)
- Check broken links: [http://validator.w3.org/checklink/](http://validator.w3.org/checklink/)
- Other validators: [http://www.w3.org/QA/Tools/](http://www.w3.org/QA/Tools/)
HTML elements can be thought of as boxes.\(^7\)

When setting the height and width of an element, you are setting the dimensions of the Content. The total size of an element includes the Margin, Border, and Padding.

\[
\text{total element width} = \text{width} + \text{left padding} + \text{right padding} + \text{left border} + \text{right border} + \text{left margin} + \text{right margin}
\]

\[
\text{total element height} = \text{height} + \text{top padding} + \text{bottom padding} + \text{top border} + \text{bottom border} + \text{top margin} + \text{bottom margin}
\]

\(^7\)We chose the XHTML 1.0 Transitional Doctype because it makes IE8 (and earlier versions) display boxes properly. (Web designers historically have strongly disliked IE for its inability to render the box model correctly.)
Problem: You only have space for an element that is 300px wide. Suppose you want an element with 10px padding all the way around and a 5px border all the way around. What do you set width equal to?

Solution:

width: 270px;
padding: 10px;
border: 5px solid gray;
margin: 0px;
You can set all of the border properties using one property `border` or multiple properties `border-width`, `border-style`, `border-color`. The order is important:

```
border: border-width border-style border-color;
```

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>border-width</code></td>
<td>the thickness of the border</td>
</tr>
<tr>
<td></td>
<td>use units, thin, medium, thick</td>
</tr>
<tr>
<td><code>border-style</code></td>
<td>border solid or not</td>
</tr>
<tr>
<td></td>
<td>none, dotted, dashed, solid, double, groove, ridge, inset, outset</td>
</tr>
<tr>
<td><code>border-color</code></td>
<td>color of the border</td>
</tr>
<tr>
<td></td>
<td>use color value, transparent</td>
</tr>
</tbody>
</table>
There are two ways that you can set the top, right, bottom, and left borders differently. We will use `border-width` as an example, but it also holds for `border-style` and `border-color`.

**Method 1**
Instead of `border-width`, write

- `border-top-width`
- `border-right-width`
- `border-bottom-width`
- `border-left-width`
**Border**

**Method 2**

`border-width: value1 value2 value 3 value 4;`

- value1 → top
- value2 → right
- value3 → bottom
- value4 → left

`border-width: value1 value2 value 3;`

- value1 → top
- value2 → right and left
- value3 → bottom

`border-width: value1 value2;`

- value1 → top and bottom
- value2 → right and left
Margin and Padding

- **Margin**: clears an area around an element (outside the border); does not have a background color (transparent)
- **Padding**: clears an area around the content (inside the border) of an element; affected by the background color of the element

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>margin</td>
<td>use units or %, auto (browser calculates a margin); negative values allowed</td>
</tr>
<tr>
<td>padding</td>
<td>use units or %</td>
</tr>
</tbody>
</table>

The margin and padding properties are defined similar to the border properties, where you can define individual sides (e.g., margin-top, margin-right, margin-bottom, margin-left) or use the short-hand as described on the previous slide ("Method 2" using, e.g., margin instead of border-width).
We used `background-color` in some of the previous examples, but what are all of the background properties?

```
background: color position size repeat attachment image;
```

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>background-color</td>
<td>solid color</td>
</tr>
<tr>
<td></td>
<td>use color value, transparent</td>
</tr>
<tr>
<td>background-position</td>
<td>starting position of image</td>
</tr>
<tr>
<td></td>
<td>use units, %, or left/right/top/bottom/center in the order width height</td>
</tr>
<tr>
<td>background-size</td>
<td>size of image if using background-image</td>
</tr>
<tr>
<td></td>
<td>use units of % in the order width height, cover, contain</td>
</tr>
<tr>
<td>background-repeat</td>
<td>repeat the image if using background-image</td>
</tr>
<tr>
<td></td>
<td>repeat, repeat-x, repeat-y, no-repeat</td>
</tr>
<tr>
<td>background-attachment</td>
<td>background is fixed or scrolls with the page</td>
</tr>
<tr>
<td></td>
<td>scroll, fixed</td>
</tr>
<tr>
<td>background-image</td>
<td>use a graphic as a background</td>
</tr>
<tr>
<td></td>
<td>url('Image URL'), none</td>
</tr>
</tbody>
</table>
Stylizing Tables

Now we have learned enough so that we can make our tables look much prettier! Try some of these:

- Define a class “rowhighlight” so that every other row is shaded (hint: `tr.rowhighlight{background-color: #ccc;}`)

- Change the border of the whole table (`table`) and/or just the header cells (`th`) and/or non-header cells (`td`)
  - To display a single border instead of double borders if you define borders for all three of the above, add `table {border-collapse: collapse;}` (note: need the XHTML 1.0 Transitional Doctype)

- Change how the text inside header and/or non-header cells aligns horizontally (`text-align`) or vertically (`vertical-align`)

- Change the space between the border and the content of the table by using padding on the header and/or non-header cells

- Change how far away the table is from the elements around it by using margin
Up to this point, you should know how to create HTML elements and stylize them with CSS. Now let’s talk about the actual layout of your web site!

