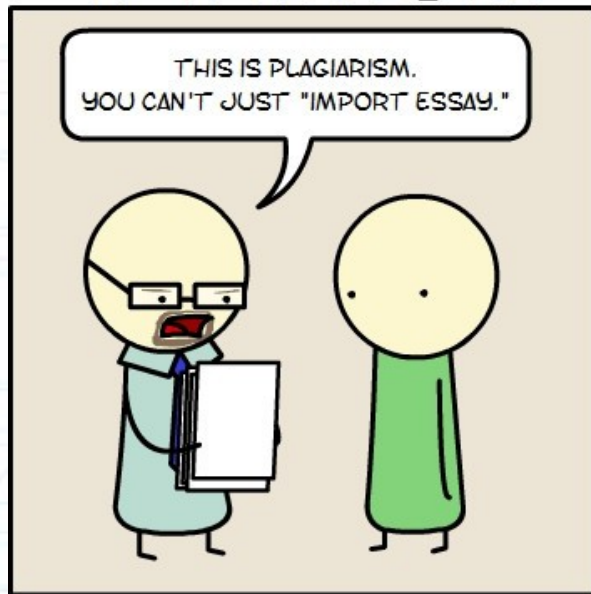


# HTML & XHTML

## PYTHON



## C++



## HTML



Comic from <http://www.somethingofthatilk.com/index.php?id=135>

CS/IT 490 WD, Fall 2013

Last update 2013-08-29

Written by Rachel J. Morris  
Creative Commons Attribution-NonCommercial-ShareAlike  
3.0 Unported License

# Breakdown

- HTML Tag Anatomy
- Deprecated Tags
- Awesome new HTML5 things
- Invisible Things
- Text
- Graphical Cues
- Data with Tables
- Layout
- Forms
- HTML 5



# HTML Tag Anatomy

- A tag has an opening and closing portion:  
`<h1></h1>`
- Or, some tags don't need inner content and can be closed in the same tag:  
``
- Tags **MUST** have opening & closing portions in order to comply with XHTML

# HTML Tag Anatomy

- ``
- Tags can have attributes  
`<img>` is the **tag**,  
`src` (source) is an **attribute**
- With XHTML, tag names and attribute names must be lower-case
- Attribute values must be enclosed within quotes (single or double)
  - Could get away without quotes before

# Deprecated Tags

- These tags are deprecated, and should NOT be used:

`<applet>`

`<center>`

`<dir>`

`<font>`

`<frame>`

`<strike>`

`<b>`

`<i>`

- Don't use `<marquee>` or `<blink>`, either.



# Awesome new HTML5 things

- New tags added to HTML5! \o/

<article>	<aside>
<audio>	<canvas>
<command>	<datalist>
<details>	<dialog>
<embed>	<figcaption>
<figure>	<footer>
<header>	<mark>
<nav>	<progress>
<source>	<summary>
<time>	<video>

And more. You can see a list of HTML elements here:  
<http://www.w3schools.com/tags/>

# Invisible Things

## <html>

This tag is required for your webpage to comply to XHTML standards.

Specifies that the document is an HTML page.

```
<html>  
<!-- Webpage Content -->  
</html>
```

# Invisible Things

## <!DOCTYPE>

This tag is required for your webpage to comply to XHTML standards.

This tag must be the first line in your page.

It specifies to what version of HTML you're complying with.

Different types...



# Invisible Things

## HTML 5

```
<!DOCTYPE html>
```

## HTML 4.01 Strict

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

## XHTML 4.01 Strict

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

There are other kinds, too:

[http://www.w3schools.com/tags/tag\\_doctype.asp](http://www.w3schools.com/tags/tag_doctype.asp)

Source: [http://www.w3schools.com/tags/tag\\_doctype.asp](http://www.w3schools.com/tags/tag_doctype.asp)

# Invisible Things

## <head>

This tag is required for your webpage to comply to XHTML standards.

Must contain the <title> tag

Can import JavaScript and CSS from within the head tag

Should contain metadata tags

```
<html>  
  <head>  
  </head>  
</html>
```

# Invisible Things

## <title>

This tag is required for your webpage to comply to XHTML standards.

