

CSCI 1720

Intermediate Web Design

Introduction / Review

East Tennessee State University
Department of Computing



CSCI 1720
Intermediate Web Design

1

Introduction

My name: Jack Ramsey
Office: Nicks 484
Phone: 423-439-5807
Email: ramseyjw@etsu.edu
Office hours: TBA

* You can arrange to meet me during office hours (it's best to email first, so I can be sure to be available), or other hours by arrangement

East Tennessee State University
Department of Computing



CSCI 1720
Intermediate Web Design

2

A Little About Me

Degrees:

AAS Police Science, 1992
 BS Criminal Justice/Criminology, 1995
 BS Computer Science, 1997
 MAT Education, 2010
 MS Computing (IT), 2013



A Little About Me

Work History:

1985-1988: U.S. Army, Hunter Army Airfield, Savannah, GA
 1998-1992: Various convenience stores while in school
 1992-1993: Charlottesville-Albemarle EOC, Charlottesville, VA
 1995-1997: Washington Co. Sheriff's Office, Jonesborough, TN
 1997-2009: Elizabethton Police Department, Elizabethton, TN
 2013-2014: Walters State Comm. College, Morristown, TN
 2014-present: East Tennessee State University, Johnson City, TN



A Little About Me

I've lived just outside the Greater Elizabethton Metropolis since early 1998

Part of the ETSU family since Spring 1994

Interests outside of computing:

- Reading (Sci-Fi, mystery, thrillers)
- Online gaming
- Photography
- Firearms / shooting

Interests in computing:

- Web design & development (HTML & CSS)
- System administration
- Amazon Web Services
- Computer networking
- Computer/network security
- Python, Java, Javascript, PHP, etc.



Syllabus / Tentative Course Schedule

Review



Review

In CSCI 1710, we learned about the Hypertext Markup Language (HTML 5), Cascading Stylesheets (CSS), web design and development principles, and a little about Hypertext Preprocessor (PHP) and forms

In production environments, HTML and CSS documents are rarely, if ever, hand-coded the way we did it in CSCI 1710

...but you have to learn to walk before you can run

This class will explore some more advanced tools and concepts that are used in production environments today



Review

This class (as did CSCI 1210) will focus primarily on front-end web development

As opposed to back-end development, which is the focus of CSCI 2910 (Server Side) and CSCI 3110 (Advanced Web)



Review

'When we discuss the "frontend" of the web, what we're really talking about is the part of the web that you can see and interact with...designers [who work] strictly in Photoshop and those who code HTML and CSS [and work] with JavaScript and jQuery'

~ <http://blog.teamtreehouse.com/i-dont-speak-your-language-frontend-vs-backend>



Review

'The backend usually consists of three parts: a server, an application, and a database. If you book a flight or buy concert tickets, you usually open a website and interact with the frontend. Once you've entered that information, the application stores it in a database that was created on a server.'

~ <http://blog.teamtreehouse.com/i-dont-speak-your-language-frontend-vs-backend>



Review - HTML

Current standard is HTML 5 (October, 2014)

Plain text files

Use 'markup' – tags – to define document structure

Browsers interpret the tags (elements) to render the structure of a web page when it is downloaded



Review - HTML

A tag is defined by **angle brackets**

<...>

An element is defined either by an opening tag, content, and a closing tag

`<p>This is where the content goes</p>`

While we referred to these as 'container elements' in CSCI 1710, [W3C](#) refers to them as 'normal elements'



Review - HTML

A tag is defined by **angle brackets**

<...>

...Or just a tag and its attributes

```

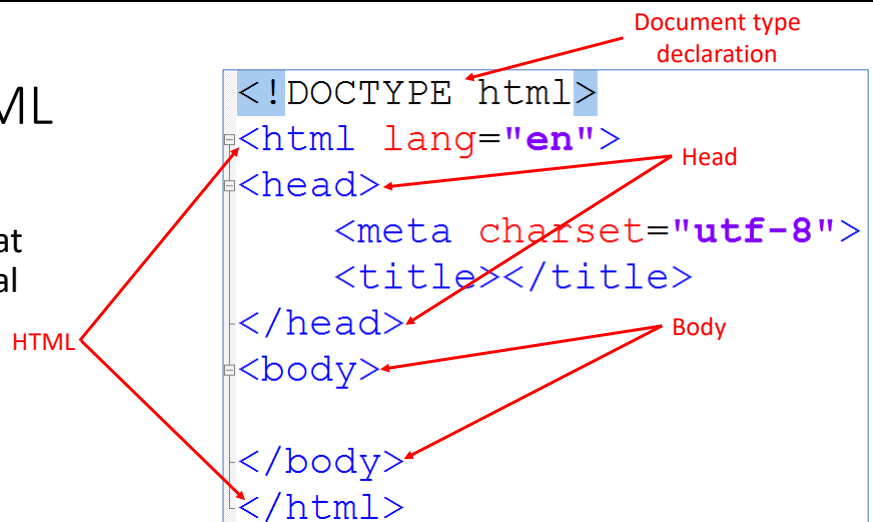
```

