Introduction to Web Programming

Open the `StartingCode` folder and try to duplicate each page below.

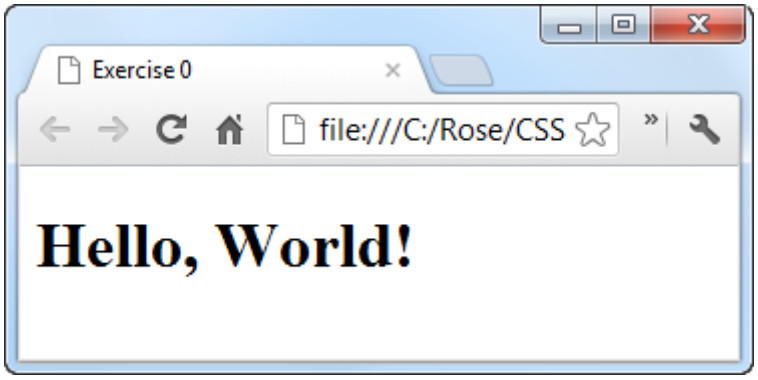
If some of the early exercises are too simple for you, feel free to skip them.

Exercise 0

Don't worry about the `<meta name="..>` tag. It is used to help get the page size for mobile browser use.

Add a Heading 1 tag with the innerHTML "Hello, World!".

http://www.w3schools.com/tags/tag_hn.asp

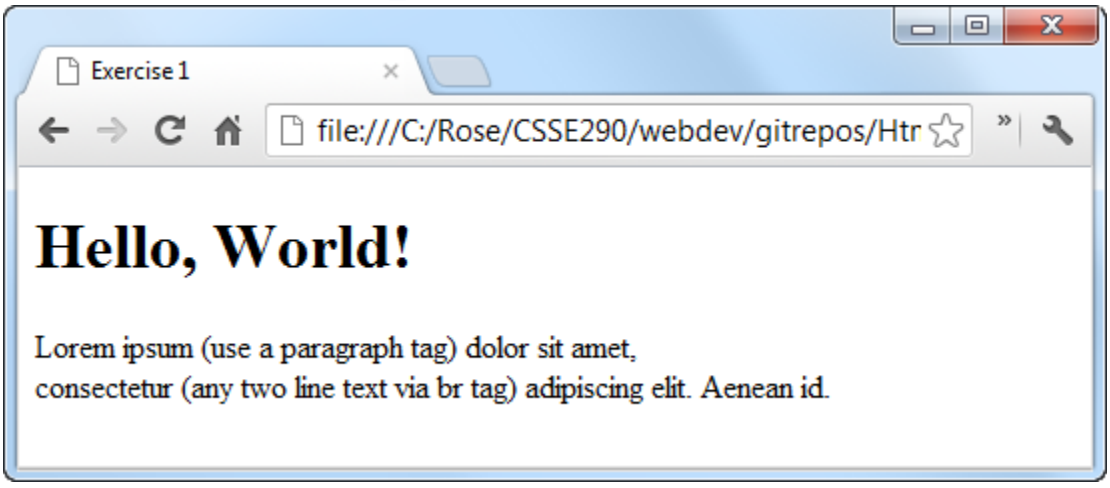


Exercise 1

Copy your exercise 0 additions into the `exercise1.html` file and add a **paragraph** tag with with at least two lines. Use a **br** tag to split the lines (the text can be anyhting you want)

