

# HTML Tags (Tutorial, Part 2):

---

## Tags that need to go inside other tags:

For validation, certain tags must go inside of other tags. The ones we've seen so far that **must** go inside other tags include:

- `<a></a>` and `<img>` can and must go inside of just about any tag, most frequently
  - the `p` tags
  - the `h1` through `h6` tags
  - the `li` tags, or
  - the `td` tags.
- `<em></em>` and `<br>` must go inside other tags, usually `<p>` tags but also `h1 – h6`, `li`, and `td`
- `<li></li>` must go inside either `<ol></ol>` or `<ul></ul>`
- `<td></td>` must go inside of `<tr></tr>`
- And `<tr></tr>` must go inside of `<table></table>`

**DO NOT COPY code from the tutorials.** The quotes will cause problems. Your browser only recognizes this " as a quote. MS word uses this " as a default quote. That quote is not even a character for your browser, so it will cause errors. Yes, something that subtle will cause problems.

## Forms:

### How Forms Work:

1. You fill out form on Web page and hit submit
2. Browser sends content of form to Web server
3. Server receives form and data and passes it to the appropriate web application (software program someone wrote) to be processed
4. Web application creates a brand new HTML page
5. Server sends new HTML page back to Browser
6. Browser gets response and displays it.

We aren't worrying about writing the web application on the server side yet (step 3, above), which means we won't be processing the forms yet.

Instead we will be mailing the content of the form to someone or you can create a new mail address just to receive the form information.

### Part I. Form Elements:

*Either create a new html page by saving a copy of the template you created for hwk 1, or add to an existing html page (NOT the template) the following elements:*

*When checking your code, get used to loading your html page into both Chrome and Firefox.*

#### Input Button:

1. Add the following html code:  
`<input type = "button" value = "Click Here" onClick = "alert('You clicked a button!')">`
2. Load your web page in a browser (Firefox and/or Chrome) and click on the button. You should get an alert box popping up that says, "You clicked a button!".

Breaking it down:

Input : a group of form elements that allow the reader of the web page (the user) to interact with the element

Type = "button": this input element is and will appear as a button on your web page.

Value = "Click Here": this is the text that will appear on the button. It can be anything – it doesn't have to be "Click Here". Try changing it and reloading the web page. You'll see the text on the button changes.

onClick = "Alert('You clicked a button!')": onClick indicates that clicking on the button will cause an action. The action in this case is to bring up an Alert box that says, "You clicked a button!". Try changing the text "You clicked a button!" to "This is an Alert Box" and reload your web page and click on the button. The new text should show up in the Alert box.

**NOTE:** The Input element is another element that doesn't require a separate closing tag.

### text field:

3. Add the following html:

```
<p>Your Name: <br>  
<input type = "text" size = "20" name = "VisitorName"> <br>
```

4. Load your web page in a browser (Firefox or Chrome) and click on the button. You should get a one-line text box you can type in.

Breaking it down:

type = "text" means it's a text box

size="20" is the length of the box.

name = "VisitorName"> is the name given to the field when it's mailed to you.

5. In the text box code, add a default value:

To add a default value for the textbox, we need to add a value to the value. Modify your textbox so that it looks like look like:

```
<input type = "text" size = "20" value = "Henrietta Messerschmidt" name =  
"VisitorName" >
```

6. Save and view in browsers. See the difference?
7. Now try changing the 20 to 40.
8. Save a view in the browsers. Notice the difference.

### Radio Button:

9. Add the following code:

```
<input type="radio" name="hotornot" value="hot" > hot button<br >  
<input type="radio" name="hotornot" value="med" > medium button<br >  
<input type="radio" name="hotornot" value="cold" > cold button<br >
```

10. save and view in browsers. This is a Radio Button. Notice that the text that shows up is actual text (e.g., "hot button") OUTSIDE the input form tag, and not the value (e.g., "hot")

11. To add a default value, modify the cold radio button as such:

```
<input type="radio" name="hotornot" value="cold" checked>
```

12. Save and view in browsers. Notice how we set a default checked button.

### Checkbox:

13. add the following code:

```
<input type= "checkbox" name="spice" value="salt" > Salt <br >  
<input type= "checkbox" name="spice" value="pepper" > Pepper <br >  
<input type= "checkbox" name="spice" value="garlic" > Garlic<br >
```

14. save and view in browsers.

