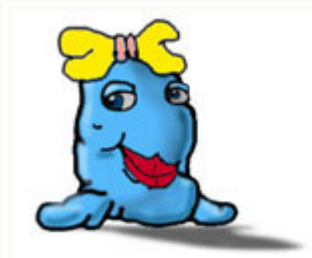


Assignment Eight: Fabulous Forms

link things? It is the same as though you were to link text, except you will place the `` and `` commands around the command for an image instead of text.

[My Updated Page #3](#)

I will link the following rough sketch of my mom to our class homepage:



Code Used

```
<a href="http://goldborder.com"></a>
```

You should notice a couple of things:

- When clicking on the image you are switched to our class homepage
- the image has a colored border around it that takes away from it being eye pleasing

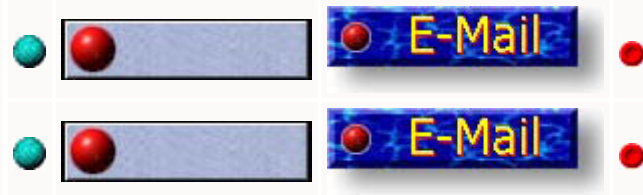
The first item (taking you to the class homepage is on purpose). The second item (colored border around image) is a default that Internet browsers do when you do not specifically tell it that you do not want it. The programmers who made the browsers assumed that you would want the image to stand out if it is clickable for the viewer. In this case, we don't want momma to be offended by the icky color surrounding her portrait, so...we will specifically tell the browser to not show a border:



Code Used

```
<a href="http://goldborder.com"></a>
```

Doesn't that look better? The additional code is the **border="0"**. You place this code as an attribute to the `` command: ``. This specifically tells the browser to not place any border around the image even if it is clickable. This comes in handy when you place graphical buttons on your page and you know the viewer understands that those are to be clicked. Here is a quick example of buttons with/without a border while being clickable:



You should notice the second row of buttons without borders looks



Far-Out Forms

This next level in your web based enrichment will deal with **forms**. Forms are used to send information to you the company/webmaster. The basic concept is, you create a form with options that the viewer can choose from, the viewer makes the choices they wish to select and send the form to you. This way the viewer/customer is able to inform you of their decisions on products/information, etc. You as the programmer will be able to format this information in a way that best deals with your needs.

The Assignment

At the end of this assignment you should be able to...

- [INTRODUCTION](#)

- (1) [Create a form](#)
- (2) [Use all the possibilities a form offers](#)
- (3) [Allow the viewer to send the form to you](#)

(1) Create A Form

You have used forms throughout your entire life. When you fill out an application for a job, a form to allow the hospital to do tests or enter a sweepstakes, you are using forms. In each of these cases the form has been designed so that you the customer can give information to the company but in a format that the company desires.

Could you imagine a company that handed people a blank piece of paper when they were looking to get hired and asked the person to just put down all their contact information, past employment, etc..? The company would receive this information in all sorts of organizational formats. John Doe may list his last name first and then his first name, Jane Smith would put her age first and not list her name until 1/2 way the paper. "What a mess!"

So this is how **forms** came about. When a company receives a form from a customer that has been filled in, they know that it will be in the same order/format as they designed (i.e. last name, first name, middle initial, phone number).



bolded line in the page that states the same message, something about a purchasing agreement for your pet.

[My Page #4](#)

Now we are going to create a **form** (are you jumping up and down with excitement?).

To create a form you simply place the **<form>** command inside your page where you want the form to start. Then you place all the nifty form

commands that create your form, and finally you place the `</form>` command at the end of your form.

Note: Even though it is possible to create more than one form in your web page, I would avoid doing so for confusing not only the viewer but also yourself when it comes to updating your code later. A good rule of thumb is, only one form per page.