Again, we called these 'standalone elements' in CSCI 1710 while the [W3C](#) refers to them as 'void elements'



Review - HTML

All of our HTML documents will contain, at least, three major 'normal elements':



Review – HTML (Other Elements)

Some of the elements we explored include

<code>div</code>	<code>strong</code>	<code>header</code>
<code>p</code>	<code>em</code>	<code>footer</code>
<code>span</code>	<code>blockquote</code>	<code>img</code>
<code>h1 - h6</code>	<code>a</code>	<code>form</code>
<code>br</code>	<code>table</code>	<code>method</code>
<code>hr</code>	<code>tr, th, td</code>	<code>input</code>
<code>ol, ul</code>	<code>nav</code>	
<code>li</code>	<code>textarea</code>	
	<code>article</code>	
	<code>section</code>	



Review – HTML (Other Elements)

While not an exhaustive list of elements, those are the more common ones

Again, HTML defines a document's **structure**



Review – HTML Comments

Comments in HTML are denoted as follows:

```
<!-- This is a comment -->
```

With longer and more involved HTML documents, it's a good idea to include comments to help you (or whoever) maintain the documents



Review – CSS

Cascading Style Sheets (CSS) was a second major topic of CSCI 1210

CSS provides style (as opposed to structure)

Its primary responsibility is modification of how HTML elements are presented when a web page is loaded by a browser



Review – CSS

CSS can be applied to HTML documents in one or more of three ways:

External

Embedded

Inline

Of the three, **external style sheets** are the preferred method

External CSS provides a central location for styling rules, which enhances maintainability of a site



Review – CSS

A CSS rule consists of three parts:

selector

property(ies)

value(s)

The syntax for CSS rules differs from that of HTML

There are two syntactical scenarios



Review – CSS

External and Embedded:

```
selector {
  property: value;
  property: value1 value2;
}
```

Properties are separated from their respective values by a colon

Rules are separated from each other by a semi-colon

Rules are grouped within braces

Examples:

```
p {
  color: #00A;
  margin: 5px 15px;
}

table {
  width: 500px;
  border-collapse: collapse;
  border: 1px solid #000;
}
```



Review – CSS

Order of operations: In the event that two property/value groups conflict, the most recent will be applied

This is of critical importance, as we'll see when we start working with CSS frameworks like Bootstrap



Review – CSS

Order of operations: The order in which CSS rules that conflict are applied is:

- In-line
- Embedded
- External
- Browser default



Review – CSS

Order of operations: By 'conflicting rules,' consider the following:

(External)

```
p {
  width: 350px;
}
```

In this situation, the paragraph would be rendered by the browser 500px wide, since the inline CSS is the more 'recent' rule

(In the HTML document)

```
<p style="width:500px">...</p>
```



Review – CSS - Classes

Classes are created in CSS to allow for the selective application of a rule or group of rules to HTML elements

Again, this is very important when using CSS frameworks, because they provide many default classes ‘out of the box’

An example of the utility of classes is hyperlinks -- the

```
<a href="#">Link</a>
```

element



Review – CSS - Classes

These are hyperlinks

You want hyperlinks to appear differently depending on where they appear in the HTML document when it is rendered

Home Scratch Pics About

Home

Introduction

Lorem ipsum dolor [sit amet](#), gravida facilisi orci nulla. Erat dolor, sagittis blandit, maecenas quis mi suscipit, risus enim tincidunt erat. Eu justo, senectus ut mattis laoreet et. Nec ultricies ultrices iaculis non asperiores urna, nec ut dolor dignissim, sodales purus pretium eget, wisi integer molestie [arcu platea](#), urna ut in ipsum laoreet. Aenean fusce donec, arcu elit, placerat arcu laoreet libero eu vitae.