15. NOTE: To add a default value, modify the pepper and garlic buttons. as such:

```
<input type="checkbox" name="spice" value="pepper" checked> Pepper <br>  
<input type= "checkbox" name="spice" value="garlic" checked> Garlic <br>
```

16. Save and view in browsers.

### Select Box:

17. add the following code:

```
<select name="characters">
<option value="Eric"> Eric Idle </option>
<option value="TGilliam"> Terry Gilliam</option>
<option value="John"> John Cleese</option>
<option value = "Michael">Michael Palin </option>
<option value = "Graham">Graham Chapman </option>
<option value = "TJones">Terry Jones </option>
</select>
```

18. save and view in browsers.

### Text Area:

19. add the following code:

```
<textarea name="comments" rows="10" cols="48">Default text goes
here</textarea><br >
```

20. save and view in browsers.
21. change the size of the text area by changing the number of rows and columns. Try it.
22. save and view in browsers.

### Datalist (this is a new, html5 form element)

23. Add the following code:

```
<h4>Enter your favorite Cookie:</h4>
<input list="cookies">
  <datalist id="cookies">
    <option value="chocolate chip">
    <option value="fig newton">
    <option value="thin mint">
    <option value="oreo">
    <option value="samoa">
    <option value="ginger snaps">
  </datalist>
```

24. Now save and load into your browsers. Nothing special so far.
25. Start typing any letter at all.  
The datalist will give you a drop-down list of all the options in the datalist that contain that letter. If you start typing an actual cookie, it will narrow down the list to that particular cookie. Cool, huh.

## How forms work:

### Notes about forms:

- To create one form, surround a bunch of form elements (we've seen a few, above) with `<form>` and end with `</form>`. The `<form></form>` tags only occur **once** for each form. Everything between those tags belongs to that particular form.
- Inside the form tags, along with the form elements, you can include paragraphs, headers, etc.
- **In addition to other elements, all forms should have 1 (and only 1) "submit" button**  
`<input type = "submit"...>`  
**inside the form tags (usually right above the `</form>` tag)**
  - The submit button takes the content of the form and does whatever action you told it to do when you started the form

Forms are normally evaluated by a software program located on a web server. Those programs are often written in php. We are not going to learn php in this class. However, forms are often preprocessed and basic actions are taken using javascript.

We will be learning JavaScript later in this class. For now we will just mail our forms by adding the mailto action (see steps below):

### Mailing a Form:

1. Create a form by surrounding the form elements you created above with the following:

```
<form action = "mailto:youremailaddress@udel.edu" method = "post">
```

- replace [youremailaddress@udel.edu](mailto:youremailaddress@udel.edu) with a valid email address so you can see what happens.

2. inside the form, **Add a submit button:**

Add:

```
<input type = "submit" value = "Email This Form">
```

- **type = "submit"** means when you click on the button formed by this, action will be taken. (the action specified at the beginning of the form, or "mailto:youremailaddress@udel.edu" .
- **value="Email This Form"** is just the text put onto the button.

3. Close the form: Add:

```
</form>
```

4. save the file. View it in a browser (either firefox or chrome).
5. Fill in the form and click on the "Email This Form" button. Check your email.

That's it for forms!!

### Comments:

Comments are, in essence, notes that you write to yourself that you don't intend for the rest of the world to see. You can use a comment to remind yourself to add code or pictures later, to fix a problem, or to remind yourself what you were trying to accomplish when you designed the page. Comments can also help other people who will be working on the page with you or will be working on the page in the future.

An example of a comment might be something like:

```
...
<h3> Pictures of Lilly
</h3>
<p> These are pictures of my adorable new puppy Lilly. She's only 9 weeks old!
<!--Reminder: Add those pictures we took of Lilly on her first day in the pool -->
</p>
...
```

Comments start with

```
<!--
```

and they end with

```
-->
```

You can put them pretty much anywhere they'll be helpful to you or others.

You can also put them around HTML code that you want the browser to ignore for now – for instance, if you're trying to decide whether to include tags and content within a web page or not, you can comment out that set of tags and content and view the web page in a browser to see how the page looks without that material. If you decide you want to include it, you can just get rid of the comment tags rather than have to retype the whole section.

- 1: Add a comment to your forms web page.
2. Save and view in browser

#### Important Note:

**Comments can be used around code in your web page to make the browser ignore part of the web page. This is incredibly useful when you're trying to figure out exactly where a problem is in your code. For instance, sometimes when you are**

validating your web page, the error message w3c.org's validator gives you is unclear. You can put comments around sections of your page and revalidate until you get rid of the error. Then you will know exactly where the error is in your code, which makes fixing the error a lot easier!!

## HTML5 Elements

HTML 5 became public in 2008, but was not standardized (and still may not technically be standard). However, now the w3c (the world wide web consortium, which defines what tags are standard and how each browser should display them) will validate html 5 code.

### Html5 semantic elements

A significant component of html5 was to add tags that conveyed semantic meaning. With this your html code becomes more readable and understandable to others who read it and possibly take over coding for you someday.