Form Code

```
<form>  
... </form>
```

One last thing before you actually code. A form should have a name. You may not have know this yet, but you can name certain things in your web page so that you can refer to them later when you get into advanced coding. For instance, you can give an image a name. By doing so you can affect a certain image on your page without affecting the other images (hint: this is how programmers create buttons that change when you move your mouse over them). But for now, we are only going to name the form. To name a form, you place an attribute inside the `<form>` command called **name**:



to name a form in a manner that has something to do with the information you are receiving from the viewer. For instance, I am going to be selling a Snort Blaster to the viewer so I will name my form **snortsale**.

Note: When naming, it is better to not have any spaces in the name!

[My Updated Page #4](#)

Did you notice after you place the code that nothing change in the look of the page? That is because you have created a form but have not

the code by using the **VIEW SOURCE** option on your internet browser you would see the code for the form.

Even though there is nothing for the viewer to see, you have created your first form!



CONGRATULATIONS !!

(2) Use All The Possibilities A Form Offers

How about we put some questions on the form for the viewer? I think that would be a good idea.

When you see forms on paper, you should notice that there are different ways they can ask you to answer a question. For instance, when asking your name, the form will usually just have a nice long box or line for you to fill in. When asking your age, the form may have little round buttons (called radio buttons) with age ranges (i.e. 19-25, 26-34, 35 and older). Some forms use boxes so that you can mark multiple answers. These are usually used with questions that deal with products or information that you may not be limited to one area. For instance, a form could ask: "What are your hobbies". The form would then list the most common hobbies with boxes (called checkboxes) next to each hobby. You then could use your pen/pencil to check each hobby you enjoy.

The same holds true for forms you create on the web. You have all the following options when placing questions on your form:

1. Text Box (like asking for a name)
2. Radio Buttons (the circles)
3. Checkboxes (so someone can pick multiple choices)
4. Scrolling Lists
5. Hidden (will discuss this soon)
6. Text Area (to allow multiple lines in an answer)

As you can see, you have a good variety of methods on how to ask

commands. Here is an example:

```
<form>  
<input>  
</form>
```

We then place an attribute inside the `<input>` command that tells the browser what type of option (input) we are looking for.

Now let's get to bottom of these options.

7. Text Box

A text box will allow the viewer to type in a single line (or less) of information. One of the most common uses of this is the request for the viewers name. To produce a text box in the form we use the following command:

```
<input type="text size="25">
```

The **size** attribute tells the browser how large we want the text box. In this case, **25** means the box should be long enough to show 25 letters/numbers. This does not mean that the viewer is limited to 25 or that they must enter 25, it just means that when type in the box, they can see 25 at a time.

You should also name all your **input options** in your form. You will soon see why this is so important. Remember earlier I mentioned that a form should have a name that reflects what the form is about? The same holds true for naming the **input options**. You should name each option to reflect what information the viewer will be entering. The more specific you are in your naming the easier it will be for you to use the information later.

In this case I am going to create a **text box** on my form that is asking for the **first name** of the viewer. This is the code I am going to use:

```
<input type="text" name="firstName" size="25">
```

Note: Did you notice in the naming of my option that I used a capital 'N' in the name **firstName**? This was on purpose. When naming something with multiple words, it is good practice to use a capital letter in any words after the first word. This allows you to easily distinguish each word from each other. For example, if I wanted to name something **bestdangdogsintheworld**, it is hard to tell where words begin and end. But by using the capitalization rule, it is easier to read:



[My Updated Page #4](#)

Did you notice something missing? How is the viewer suppose to know what to type in the box?

You can place text in front or after **input options** just as you do with any other information. By placing the phrase **First Name** before the box, the viewer understand what you are asking for.

For Example:

```
First Name <input type="text" name="firstName" size="25">
```



Place the phrase **First Name** before your **input option**.

[My Updated Page #4](#)

Note: You should have a space between text and your **input options**. This makes the form easier on the eyes for your viewer. Make sure you have a space between the **First Name** text and the **<input ...>**



Now ask for the viewers last name, street address, city, state and zip. For now I want the questions placed on their own lines for readability. You should name them **lastName**, **streetAddress**, **city**, **state** and **zip**.