The webpage's title shows up in the title bar of your web-browser, and the search result name

```
<html>  
  <head>  
    <title>Save Walter White</title>  
  </head>  
</html>
```



# Invisible Things

## <style>

Used to specify CSS styling, hard-coded into the page. *Generally, we'll want to **Link** them instead!*

```
<html>
  <head>
    <style type="text/css">
      body { color: #FF0000; }
    </style>
  </head>
</html>
```

# Invisible Things

## `<script>`

Used to include JavaScript files or hard-code scripts into the page.

```
<html>
  <head>
    <script type="text/javascript">
      alert( "Hello!" );
    </script>
  </head>
</html>
```

**Hard-coding JavaScript into a page**

# Invisible Things

## `<script>`

Used to include JavaScript files or hard-code scripts into the page.

```
<html>  
  <head>  
    <script src="scripts/thumbnails.js"></script>  
  </head>  
</html>
```

**Including an external .js file in your page**



# Invisible Things

## <link>

Used to link to external resources - “Favicon”, CSS stylesheets, etc.

```
<html>
  <head>
    <link rel="icon" type="image/png" href="/favicon.png" />
    <link rel="stylesheet" type="text/css" href="/styles/base.css" />
  </head>
</html>
```

# Invisible Things

## Metadata

Metatags let you define keywords, a description, etc. for your website.

A search engine *might* use your description, but it's more likely to use content found on the website itself.

# Invisible Things

Metadata goes inside the `<head></head>` tags

- **Keywords**

`<meta name="keywords" content="blog, video games, xbox, playstation, nintendo">`

- **Description**

`<meta name="description" content="User-created reviews on the latest games">`

- **Author**

`<meta name="author" content="Cliffy B">`

- **Refresh**

- **Encoding**



# Invisible Things

Metadata goes inside the <head></head> tags

- Refresh – Used to refresh the page periodically

```
<meta http-equiv="refresh" content="30">
```

- Encoding

HTML 4:

```
<meta http-equiv="content-type" content="text/html; charset=UTF-8">
```

HTML 5:

```
<meta charset="UTF-8">
```

# Invisible Things

## **SEO Tip: Character Encoding**

Specifying your page's encoding (such as UTF-8) is good practice – Users should generally have a UTF-8 (unicode) encoding installed on their computer.

User's computers might be running a different encoding by default (for example, our machines use Latin letters and originally didn't even contain accented letters!)

Specifying encoding is a **usability** point  
Usability features generally mean better SEO and better user experience in general!

Make sure to read this article to learn about unicode and localization, which is important in today's society!

<http://www.joelonsoftware.com/articles/Unicode.html>

# Invisible Things

`<!-- comments -->`

Comments can be used just like in code to clarify the use of a portion of markup, or to section off regions.

Tip: Don't use comments to comment-out code/markup:

***That's what source control is for!***



# Invisible Things

## <body>

This tag is required for your webpage to comply to XHTML standards.

The body tag begins the actual webpage; tags within **body** are rendered by the browser.

```
<html>  
  <body>  
  </body>  
</html>
```

# Text

## `<p>` Paragraph Tag

Text should be contained within `<p></p>` tags.

Automatically has margins (spacing) added by default, but you can override with CSS.

# Text

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Sample Page</title>
5  </head>
6
7  <body>
8    <p>
9      Lorem ipsum dolor sit amet,
10     consectetur adipiscing elit.
11   </p>
12 </body>
13 </html>
14
```

Note: even though this text is on a different line in the markup, without a **line-break** or **new paragraph**, no new-line will appear on the webpage itself





# Text

- **What is Lorem Ipsum?**
  - *Lorem Ipsum* is jumbled up Latin.
  - It is usually used to fill in a webpage with content without actually needing content. (Helps the page look filled out)
  - You can generate varying lengths of Lorem Ipsum copy at:  
<http://www.lipsum.com/>

# Text

## `<br/>` Line Break tag

A Line Break tag can be used to add a new-line to text. The Line Break does not contain the same default margins as a paragraph tag.

# Text