Review – CSS - Classes

We can apply multiple classes to a given element:

```
<p class="blueText narrowWidth serifFont">
```

by separating each class name with a space



Review – CSS - pseudo-Classes

A pseudo-class is used to define a special state of an element

For example, it can be used to

- Style an element when a user mouses over it
- Style visited and unvisited links differently
- Style an element when it gets focus



Review – CSS - pseudo-Classes

Pseudo-classes are defined using a colon (':'):

:link
:visited
:hover
:active

```
a:hover {
  color: #00A;
  background-color: #A00;
}
```

There are a lot of other pseudo-classes, but these are the ones we use the most



Review – CSS - Colors

There are a variety of ways to define colors in CSS

Probably the most often-used is hexadecimal

```
p {
  color: #FFFFFF;
}
```

```
p {
  color: rgb(255,255,255);
}
```

There is also rgb and rgba notation

```
p {
  color: rgba(255,255,255,.5);
}
```



Review – CSS - Colors

There are a variety of ways to define colors in CSS

We can also use color names

```
p {
    color: white;
}
```

(Also HSL (CSS 3) and HWB (CSS 4), but they haven't been fully accepted, yet)



Review – CSS - Text

We can modify than way text displays in a variety of ways

color	<code>color: red;</code>
alignment	<code>text-align: justify;</code>
decoration	<code>text-decoration: line-through;</code>
transformation	<code>text-transform: uppercase;</code>
indentation	<code>text-indent: 50px;</code>
letter spacing	<code>letter-spacing: 3px;</code>
line height	<code>line-height: 1.8;</code>
text direction	<code>direction: rtl;</code>
word spacing	<code>word-spacing: 10px;</code>
text shadow	<code>text-shadow: 3px 2px red;</code>



Review – CSS - Fonts

CSS enables modification of fonts

font family	<code>font-family: Arial, Helvetica, sans-serif</code>
font style	<code>font-style: italic;</code>
font size	<code>font-size: 30px;</code>
font weight	<code>font-weight: bold;</code>
font variant	<code>font-variant: small-caps;</code>



Review – CSS - Position

The CSS position property enables modification of where elements will be displayed when a page loads

Values for the position element are

static	default
relative	relative to the element's static position
absolute	positioned relative to the nearest positioned ancestor if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling
fixed	positioned relative to the viewport; stays in same place, even if the page is scrolled



Review – CSS - Position

The CSS position property enables modification of where elements will be displayed when a page loads

Property	Description
bottom	Sets the bottom margin edge for a positioned box
clip	Clips an absolutely positioned element
left	Sets the left margin edge for a positioned box
position	Specifies the type of positioning for an element
right	Sets the right margin edge for a positioned box
top	Sets the top margin edge for a positioned box
z-index	Sets the stack order of an element



Review – CSS - Float

The CSS float property, as we learned in CSCI 1210, has been used in the past to create, among other things, column displays

We also learned that it can be unpredictable and frustrating to use

We'll learn, this semester, about newer alternatives that give better, more predictable control over our web pages' layouts



Review – CSS - Combinators

A combinator is something that explains the relationship between the selectors

descendant selector (space)	all descendants
child selector (>)	immediate children
adjacent sibling selector (+)	adjacent sibling (immediately following parent)
general sibling selector (~)	all elements that are siblings of a specified element



Review – CSS - Combining Selectors

If there is a particular rule that we want to apply to more than one selector, we can shorthand it in the CSS stylesheets using a comma-separated list

```
h1, h2, h3, p, div {
    color: #00A;
}
```



Review Conclusion

This has been a quick look at topics covered more thoroughly in CSCI 1710
Hopefully, it will serve to both remind you of some of the finer points of HTML/CSS coding and prepare you for what we'll be covering this semester
Looking forward, we'll be exploring some of the following topics:



Review Conclusion

Responsive web sites
Bootstrap
Sass/SCSS
JavaScript
jQuery
PHP



Text Styles

CSCI 1720

East Tennessee State University
Department of Computing



CSCI 1720
Intermediate Web Design

41

Fonts / Typography

East Tennessee State University
Department of Computing



CSCI 1720
Intermediate Web Design

42

Typography

The art and technique of arranging type to make written language legible, readable, and appealing when displayed