[My Updated Page #4](#)

You should notice a couple things that do not look right.

First, that the boxes for state and zip are too long for the information that would be entered in those. You would want a person living in **Washington** to enter the initials **WA**. By having such a big box, your viewer thinks you want the state name typed out. The same holds true for the zip. A zip code is usually 5 number.



ttributes

outside of the text box, making it easier to read for the viewer. In this case, I would set the **state** size to three (3) and the **zip** size to six (6).

[My Updated Page #4](#)

Second, the form doesn't look very organized when the input boxes are not aligned or the text. It looks like we just slapped the form together instead of thinking about readability. To make your form easier to read, we are going to use tables (remember I said the `<table>` command would be one of the most useful ones you use).

To use a table and make the form look nicer, I am going to place each question on its own row (`<tr>`), and also place the text in a separate cell (`<td>`) from the text box. This will cause the text and boxes to line up nicely. Here is what I am code to accomplish this:


```
<table border="0" cellpadding="3">
  <tr>
    <td>First Name</td>
    <td><input type="text" name="firstName" size="25"></td>
  </tr>
  <tr>
    <td>Last Name</td>
    <td><input type="text" name="lastName" size="25"></td>
  </tr>
  <tr>
    <td>Street Address</td>
    <td><input type="text" name="street" size="25"></td>
  </tr>
  <tr>
    <td>City</td>
    <td><input type="text" name="city" size="25"></td>
  </tr>
  <tr>
    <td>State</td>
    <td><input type="text" name="state" size="3"></td>
  </tr>
  <tr>
    <td>Zip</td>
    <td><input type="text" name="zip" size="6"></td>
  </tr>
</table>
```



Place a table around your question on your form to organize them. Remember to remove the
 commands. You will not need them since you will be place the question in their own rows.

[My Updated Page #4](#)



Pretty nice looking form!

Before we move on to the other options, let's apply some more techniques to our table/form so that we can really make it look organized and easy for the viewer to understand.

The next thing you can do to your form is to align the text before each input option to the right. This will line up all the text nicely and place the text next to the box which reaffirms what you are asking for. To accomplish this, you place an attribute inside the cell command for the text for aligning the cell information to the right. For example:

```
<td align="right">
```



trying to perform.

[My Updated Page #4](#)

Now to further show that the text are just titles and the text boxes are for input, we can put a background color in the cells for the text. We accomplish this with the **bgColor** attribute in the cell command. For Example:

```
<td align="right" bgcolor="#DDDDFF">
```



Put the **bgcolor="..."** attribute in the cell command for all the text in your table on your form.

[My Updated Page #4](#)

The last thing we are going to do to the look of the table for now is to give it a border. This will better define the cells and make viewing



he

[My Updated Page #4](#)

Just so you can see the improvement, here are the two tables from my page. The first is how we started, the second has all the techniques applied for better viewing:

First Name	<input type="text"/>	First Name	<input type="text"/>
Last Name	<input type="text"/>	Last Name	<input type="text"/>
Street Address	<input type="text"/>	Street Address	<input type="text"/>
City	<input type="text"/>	City	<input type="text"/>
State	<input type="text"/>	State	<input type="text"/>
Zip	<input type="text"/>	Zip	<input type="text"/>

Radio Buttons

One of the hardest options in a form to program is the **radio button** (bet you didn't want to hear that). The reason is that the programmer doesn't take the time to learn the basic concept of radio buttons. Well, I'll teach you that concept first and then we will dive right in.

Radio Button Concept: When placing radio buttons on your page you need to remember three (3) fundamental rules. By understanding these rules you can use radio buttons easily. Forget the rules and they will become a nightmare to use or you will avoid them all together, thus not using a very valuable tool.