```
8  <p>
9  This is my first paragraph where
10 I have no line-breaks. Lorem ipsum dolor sit amet, consectetur adipiscing elit.
11 Aliquam tempus faucibus suscipit. Integer et est placerat,
12 </p>
13 <p>
14 This is a second paragraph, with
15 its own p tags. Lorem ipsum dolor sit amet, consectetur adipiscing elit.
16 Aliquam tempus faucibus suscipit. Integer et est placerat,
17 </p>
18 <p>
19 This is a third paragraph and <br/>
20 it has a line-break. <br/>
21 Lorem ipsum dolor sit amet, consectetur adipiscing elit. <br/>
22 Aliquam tempus faucibus suscipit. Integer et est placerat,
23 </p>
```

This is my first paragraph where I have no line-breaks. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam tempus faucibus suscipit. Integer et est placerat,

This is a second paragraph, with its own p tags. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam tempus faucibus suscipit. Integer et est placerat,

This is a third paragraph and  
it has a line-break.  
Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
Aliquam tempus faucibus suscipit. Integer et est placerat,

The text lines go until there's no more space then wrap around.

Separate paragraphs have more spacing than a single line-break



# Text

## Headers

Headers are used to specify a hierarchy of information on your webpage.

Headers include `<h1>`, `<h2>`, `<h3>`, `<h4>`, etc.

# Text

```
<h1>1990s style video game reviews!</h1>
```

```
<p>
  Tired of CSS and JavaScript mucking up your video gaming
  experience?! We all know that gamers only care about content
  and graphics don't matter at all!
</p>
```

```
<h2>Latest Games</h2>
```

```
<h3>This Month</h3>
```

```
<p>
  Saints Row IV<br/>
  Gone Home<br/>
</p>
```

```
<h3>This Year</h3>
```

```
<p>
  Animal Crossing New Leaf<br/>
  The Last of Us<br/>
</p>
```

```
<h2>Search by System...</h2>
```

## 1990s style video game reviews!

Tired of CSS and JavaScript mucking up your video gaming experience?! We all know that gamers only care about content and graphics don't matter at all!

### Latest Games

#### This Month

Saints Row IV  
Gone Home

#### This Year

Animal Crossing New Leaf  
The Last of Us

### Search by System...

Playstation 3

# Text

## **SEO Tip: Headers**

A page should only have **ONE** <h1> tag, but you can have as many <h2>, <h3>, etc. tags as you would like.

Think of the <h1> tag as the header stating what the page is all about.

Be sure to mark sub-headings with the appropriate header-tags as well! This helps the search engine figure out key terms of the webpage.



# Text

## Emphasizing text

While HTML tags for styling are deprecated (such as `<b>` and `<i>`), you can use tags like `<em>` and `<strong>` to specify what should have *emphasis* or be *stronger* on the webpage.

You can then style these with CSS, but by default `<em>` is italic and `<strong>` is bold.

# Text

```
<h1>1990s style video game reviews!</h1>
```

```
<p>
  Tired of CSS and JavaScript <strong>mucking up your video gaming
  experience?!</strong> We all know that gamers only care about content
  and graphics don't matter at all!
```

```
</p>
```

```
<p>
  <em>Note: This page is under construction!</em>
```

```
</p>
```

Tired of CSS and JavaScript **mucking up your video gaming experience?! We all know that gamers only care about content and graphics don't matter at all!**

*Note: This page is under construction!*

# Text

## Lists

When you have a list of data (steps in an algorithm, links in a navigation bar), it is good to store it with an **<ol>** (ordered-list) or **<ul>** (unordered-list) tag.

Nested within an **<ol></ol>** set is the **<li>** tag (list-item).

By default, they show up as a hierarchy, each list-item on its own line, but you can style these in CSS as well!



# Text

```
<h2>Recent Games</h2>
```

```
<ul>
```

```
  <li>Saints Row IV</li>
```

```
  <li>Gone Home</li>
```

```
</ul>
```

```
<h3>Top Games</h3>
```

```
<ol>
```

```
  <li>The Secret of Monkey Island</li>
```

```
  <li>Commander Keen</li>
```

```
  <li>Klonoa</li>
```

```
</ol>
```

## Recent Games

- Saints Row IV
- Gone Home

## Top Games

1. The Secret of Monkey Island
2. Commander Keen
3. Klonoa

# Text

## Nested Lists

Nested lists are a bit of a pain, but possible.

Within a List-Item tag (<li>), you need to open up a **new list...**

# Text