- Typefaces
- Point sizes
- Line lengths
- Line-spacing (leading)
- Letter-spacing (tracking)
- Space between pairs of letters (kerning)

Also applied to the style, arrangement, and appearance of the letters, numbers, and symbols



Typography

Type – letters and other symbols that create words and phrases

Type family – several typefaces that share the same basic design form
Helvetica, Arial, Times New Roman (common type families)

Typeface (face) – all type of a single design and style

TR regular, TR italic, TR bold, TR bold-italic are all typefaces within the Times New Roman family

Most common type families usually have at least these 4 styles:
regular, bold (boldface), italic and bold-italic



CSS fonts

CSS recognizes 5 generic font families:

- Sans-serif
- Serif
- Monospace
- Cursive
- Fantasy



Serif/Sans-serif

Serif – short cross-stroke in font

A B C D E F G

Sans-serif – font with no cross strokes

A B C D E F G



Monospaced

Proportional – width is set per letter

This text is proportional using the Arial font

Monospaced – each letter is the same width

This text is monospaced using the courier font



Cursive

A font designed to resemble handwriting is called cursive

Brush Script
Edwardian Script
Freestyle Script
French Script
Kunstler Script
Lucida Handwriting
Mistral

No serifs, usually rounder, might be connected and lean to the right



Fantasy

Difficult to read, so use very sparingly

Chiller

Jokerman

Old English Text

Playbill

Rosewood

STENCIL



em

Equal to the currently specified point-size

Use *em* to make style sheets scalable

Historically, based on the width of letter 'm' in typesetting

Relative to the font-size of its direct **or** nearest parent





em

Default font size == 16px

1em == 16px

1.5em == 24px

2em == 32px

```
div {
  font-size: 1em;
}
p {
  font-size: 1em;
}
```

```
<div>
  <p>
    Lorem Ipsum
  </p>
</div>
```

How big would the text in the paragraph be?

32 pixels



rem

em is useful, but can get complicated

rem (Root EM) – size is calculated to be relative to the root (<html>) size

Can simplify size calculation

May require re-declaration of sizes for selectors





rem

Default font size == 16px

$1rem == 16px$

$1.5rem == 24px$

$2rem == 32px$

```
div {
  font-size: 1rem;
}
p {
  font-size: 1rem;
}
```

```
<div>
  <p>
    Lorem Ipsum
  </p>
</div>
```

How big would the text in the paragraph be?

16 pixels



Writing for the Web



usability.gov

People read differently online than they do when they read print materials -- **web users typically scan for information**

When writing for the web, using plain language allows users to find what they need, understand what they have found, and then use it to meet their needs

It should also be actionable, findable, and shareable

It's important to understand how what you are writing fits into the overall content strategy, what the content lifecycle entails, and who is involved in the process



usability.gov - Identify Users' Top Tasks

People come to your website with a specific task in mind

When developing your site's content, keep your users' tasks in mind and write to ensure you are helping them accomplish those tasks

If your website doesn't help them complete that task, they'll leave

Conduct market research, perform a task analysis and other types of user research, and analyze metrics to better understand what users are looking to accomplish

(We talked about requirements elicitation in CSCI 1710, remember?)



usability.gov - Identify Users' Top Tasks

Knowing your users' top tasks can help you identify:

Content to feature on your homepage or landing pages

Page headers and sub headers

A logical structure to each page's content



usability.gov

It's important to target your audience when writing for the web

By knowing who you are writing for, you can write at a level that will be meaningful for them

Use the personas you created while designing the site to help you visualize who you are writing for



usability.gov

Use the words your users use. By using keywords that your users use, you will help them understand the copy and will help optimize it for search engines

Chunk your content. Chunking makes your content more scannable by breaking it into manageable sections

Front-load the important information. Use the journalism model of the “inverted pyramid.” Start with the content that is most important to your audience, and then provide additional details

Use pronouns. The user is “you.” The organization or government agency is “we.” This creates cleaner sentence structure and more approachable content

Use active voice. “The board proposed the legislation” not “The legislation was proposed by the board”



usability.gov

Use short sentences and paragraphs. The ideal standard is no more than 20 words per sentence, five sentences per paragraph

Use dashes instead of semi-colons or, better yet, break the sentence into two

It is ok to start a sentence with “and,” “but,” or “or” if it makes things clear and brief

Use bullets and numbered lists. Don’t limit yourself to using this for long lists—one sentence and two bullets is easier to read than three sentences



usability.gov

Use clear headlines and subheads. Questions, especially those with pronouns, are particularly effective