- **Rule #1: (grouping)**

Radio buttons are grouped together by their name. In the earlier section on **text boxes**, you used a different name for each box. When it comes to radio buttons you use the same name on groups of buttons. This grouping allows the browser to control which button is used when the forms information gets sent to you. For instance, let's say you want to find out what age group the viewer is in. You create appropriate radio buttons with the groups of ages you want defined but give each radio button the same name. The basic radio button code is:

```
<input type="radio" name="..." value="...">
```

You already understand what the **name** attribute is used for. The new attribute is the **value**. The reason for this attribute in the radio button is that the viewer never enters information. The viewer just clicks on the radio button located next to the choice they desire. It is up to you to define what information (value) will be sent to you when they choose a certain button.

Example:

	Code Used
<input type="checkbox"/> 19-25	<pre><input type="radio" name="radioExampleAge" value="19-25"></pre>
<input type="checkbox"/> 26-34	<pre>
 <input type="radio" name="radioExampleAge" value="26-34"> 26-34</pre>
<input type="checkbox"/> 35-99	<pre>
<input type="radio" name="radioExampleAge" value="35-99"> 35-99</pre>

Go ahead and click on the radio buttons one at a time. You'll notice the browser will uncheck any other radio button in that group when you check another one. So if you click on the 19-25 age range and then click on the 26-34, the 19-25 age range radio button becomes unchecked. Only **one** radio button in a group can be checked at a time. How did it know to do this?

The browser knows because you gave each radio the same name. In essence you are telling the browser that all three radio buttons are tied together but have different values. When the viewer finally send on the form to you, whichever radio button is highlighted will send its value to you. So if the 35-99 radio button is pushed it will send a value of '35-



[My Updated Page #4](#)

You should have notice that instead of place the radio button on separate lines (like the example above), I place them next to each other. I do this since people are used to seeing them in this manner. Look on most paper forms and you will see radio buttons like this shown left to right, instead of up/down. Remember you want the viewer to feel comfortable filling out your form.

How about another group of radio buttons? As long as you keep groups of radio buttons separated by their names, you can use as many groups as you wish. In the following example I am adding another group of radio buttons that asks what planet the viewer is from along with asking their age:

	Code Used
<input type="checkbox"/> 19-25	
<input type="checkbox"/> 26-34	<pre><input type="radio" name="radioExampleAge" value="19-25"> 19-25</pre>

```
 35-99 <br> <input type="radio" name="radioExampleAge" value="26-34"> 26-34
 Earth <br> <input type="radio" name="radioExampleAge" value="35-99"> 35-99
 Mars <br> <input type="radio" name="radioExamplePlanet" value="The Blue One"> Earth
 Tatooine <br> <input type="radio" name="radioExamplePlanet" value="Lots Of Dust"> Mars
<br> <input type="radio" name="radioExamplePlanet" value="Luke Skywalkers Home"> Tatooine
```

I want you to notice a couple of things about the second group. First, I named the second set of radio buttons differently (radioExamplePlanet instead of radioExampleAge). This allows the viewer to select a radio button in each group without affecting the other group. Go ahead, click on an age group, then click on a planet. Notice how the clicking in one group does not affect the other? Now click on another planet. Notice the browser takes care of unchecking any other planets checked but doesn't change the age group radio buttons?

- Rule #2: (values)

Second, do you notice the values for the second group are "The Blue One", "Lots Of Dust" and "Luke Skywalkers Home"? This is another important train of thought. What the viewer sees next to the radio button can be totally different than what information will be sent to you when the form is sent. In this case, if the viewer had selected the planet called 'Tatooine' and sent the form, you would receive information that looks like:

radioExamplePlanet = Luke Skywalkers Home

It would then be up to you what you do with that information.



will use the hex code. So when I receive the forms information, I will receive hex code values for the color they desire. You will probably want a name like 'animalColor'.

<http://goldborder.com>