```
<h2>Top games by System</h2>
```

```
<ul>
```

```
  <li>Playstation 3
```

```
    <ol>
```

```
      <li>The Last of Us</li>
```

```
      <li>Uncharted 2</li>
```

```
      <li>LittleBigPlanet</li>
```

```
    </ol>
```

```
  </li> <!-- Playstation 3 -->
```

```
  <li>Xbox 360
```

```
    <ol>
```

```
      <li>Gears of War</li>
```

```
      <li>Halo 3</li>
```

```
      <li>Fable 2</li>
```

```
    </ol>
```

```
  </li> <!-- Xbox 360 -->
```

```
  <li>Wii
```

```
    <ol>
```

```
      <li>Super Mario Galaxy</li>
```

```
      <li>Super Smash Bros Brawl</li>
```

```
      <li>The Legend of Zelda</li>
```

```
    </ol>
```

```
  </li> <!-- Wii-U -->
```

```
</ul>
```

## Top games by System

- Playstation 3
  1. The Last of Us
  2. Uncharted 2
  3. LittleBigPlanet
- Xbox 360
  1. Gears of War
  2. Halo 3
  3. Fable 2
- Wii
  1. Super Mario Galaxy
  2. Super Smash Bros Brawl
  3. The Legend of Zelda



# Text

## Links

To link to content on another website, or another page on your own site, you use an **anchor tag**:

```
<a href="http://www.metacritic.com/">View Metacritic</a>
```

The Anchor tag `<a>` has attributes:

- href – The page being linked to
- title – Title of the link (will pop up a tool-tip on mouse-hover, also used for accessibility for blind people)

# Text

## Link attributes:

- name – The name of this link (you can also link to a section on the same webpage)
- target – Whether to open the link in a new window or current window

# Text

Links to another page:

```
<h2><a name="Links">Links</a></h2>

<p>
  <a href="http://www.metacritic.com/">View Metacritic</a>
</p>
<p>
  <a href="http://www.gametrailers.com/">View GameTrailers</a>
</p>
```

## Links

[View Metacritic](http://www.metacritic.com/)

[View GameTrailers](http://www.gametrailers.com/)



# Text

## Link to a section of the current page:

```
<h2><a name="Recent">Recent Games</a></h2>
<ul>
  <li>Saints Row IV</li>
  <li>Gone Home</li>
</ul>
```

These two headers have **anchor tags**, but instead of an href attribute we specify a name.

```
<h2><a name="TopGames">Top Games</a></h2>
<ol>
```

---

```
<h2>Table of Contents</h2>
<ul>
  <li><a href="#Recent">Recent Games</a></li>
  <li><a href="#TopGames">Top Games</a></li>
  <li><a href="#TopSystem">Top Games by System</a></li>
  <li><a href="#Links">Links</a></li>
</ul>
```

Rather than linking to a page, we can link to a #name attribute

---

## Table of Contents

- [Recent Games](#)
- [Top Games](#)
- [Top Games by System](#)
- [Links](#)

Can be used to create a table of contents, or allow a user to link to a specific section of a webpage easily.

# Graphical Cues

`<img/>`

You can display an image file on your page with the **img** tag.

The Image tag has attributes:

- **src** – The URL of the image (could be relative to the current page, or a full URL)
- **title** – The name of the image (will pop up a tool-tip on mouse-hover, also used for accessibility for blind people)
- **Do not use width and height attributes – they're deprecated!**

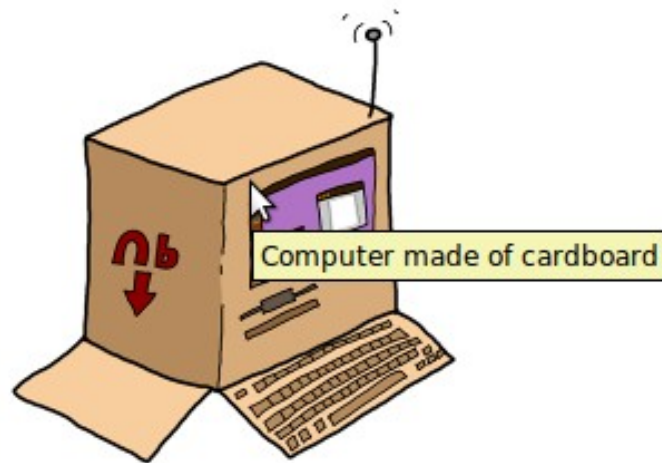


# Graphical Cues

```

```

## 1990s style video game reviews!



Tired of CSS and JavaScript **mucking up your video gaming experience?!**



# Graphical Cues

## SEO Tip: Titles

You should always include the **title** attribute for your anchor tags `<a>` and for any relevant image tags.

For image tags, describe what the image is a picture of. Don't worry about background images and nonimportant content, but try to add titles to anything pertinent.

There are accessibility tools for blind and hard-of-seeing people to use the internet – there are even blind programmers!

# Graphical Cues

`<hr/>`

You can use the **horizontal rule** tag to draw a line across the page.

It has a default styling, but you can update it with CSS.

There is nothing to store within an `<hr>` tag, so close it immediately.

# Graphical Cues