http://www.w3schools.com/tags/tag_p.asp

http://www.w3schools.com/tags/tag_br.asp

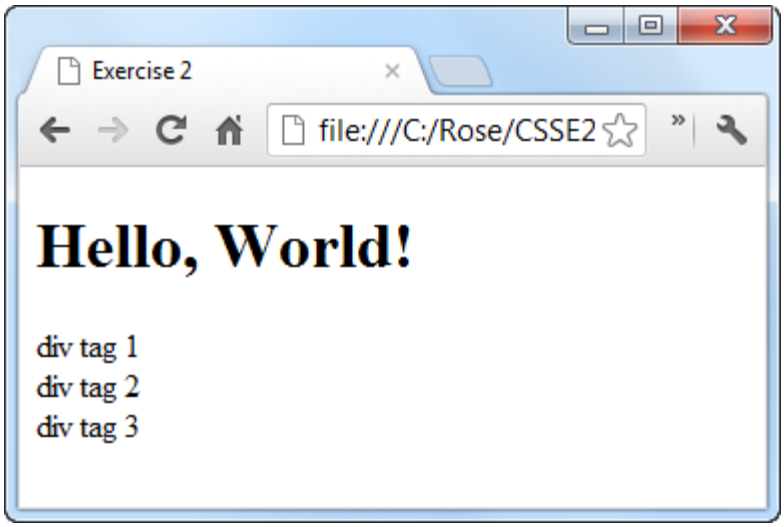


Exercise 2

Copy your exercise 0 addition into the exercise2.html file, but this time make 3 elements that use div tags (instead of a paragraph). Notice how a div's **display** defaults to a **block** to take the entire line.

http://www.w3schools.com/tags/tag_div.asp

The **div** tag is very popular. Interestingly using CSS you *could* make a div tag look like about anything (buttons, images, input elements, etc)



Exercise 3

Copy your exercise 0 addition into the exercise3.html file, but instead of adding 3 **div**s, add three elements with **span** tags. Notice how the **display** defaults to **inline**.

http://www.w3schools.com/tags/tag_span.asp

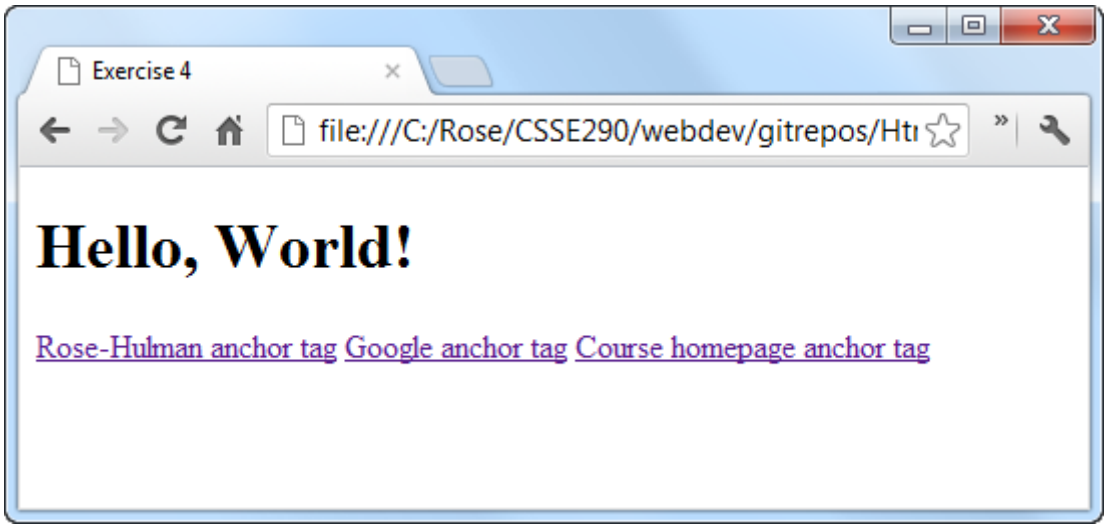
Even though a **div display** defaults to **block** and a **span display** defaults to **inline**, by using CSS you *could* completely change that default behavior.

Exercise 4

Copy your exercise 0 addition into the exercise4.html file, but, instead of adding 3 divs or 3 spans, add 3 elements with **anchor** tags <a>.
http://www.w3schools.com/tags/tag_a.asp
Set the href attribute on the anchors tags so that they point to:

- https://developer.mozilla.org/en-US/
- http://www.google.com
- https://developer.mozilla.org/en-US/docs/Web/HTML/Element/a

Notice they are **inline** (similar to a span).



Make sure that when you click the links, the browser takes you to the new locations.