There are a number of html5 tags used to convey meaning, but the ones you are most likely to use (and should use to make your code more readable) include:

`<header></header>`: this tag should go around the web page's head title (the h1, most likely) along with the web page's logo or main image, and possibly the web site's navigation. In other words, it goes around all the stuff that will most likely appear on every web page in a web site – all the header content).

`<footer></footer>`: This tag goes at the bottom of your web page, and goes around elements that usually appear on every web page in a web site. It usually includes the author's name, the date, copyright information, stuff like that.

`<section></section>` This tag goes around a bunch of elements that belong in one section. So, for instance, if your web page on mammals had a section on dogs, with a h2 header related to dogs, a few paragraphs on dogs, some links to dog-related information, and maybe an image or two of dogs, you'd put the section tag around all the dog-related content on your web page. There can be more than one section on a web page.

`<nav></nav>` This tag goes around site navigation links. So in a web site, each page usually has links to other web pages within the web site. The `<nav>` tags would go around these links.

`<aside></aside>` This tag is for content that is related to the other content within a section, but not directly in the flow – it's an aside.

1. In the web pages you've created so far, add a header tag, a footer tag, a section tag, and a nav tag. If you like, add content so that these tags make sense.
2. View the page in the browsers. It shouldn't look any different – these tags are for making your code more readable and for styling purposes (which we will be getting to in the near future).

### HTML5 Media Elements:

HTML5 also allows for media elements, like audio and video.

#### *Audio:*

To add an audio element, first make sure you've got an mp3 sound file downloaded to your computer. Place that file in the same folder as the web page you are editing.

NOTE: There are 3 types of sound files: mp3, ogg, and wav. Firefox and Chrome will play all three types. Safari does not handle ogg files, and IE does not handle ogg or wav files. For our class, you can use all 3 sound file types, but if you want to be sure that the sound file will play on all browsers, you should stick with mp3 files.

1. Add the following code to your web page:  

```
<audio controls>  
  <source src = "song.mp3" type = "audio/mpeg">  
</audio>
```
2. Replace song.mp3 with the name of the mp3 file you downloaded onto your computer.
3. Save and open the web page in Chrome and Firefox. Test the audio to make sure you can hear the audio.

## Video:

To add a video element, first make sure you've got an mp4 video file downloaded to your computer. Place that file in the same folder as the web page you are editing.

NOTE: There are 3 types of video files: mp4, ogg, and WebM. Firefox and Chrome will play all three types. Safari and IE do not handle ogg or WebM files. For our class, you can use all 3 sound file types, but if you want to be sure that the video file will play on all browsers, you should stick with mp4 files.

1. Add the following code to your web page:

```
<video width = "320" height = "240" controls>  
    <source src = "movie.mp4" type = "video/mp4">  
</video>
```
2. Replace movie.mp4 with the name of the mp4 file you downloaded onto your computer.
3. Save and open the web page in Chrome and Firefox. Test the video to make sure you can play it.

## Viewport:

You can control the visible area of your web page depending on the device in which it is being viewed. In other words, if the web page is being viewed on a phone, you want the visible area to be smaller than if it is on a computer screen, and again, you want it to be a different size on a tablet.

You can do this using viewport.

To set the viewport to be the width of the device, you'd add the following code in the head section (NOT the header section you just added) of your web page as follows:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Width=device-width sets the width of the web page to be the same as the width of the device it is being viewed on, and initial-scale=1.0 means the entire web page's width will be set to the width of the device.

Note that this doesn't work if you've set element sizes to be a fixed width in pixels (as opposed to % sizes).

Note2: you can set the width to be an exact size (width=500) if you are designing your web page to fit precisely on a particular device.

This is an imperfect but relatively quick way of making sure that your web page is viewable on more than one device today.

## One More thing...

### Special Characters:

As you know, the browser ignores spaces, tabs, and carriage returns (when you create a new line). We know the tag to create a new line is <br >.

What if you want to add spaces, and you don't want the browser to ignore them? For instance, if you've got

```
<p> This  is          delayed gratification! </p>
```

The browser will show,

```
This is delayed gratification!
```

which probably isn't what you want. You can force the browser to add spaces using:

```
<p> This  is &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; delayed gratification! </p>
```

Each &nbsp; is interpreted as a space by the browser.

There is a list of special characters and how you can include them at:

<http://www.utexas.edu/learn/html/spchar.html>

### That's it.

There are more html tags (e.g., <code>,<pre>, etc.), but we've covered all the basic, most frequently used tags..

## For more information on html tags:

<http://www.w3schools.com/tags/default.asp>