```
<p>  
  <em>Note: This page is under construction!</em>  
</p>  
  
<hr/>  
  
<h2>Table of Contents</h2>  
...
```

*Note: This page is under construction!*

---

## **Table of Contents**



# Data with Tables

- Tables are used to store data, similar to a spreadsheet.
- Tables **should not be used** for creating a webpage layout. That's what CSS is for.

# Data with Tables

asdf is what  
you type  
in the subject  
line  
when you have  
no subject.

asdf are the first  
four letters you  
learn in typing.

asdf likes  
jkl;

asdf is free.

asdf is  
what  
happens  
to fads.

asdf is nothing.

asdf just is.

asdf  
is  
green?

asdf is worth 8 points in  
scrabble.

asdf is the title  
of  
an unwritten  
zine.

aoeu is  
asdf's cousin

asdf is a  
four  
letter  
word.

asdf should  
be  
capitalized,  
but isn't.

asdf is a significant  
difference.

asdf could be this

asdf  
sounds like  
this?

Looks innocent enough, right?

Page from <http://asdf.com/whatisasdf.html>

# Data with Tables



Let's make the borders visible via FireBug...



# Data with Tables

asdf is what you type in the subject line when you have no subject.			asdf likes jkl;	
	asdf are the first four letters you learn in typing.		asdf is free.	asdf is what happens to fads.
			asdf is nothing.	
	asdf just is.	asdf is green?	asdf is worth 8 points in scrabble.	
asdf is the title of an unwritten zine.	aoeu is asdf's cousin		asdf is a four letter word.	asdf should be capitalized, but isn't.
	asdf is a significant difference.			
asdf could be this			asdf sounds like this?	

NO, THIS IS NOT WHAT TABLES ARE FOR.

Page from <http://asdf.com/whatisasdf.html>

Last update 2013-08-29

Written by Rachel J. Morris  
Creative Commons Attribution-NonCommercial-ShareAlike  
3.0 Unported License

# Data with Tables

## Table-related tags:

- `<table>` - Encloses all the table tags
- `<tr>` - Table Row
- `<td>` and `<th>` - Table cell
- `<thead>` - Wrap header `<tr>` and `<th>` within
- `<tbody>` - Wrap body `<tr>` and `<td>` within

# Data with Tables

```
<table>
  <thead>
  </thead>

  <tbody>
  </tbody>
</table>
```

Start with the outer `<table>` tag and defining the `<thead>` and `<tbody>`

```
<table>
  <thead>
    <tr>
      <th>Title</th>
      <th>Author</th>
      <th>ISBN</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>The Pragmatic Programmer</td>
      <td>Andrew Hunt, David Thomas</td>
      <td>978-0201616224</td>
    </tr> <!-- End Row -->
  </tbody>
</table>
```

Each row in a table is denoted by `<tr></tr>`

For the header, we use `<th>` for the cells

For the body, we use `<td>` for the cells



# Data with Tables

Tip: There are JavaScript libraries out there that will allow your table to be sorted by any column (or, columns you specify!)

Some libraries will also automatically style table rows so that alternating rows have different colors

# Data with Tables