From here on out we'll stop adding the Hello, World! `h1` tag. :)

Exercise 5

This time we'll load three images. Add a Rose logo, a google logo, then set the third image to anything you like.
http://www.w3schools.com/tags/tag_img.asp
I used these values for the src attribute:

- http://www.rose-hulman.edu/media/78/logo_rhit.png
- https://www.google.com/images/srpr/logo3w.png
- https://lh5.googleusercontent.com/-vB9HMliwbIY/AAAAAAAAAAI/AAAAAAAAGQY/gjPsSDgHxHc/s250-c-k/photo.jpg

It's fun that it doesn't matter where an image is at on the web. Use images on your site or some other site works exactly the same. **Just make sure they don't move the images on you (which is why it's safer to use images you control).**

In addition to the `src` attribute, set the **width attribute to "200px"** and set the alt attribute for all img tags. Best practice says you should ALWAYS set the alt attribute on an image (no exceptions, even if you are 100% sure the image will be there and load). **Add an `hr` tag between images to keep them separate.**

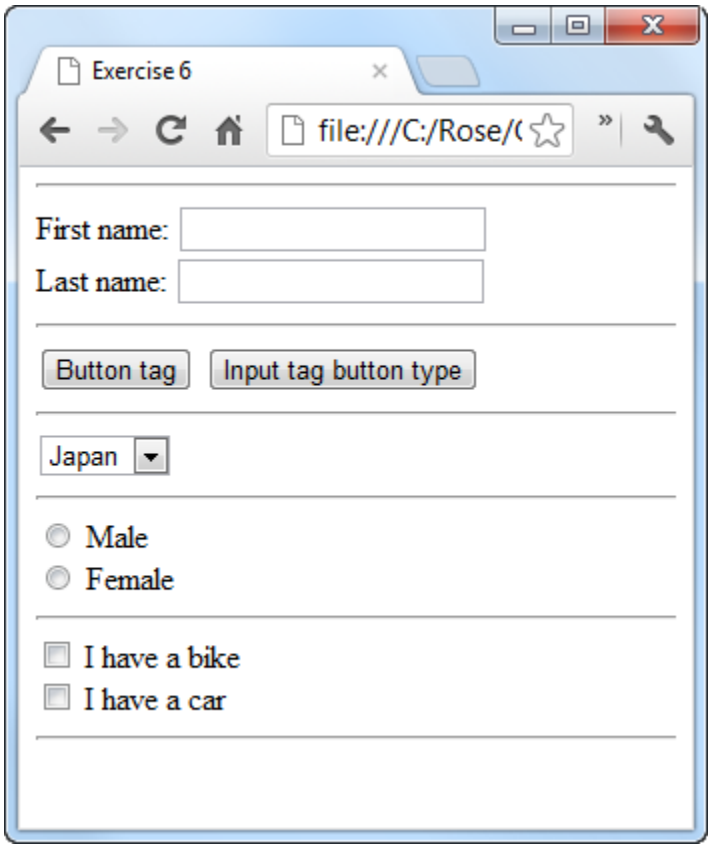
Note, we can also add images *via* the **background-image** CSS property. We'll use that CSS property on non-img elements later in the course as it has some advantages compared to the **img** tag.

Exercise 6 (another day)

Warning: this exercise is a bigger one than exercises 0-5.

There are many ways for a user to provide input to a page. For this exercise we'll make will make...

- Two **input** elements of type = **text**
- One **button** element
- One **input** element of type = **button**
- One **select** element
- Two **input** elements of type = **radio**
- Two **input** elements of type = **checkbox**



(select element options not shown... Japan, US, China, Ireland)

Take a look at this reference page and look at the **input** tag's **type** attribute. http://www.w3schools.com/tags/tag_input.asp Input elements with different **types** are VERY different, so that attribute is much more important than a typical attribute. To be honest, the input's **type** attribute feels more like a tag than an attribute since different types are so different.