It’s best to draw a diagram, called a wireframe, of where you want your content to go before you start trying to code it. We will work with this diagram:
It would be logical to group all the HTML elements together that give your contact information, or that provide navigation through your web site, or that contain your actual content. We need the following:

```html
<div>
</div>
```

**tag**

- defines a division or section in an HTML document
- used to group block elements to format them with styles
- browsers always place a line break before and after

We will use the `id` selector a lot along with CSS to group HTML elements.
Add <div> tags to HTML

Let’s look at our index.html page to add <div> tags to the sections specified in our wireframe.

```html
<div id="topnav">
</div>
<div id="pagewrapper">
  <div id="contact">
  </div>
  <div id="photo">
  </div>
  <div id="nav">
  </div>
  <div id="content">
  </div>
</div>
<div id="footer">
</div>
```
Stylize <div> tags with CSS

At the beginning of your style.css file, it’s a good idea to set the padding and margins to 0 for all possible HTML elements:

* {padding: 0; margin: 0;}

Let’s stylize each of the sections separately.

#topnav

- The div will extend to the width of the browser window, so we may want to just specify the height and background-color
Stylize `<div>` tags with CSS

#pagewrapper

- It is important to consider what screen resolution your users will have so they don’t have to scroll horizontally to view your content\(^8\). Setting the width of the majority of your site can be done through a wrapper div.
- Divs can be centered using margin.

```css
#pagewrapper {width: 800px; margin: 0 auto;}
```

#contact and #photo

- We want #photo to be a fixed width and height, and so we will also need to set the height of #contact the same and the width of #contact to be the width of #pagewrapper minus the width of #photo

```css
#contact {width: 300px; height: 300px;}
#photo {width: 500px; height: 300px; background-image: url('Image URL');}
```

\(^8\) Stats on users’ screen resolution: [http://www.w3schools.com/browsers/browsers_display.asp](http://www.w3schools.com/browsers/browsers_display.asp)
Stylize `<div>` tags with CSS

#contact and #photo

- Remember that browsers always place a line break before and after divs? To get two divs next to one another, we need to float both.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>float</td>
<td>how should a box float right, left</td>
</tr>
</tbody>
</table>

#contact {width: 300px; height: 300px; float: left;}
#photo {width: 500px; height: 300px; background-image: url('Image URL'); float: left;}

**Note:** In the HTML, right now you will only have `<div id="photo"></div>` (nothing else inside the tags)
Stylize `<div>` tags with CSS

```html
#nav
- Elements after floating elements will flow around, so we need to clear subsequent divs.
- We’ll stylize the navigation in a few slides.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>clear</td>
<td>which sides of an element other floats are not allowed right, left, both</td>
</tr>
</tbody>
</table>

#content
- It is easier to read dark text on a light background as opposed to light text on a dark background.

#footer
- If you want the footer to extend to the width of the browser window, it must be placed outside of the `#pagewrapper` div.
We can use CSS to make dynamic link rollovers for our navigation bar.

Note that display: block makes the whole link area clickable instead of just the text and is needed to specify the width and height.

**Horizontal Navigation Bar**

```css
#nav {clear: both; background-color: #3f9; height: 40px;}
#nav ul {list-style-type: none; margin: 0; padding: 0;}
#nav li {float: left;}
#nav a:link, a:visited {display: block; width: 142px; 
  height: 32px; font-weight: bold; font-size: 1.5em; 
  color: #006; background-color: #3f9; text-align: center; 
  vertical-align: middle; padding: 4px; 
  text-decoration: none; font-variant: small-caps;}
#nav a:hover, a:active {color: #c00; background-color: #f9f;}
```
Vertical Navigation Bar

You would need to change the height of #nav and re-do the wireframe, but the only thing different from the horizontal navigation bar is that you need to delete the line

```css
#nav li {float: left;}
```
**Indicating current page**

It’s nice to let your users know where they are in your web site, so make the link for the current page in your navigation slightly different.

**HTML**