Use images, diagrams, or multimedia to visually represent ideas in the content. Videos and images should reinforce the text on your page

Use white space. Using white space allows you to reduce noise by visually separate information



usability.gov

Use Microsoft Word's Readability Statistics feature—part of the Spelling & Grammar check—to measure your progress as you write and edit copy

Try to make your reading ease number go up and your grade level go down

You can improve your readability by using active voice and short words, sentences, and paragraphs



9 Tips

1. Treat your web visitors like wild animals

Your website visitors behave like wild animals

They're hunting for information or a product to buy – just like a hungry panther hunts for his next meal

When a panther sniffs a scent trail he quickly decides: will the scent trail lead to a good meal? And will it be an easy catch?

Your web visitors consider the same two things: Does your website offer what they're looking for? And can they find it easily?



9 Tips

2. Put your most important information first

Writing for the web is completely different from writing an essay or a paper

An essay might go like this: First, explain what you're going to discuss. Then, present an overview of the literature. Next, discuss; and finally draw your conclusion. The most important point you make is in the conclusion – at the end of your essay!

On web pages you have to do the opposite: your most important points always come first



9 Tips

3. Don't try to be clever or creative

On the web it's rare that a reader hangs on to every word you write. No time. He or she's in a hurry because he or she could check out several other scent trails – websites – instead of wasting time trying to figure out what you do

Simple statements often work best



9 Tips

4. Write for scanners

Research suggests that only 16% of people read web pages word-for-word

Most people scan

How can you write for scanners? A check list:

- Does your headline communicate what you're about?
- Does your image caption communicate a sales message?
- Do your sub headlines summarize your key points?
- Do easy-to-scan bullet points reduce wordiness?



9 Tips

5. Use Familiar Words

Carewords are the words people are looking for

We often like to make ourselves sound better than we are; embellish what we do; or try to sound scientific, fancy, or special

But your web visitor is looking for familiar words – carewords – because they're the scent trail that tells users that they're in the right place



9 Tips

6. Write for lazy people

Make your copy easy to read:

- Use short paragraphs – four sentences max
- Use short sentences – twelve words on average
- Skip unnecessary words
- Avoid jargon
- Avoid the passive tense
- Avoid needless repetition
- Address your web visitors directly
- Shorten your text



9 Tips

7. Expect people to arrive anywhere on your website

Most web visitors will not start reading at your home page

They may arrive on any of your web pages

If each web page can be an entry page what does that mean?

- Each page should be easy to scan

- Each page should clarify to people where they are; and what your site is about

- Each page should have a call to action telling people where to go next



9 Tips

8. Make it easy for hunters to find you

Potential customers are hunting for information or products

How can you help them find you?

Lure potential customers to your website by providing useful information. That's how writing for SEO (Search Engine Optimization) basically works:

- Answer the questions potential customers are asking

- Discuss one key topic for each page

- Include links to relevant pages on your own website or to other websites

- Use phrases and words your potential customers are looking for



9 Tips

9. Make a visual impression

Replace text by photographs or videos

Consider different font sizes – think about people scanning large text first

Emphasize quotes of customers (or experts) to add credibility

Play around with **highlights**, **bold text**, CAPS, or *italics*

Break a long headline into a headline with a sub headline

Change paragraphs into bullet points



Sources

<https://www.w3schools.com/css/default.asp>

<https://www.usability.gov/how-to-and-tools/methods/writing-for-the-web.html>

<https://www.enchantingmarketing.com/writing-for-the-web-vs-print/>

<https://www.nngroup.com/articles/how-users-read-on-the-web/>



Copyrights



Presentation prepared by and copyright of John Ramsey,
East Tennessee State University, Department of
Computing . (ramseyjw@etsu.edu)



- Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.
- IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWERS+, POWERS, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.
- Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.
- Oracle is a registered trademark of Oracle Corporation.
- HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.
- Java is a registered trademark of Sun Microsystems, Inc.
- JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.
- SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP Business ByDesign, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.
- Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects S.A. in the United States and in other countries. Business Objects is an SAP company.
- ERPsim is a registered copyright of ERPsim Labs, HEC Montreal.
- Other products mentioned in this presentation are trademarks of their respective owners.

East Tennessee State University
Department of Computing



CSCI 1720
Intermediate Web Design