Another element on the page is a **button** tag. I find it interesting that a button tag is nearly identical to a input tag with the type set to button. However, with a button you have **innerHTML for the text** and with an input type=button you have a **value for the text**.

In addition to the 7 inputs and button on the page we'll also make a **select** element with the **option**s Japan, US, China, and Ireland. Notice that the **innerHTML displayed to the user** doesn't have to match the **value that will be sent to your server** if that option is selected. My values for the options are JP, US, CN, and IE.
http://www.w3schools.com/tags/tag_select.asp
http://www.w3schools.com/tags/tag_option.asp

(depending on your OS + browser some elements might look different than what is shown here)

See if you can knock it out solo (notice the `hr` elements dividing the input sections and `br` between the First and Last name inputs). If you can't figure it out, look in the solution folder, but try hard to work it out before cheating. :)

Exercise 7 (another day)

In this exercise we'll structure the display of our data with **tables** and **lists**. Tables are old school tags that have many types of attributes (most of them are now deprecated). The table tag is the #1 most overused layout tool in web development history. It's very easy to make up front, but often doesn't work out well if you ever want to make changes to the layout. Numerous web developers have given you the warning to **only use a table tag for an actual table** (even though I break that rule sometimes myself). You should almost never use the `table` tag for your main page layout. However, tables are a good mechanism for presenting certain data-sets and lists.

Elements using the **table** tag can have a body tag within them called **tbody.** There are also **thead** and **tfoot** tags. In this example we'll use a **thead** and a **tbody**, but no **tfoot**. The **thead** and **tbody** sections have **tr** (tr = table row) elements in them for each row in the table (usually a **thead** has only 1 row). Each table row has either a th (...) or a td (...) for each cell in the row. For basic tables, each row really should have the exact same number of cells (or else it looks like a missing tooth in your table). There are ways to span multiple columns (colspan), but we won't do that here (so your number of cells should be the same in each row). Here are the reference pages that talk about tags we will be adding and some information about their attributes...

- http://www.w3schools.com/tags/tag_table.asp
- http://www.w3schools.com/tags/tag_thead.asp
- http://www.w3schools.com/tags/tag_tbody.asp
- http://www.w3schools.com/tags/tag_tfoot.asp
- http://www.w3schools.com/tags/tag_tr.asp
- http://www.w3schools.com/tags/tag_th.asp
- http://www.w3schools.com/tags/tag_td.asp

On the table

- Set the **border** attribute on the table to “2px”
- Set the **cellpadding** attribute on the table to “4px”
- Set the **cellspacing** attribute on the table to “0”

Note that whenever you set a non-zero value always add the units. However, if you set a value to 0, omit the units and just say 0. Attributes like these feel more like CSS properties than tag attributes. That’s because tables are really old and predate fancy CSS solutions.

Within the thead on the one and only tr

- Set the **bgcolor attribute** to a light gray such as **bgcolor=”#ddd”**

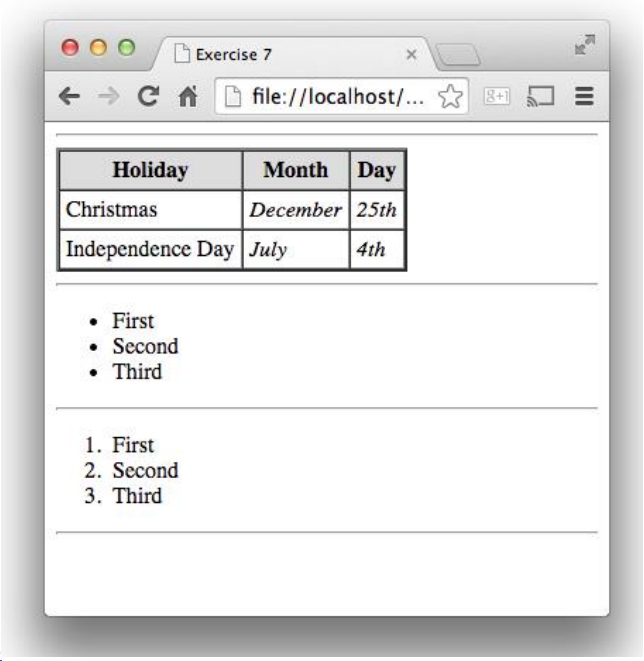
Note that these attributes is deprecated and shouldn’t be used. When we look at CSS, we will find better ways of accomplishing these things.