```
<div id="nav">
  <ul>
    <li id="current"><a href="index.html">Home</a></li>
    <li><a href="research.html">Research</a></li>
  </ul>
</div>
```

**CSS**

```
#current a:link,a:visited {display: block; width: 142px;
  height: 32px; font-weight: bold; font-size: 1.5em; color: #c00;
  background-color: #f9f; text-align: center; vertical-align: middle;
  padding: 4px; text-decoration: none; font-variant: small-caps;}
#current a:hover,a:active {color: #c00; background-color: #f9f;}
```
Using images

We can use images for our link rollovers to make things look even fancier.

One benefit is then we can use special fonts that may not be on users’ computers.

Open an image editor and open the provided file navigation.psd

Once desired changes have been made, select File > Save for Web & Devices to save your image. If your image is transparent, save it as a gif. If your image is not transparent and needs to be higher quality, save it as a jpg.
#nav {clear: both; background-color: #3f9; height: 40px; width: 900px; float: left; position: relative;}
#nav ul {margin: 0; padding: 0;}
#nav li {margin: 0; padding: 0; position: absolute; background: url('nav.gif') 0 no-repeat; display: block; list-style: none;}
#nav a {height: 40px; display: block; text-indent: -9999px; text-decoration: none; outline: none;}

#nav #home {left: 0px; width: 60px; background-position: 0 0;}
#nav #research {left: 60px; width: 105px; background-position: -60px 0;}
#nav #teaching {left: 165px; width: 100px; background-position: -165px 0;}
#nav #contactinfo {left: 265px; width: 95px; background-position: -265px 0;}
#nav #personal {left: 360px; width: 100px; background-position: -365px 0;}
#nav #links {left: 460px; width: 440px; background-position: -460px 0;}

#nav #home a:hover, #home a.active {background: url('nav.gif') 0 -50px no-repeat;}
#nav #research a:hover, #research a.active {background: url('nav.gif') -60px -50px no-repeat;}
#nav #teaching a:hover, #teaching a.active {background: url('nav.gif') -165px -50px no-repeat;}
#nav #contactinfo a:hover, #contactinfo a.active {background: url('nav.gif') -265px -50px no-repeat;}
#nav #personal a:hover, #personal a.active {background: url('nav.gif') -365px -50px no-repeat;}
#nav #links a:hover, #links a.active {background: url('nav.gif') -460px -50px no-repeat;}

Navigation
Want to see even more examples?

Other random things you may want to know

Different stylesheets for computer, printing, mobile

You can create multiple style sheets so your site can be viewed differently on computer screens, smartphones, and in print-outs.

In the `<link>` tag, add

- `media="screen"` for computer screens (this is the default)
- `media="handheld"` for handheld devices (e.g., smartphones)
- `media="print"` for print-outs
- `media="all"` for all devices
Google analytics

Want to know how many visitors are coming to your site a day? Where are they located? What browser are they using? What operating system are they using? If they searched for you, what search words did they use?

The questions go on and on. Sign up for Google Analytics at http://www.google.com/analytics/.
Other random things you may want to know

Web Developer Toolbar for Firefox/Chrome

This is an amazing tool for web design! Some of the most useful features are the ability to outline HTML elements (static or via rollover), temporarily edit CSS/HTML within the browser, and validate code within the toolbar. Download it!

http://chrispederick.com/work/web-developer/
Want design ideas?

The CSS Zen Garden has beautiful designs that use the exact same HTML code, but different CSS. It is a great source of ideas for what you can do with the design of your web site.

http://www.csszengarden.com/

It is definitely okay to get design inspiration from other web sites, just please do not plagiarize!
References, Resources, and Links

- Tutorials in HTML, CSS, etc.: http://www.w3schools.com/
- E-mail address encoder: http://www.wbwip.com/wbw/emailencoder.html
- Fonts: http://www.dafont.com/
- Pitt web toolbox: http://www.umc.pitt.edu/toolbox/
- Your best friend: http://www.google.com/
- Google Analytics: http://www.google.com/analytics/

Validate code

- HTML: http://validator.w3.org/
- CSS: http://jigsaw.w3.org/css-validator/
- Check broken links: http://validator.w3.org/checklink/
- Other validators: http://www.w3.org/QA/Tools/
Usability

- World Wide Web Consortium: http://www.w3.org/
- Web Standards Project: http://www.webstandards.org/
- Usability resources:
- Usability Checklist:

CSS

- Using CSS for dynamic lists:
- Gallery of layouts using the same HTML code but different CSS:
  http://www.csszengarden.com/
- CSS colors hex values:
  http://www.w3schools.com/cssref/css_colorsfull.asp

Image editing

- GIMP tutorials: http://www.gimp.org/tutorials/