```
<tbody>
  <tr>
    <td>The Pragmatic Programmer</td>
    <td>Andrew Hunt, David Thomas</td>
    <td>978-0201616224</td>
  </tr> <!-- End Row -->
  <tr>
    <td>Head First Design Patterns</td>
    <td>Elisabeth Freeman</td>
    <td>978-0596007126</td>
  </tr> <!-- End Row -->
  <tr>
    <td>Effective C++</td>
    <td>Scott Meyers</td>
    <td>978-0321334879</td>
  </tr> <!-- End Row -->
</tbody>
</table>
```

Each “Record” gets its own row,  
and each Column is specified by  
<td>...</td>

Use CSS to add borders and  
padding/margins

## Library Lookup

Title	Author	ISBN
The Pragmatic Programmer	Andrew Hunt, David Thomas	978-0201616224
Head First Design Patterns	Elisabeth Freeman	978-0596007126
Effective C++	Scott Meyers	978-0321334879

# Data with Tables

Tip: There are JavaScript libraries out there that will allow your table to be sorted by any column (or, columns you specify!)

Some libraries will also automatically style table rows so that alternating rows have different colors



# Forms

- Form Elements can be used to gather data and then push it to the server for it to be processed (via PHP or whatever you're using)
- Form elements have attributes like **name**, **id**, **type**, etc. to help differentiate between them when pushing data to the server.
- The data that comes out is essentially a key-value pair.
- We'll worry more about this later when we get to PHP

# Forms

## <form> tag

Any data that will be pushed to the server needs to be contained in a form tag.

You can have multiple form tags with different names. If a **submit button** is pressed within one form, data from the other forms won't be pushed.

You will want to include the **method** attribute in your form tag, set to either “post” or “get” – don't worry about this right now, we will discuss this in the PHP section.



# Forms

Form elements can have a **name** attribute and an **id** attribute.

- The name attribute is for the server: this is what will become our “keys” when the data is pushed as a key-value pair.
- The id attribute is for HTML: It identifies the tag itself and we can use it to link certain tags together.



# Forms

## <label> tag

A Label tag will link a text label to an input item.

Some input items don't automatically display their own labels (checkboxes, radio buttons, etc.)

Additionally, if you click a *label* that belongs to an input element, it will automatically toggle that element for you.

# Forms

`<label>` tag

Has an attribute called **for**. This should be the same value as the **id** of the input item we're linking it to.

# Forms

## <label> tag

```
<p>Features:  
<input type="checkbox" name="featureA" id="featureA" />  
<label for="featureA">Message Board</label>  
  
<input type="checkbox" name="featureB" id="featureB" />  
<label for="featureB">Email</label>  
  
<input type="checkbox" name="featureC" id="featureC" />  
<label for="featureC">Blog</label>  
</p>
```

Features: ☐ Message Board ☒ Email ☐ Blog



# Forms

`<button>` tag

Generates a button. You will have an opening and closing tag, and the text for the button in-between:

```
<button type="submit">Submit Form</button>
```

Button type can be **button**, **submit**, and **reset**.

You can also use the input tag to create buttons.

# Forms

## `<textarea>` tag

This is a large text input box, rather than a one-line text input box (that is the `input` tag).

For example, you would center comments or a review within a text area, while name and email would go in an `input` tag.



# Forms

## <select> tag

Select generates a drop-down box. To specify the contents of the dropdown box, you use the <option> tag.

```
<select>  
  <option value="KS">Kansas</option>  
  <option value="MO">Missouri</option>  
</select>
```



# Forms

## <input> tag

The input tag can be used as various types of elements, depending on what you specify as the **type** attribute.

### Buttons:

- type="button"
- type="submit"
- type="reset"
- type="image"

With type="image", specify a path to an image with the **src** attribute. This acts as an image button that submits a form.

# Forms

<input> tag

Text boxes:

- type="text"
- type="password"
- type="email"
- type="month"
- type="number"
- type="tel"
- type="time"
- type="week"
- type="date"
- type="search"
- type="range"



# Forms

`<input>` tag

More form elements:

- `type="checkbox"`
- `type="radio"`
- `type="file"`
- `type="hidden"`

Remember to use the checkbox and radio button with labels!



# Forms

```
<input type="checkbox" name="isInUSA" id="isInUSA" value="true" />
```

Checkboxes essentially act as boolean values – if it is checked, then the value shows up in our data array, and if it is not checked then it will NOT show up.

We'll learn more about this in PHP, but for example if this were submitted you would have something like

```
$_POST["myForm"]["isInUSA"] = "true"
```

# Forms

```
<input type="radio" name="shirtSize" id="size-s" value="small" />  
<input type="radio" name="shirtSize" id="size-m" value="medium" />  
<input type="radio" name="shirtSize" id="size-l" value="large" />
```

Radio buttons allow you to select one of  $n$  values from a list (which shows up as radio buttons).

Notice that the **name** for all these radio buttons are the same – This is the variable name in the data.

HTML elements must have different **ids**, however, so they have unique IDs

And, finally the **value** assigned is what will show up as the value of the “shirtSize” key->value.



# Forms

```
<input type="radio" name="shirtSize" id="size-s" value="small" />  
<input type="radio" name="shirtSize" id="size-m" value="medium" />  
<input type="radio" name="shirtSize" id="size-l" value="large" />
```

For a given one selected, our key-value pair will be something like