Within the tbody

Additionally within the last two columns of the last two rows, I used **i** (or **em**) tags for italics. Note, the only difference between things like **i** vs **em** or **b** vs **strong** is a connotative difference to screen readers. You could also use CSS to do font styling.

Below the table

Below the table is an unordered list and an ordered list. These should be fairly straightforward to figure out compared to all the details in the table. :)



- http://www.w3schools.com/tags/tag_ul.asp
- http://www.w3schools.com/tags/tag_ol.asp
- http://www.w3schools.com/tags/tag_li.asp

Notice the `hr` elements dividing the sections.

Exercise 8 (optional)

Let's talk about some new HTML5 tags. We'll start with some new tags that aren't as exciting, then show you some cooler new tags later. The less exciting tags are great for organizational purposes, but they don't really DO anything for you beyond what a basic div does already. Here are some examples of new organizational tags:

- http://www.w3schools.com/tags/tag_header.asp
- http://www.w3schools.com/tags/tag_nav.asp
- http://www.w3schools.com/tags/tag_article.asp
- http://www.w3schools.com/tags/tag_aside.asp
- http://www.w3schools.com/tags/tag_footer.asp

In the exercise8.html file create the page shown in the image below. Create a norml h1 then use each of the new tags too see the (not that exciting results). I'll get you started....

```
<h1>HTML5 tags to help screen readers (behave like divs)</h1>
<header>
  Header tag, for the top area of the page.
</header>
…
<footer>
  Footer tag, for the bottom area of the page.
<footer>
```

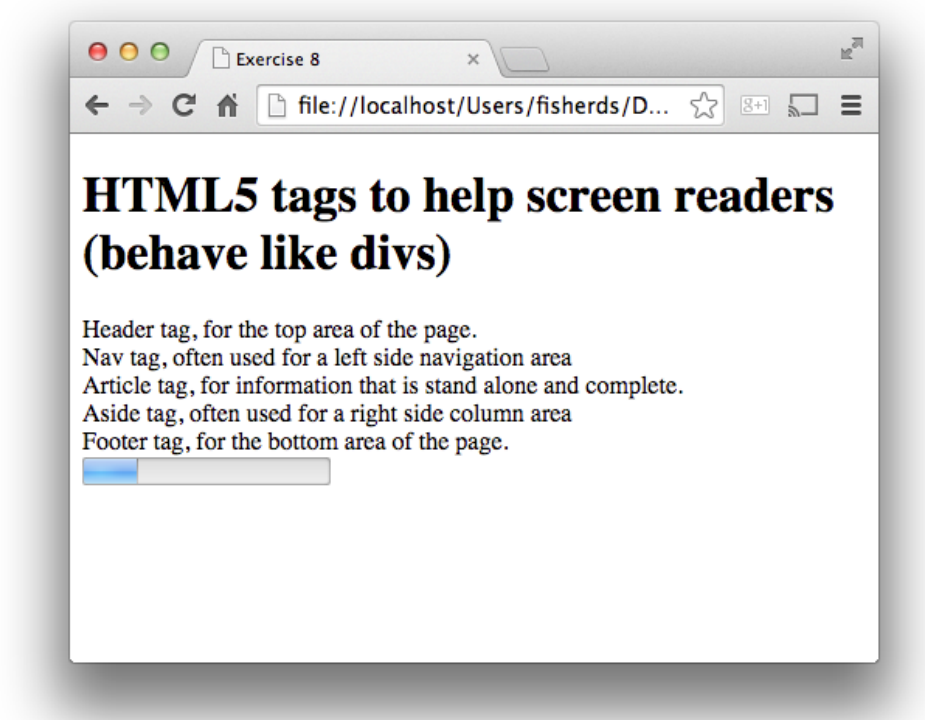
Then to make this introduction to HTML5 tags not completely boring add a progress tag for fun. At the bottom
http://www.w3schools.com/tags/tag_progress.asp
For example:

<**progress** value="22" max="100">

A **meter** tag is similar to a **progress** tag, but there are differences when using interaction (ie

click events that get fired)

Here is the solution with the progress tag added:



(ignore the fact that this version was taken on a Mac instead of a Windows machine)

Exercise 9 (Optional)

Two of the more impressive HTML5 tags are the **video** and **audio** tags. They have really cleaned up the mess that was web media playback on the web. If you've never done media display on the web before just assume it was always this easy.

Video tag

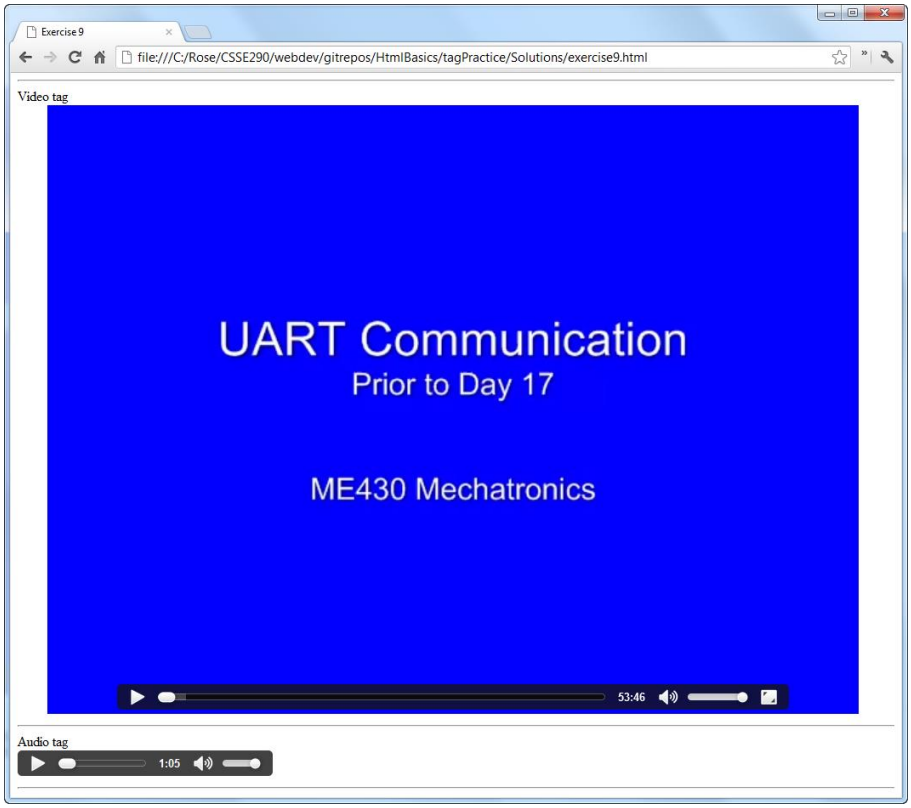
- http://www.w3schools.com/html/html5_video.asp
- http://www.w3schools.com/tags/tag_source.asp

Set the **width** attribute to 1024 and the **height** to 716 to match the video. Then display the standard **controls** by setting the **controls** attribute to "controls". Add only a single **source** tag in this example. Set the mime **type** attribute for that **source** to be "video/mp4" and set the **src** of the **source** to be:
http://www.rose-hulman.edu/class/me/HTML/ME430_Material_2_0/videos/Le17%20UART%20Communication/Le17%20UART%20Communication.mp4

Audio tag

- http://www.w3schools.com/html/html5_audio.asp

Set the **type** of the **source** to "audio/mp3" and the **src** of the **source** to http://www.rose-hulman.edu/users/groups/Chorus/download/FightSong.mp3



Exercise 10:

Find menu template to add link to html files for above Exercises and upload all files to host: Github, Amazon