```
$_POST["myForm"]["shirtSize"] = "small"
```



# Forms

Notice that the **value** corresponds to the key-value array that will be sent to the server.

The Value will not automatically make a label display for radio/checkboxes – you need a label tag for that.

However, for an input button, the value will show up on the button's surface.

# Layout

- Layout tags don't necessarily have an appearance (unless you add, say, a border with CSS) but help you divide up the page into sections, which can be positioned and styled.



# Layout

<div> - Division, specify a section on your webpage. Example:

```
<div class="news-article">  
</div>
```

```
<div class="registration-form">  
</div>
```



# Layout

<div> - Division, specify a section on your webpage. Example:

```
<div class="news-article">  
</div>
```

```
<div class="registration-form">  
</div>
```

# Layout

`<span>` - you can wrap text within a span to give a custom style.

```
<p>
```

```
Registration Form
```

```
<span class="error">
```

```
There was an error processing the form:  
Password is only one character.
```

```
</span>
```

```
</p>
```

# Layout

`<code>` - Text within is treated as computer code; can be styled.

`<blockquote>` - Text within is treated as a comment.



# Layout

`<pre>` - Pre-formatted text goes in here. Anything within the pre tag will respect tabs and whitespace (usually extra spacing is ignored).

# HTML 5

Here are some tags new to HTML5

- `<article>`
- `<header>`
- `<source>`
- `<time>`
- `<embed>`
- `<canvas>`
- `<nav>`
- `<dialog>`
- `<figcaption>`
- `<footer>`
- `<summary>`
- `<video>`
- `<audio>`



# HTML 5

<article>

<header>

<footer>

<summary>

<source>

<nav>

<time>

<dialog>

<figcaption>

<time>

Surround corresponding content with these tags; you can style them or just use as a way to mark regions in your webpage. Google may also use for SEO.



# HTML 5

`<canvas>`

The HTML5 canvas is a fancy new feature of HTML5 – can be used to draw graphics with JavaScript.

It's somewhat meant to replace Flash with a less-proprietary technology.

e.x., there is an HTML5 version of the YouTube player

# HTML 5

<canvas>

Samples:

<http://www.kevs3d.co.uk/dev/phoria/>

<http://www.kevs3d.co.uk/dev/arena5/>

<http://wolf3d.atw.hu/>

List of demos:

<http://www.kevs3d.co.uk/dev/>



# HTML 5

`<embed>`

Allows you to link to an external plugin. Use the **src** attribute to specify the path.

Can include:

- Flash files (.swf)
- PDF files
- Anything that supports in-browser viewing?



# HTML 5

`<audio>`

Allows you to embed audio files of various formats (.mp3, .ogg, etc.)

Some attributes:

- `autoplay`
- `loop`
- `preload`
- `src`

Tip: NOBODY likes autoplaying music on webpages!

# HTML 5

`<video>`

Allows you to embed video files of various formats (.mp4, .ogg)

Some attributes:

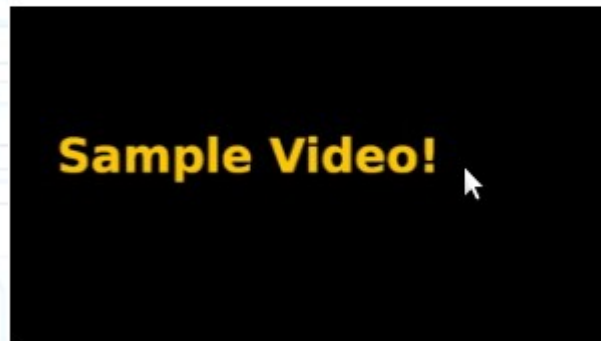
- autoplay
- width/height
- loop
- src

# HTML 5

The video tag by itself might not give you a really good method for playing video, you might want something closer to YouTube (i.e., with play button and navigation).

## **Video:**

OGV:





# HTML 5

There are various JavaScript libraries that give you a better video experience:

- VideoJS  
<http://www.videojs.com/>
- PopcornJS  
<http://popcornjs.org/>
- HTML5Video  
<http://html5video.org/>

# HTML 5

Same for audio, when embedding audio doesn't give you a nice interface:

- Jplayer  
<http://jplayer.org/>
- MediaElementJS (Audio & Video):  
<http://mediaelementjs.com/>

# HTML Tag Reference

- View a list of HTML tags here:  
<http://www.w3schools.com/tags/default.asp>



# References

-