**Forms**

HTML Forms are one of the main points of interaction between a user and a web site or application. They are a type of widget (a small computer application that enables a user to perform a small function) allow users to send data to the web site, and website owners to harvest this data. Most of the time that data is sent to the web server, but the web page can also intercept it to use it on its own. The main difference between an HTML form and a regular HTML document is that, most of the time, the data collected by the form is sent to a web server. In that case, you need to set up a web server to receive and process the data. How to set up such a server is beyond the scope of this course.

**Some Helpful Links**

[Code Academy: Web Form – Hands-On Mini-Course](https://www.codecademy.com/courses/web-beginner-en-Vfmnp/0/1)

[W3Schools – All About Forms](http://www.w3schools.com/html/html_forms.asp)

[Sending Form Data – Web server info if you’re curious](https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Forms/Sending_and_retrieving_form_data)

**Tips For Designing Your Forms:**

* Prior to coding your form, create a quick mockup on paper to identify the set of data you will **need** (only ask for what you need – keep it to-the-point) to acquire from your user
* The bigger your form, the greater the risk of losing your user (again, ask only for what you need)
* The more simplistic and uniform your formatting, the easier it is for your user to read and understand (just like anything else on your website)

**Common Types of Forms**

* Contact (name, return email, message box, send button)
	+ **FYI: Email Button:** If you want to keep things REALLY basic when it comes to creating a Contact Us form, you can always stick to the simple html link <a href=“url”></a> method… except instead of using href=“url” replace the “url” with “mailto:emailaddress”!
		- Example: <a href="mailto:bhammond@bpsd.mb.ca">Send Us an Email!</a>
* Order form (name, return email, checkbox/radio button options)
* Comment form (name, return email, message box, send button)
* Poll/Survey (name, checkbox/radio button options, submit button)
	+ Try creating a poll for your website using <http://pollcode.com/> (The cool thing about this site is that it allows you to choose your options and it generates your code for you – and it uses an html form, just like what you’re learning, to do it! Unlike the forms you create, however, pollcode’s forms are connected to a Web Server, which is how users are able to submit and view data.)

**Coding an HTML Form**

***The Tags***

(all of which are container, meaning they have an “open” and a “close” tag, except for the <input> tag, which is an empty tag, meaning that is has no “close” tag):

* <form> - the initial tag needed to define a form; wraps all other tags within it
	+ “action” attribute: defines the location (URL) where the form’s data will be sent
	+ “method” attribute: defines which HTTP method (get or post) is used to send the data
	+ When no “action” or “method” is specified, the data is automatically sent to the same page that contains the form… but it cannot be parsed without a server-side language
* <input> - defines a single-line input field
	+ “type” attribute: extremely important; defines the way the <input> element behaves
		- “text” – default value that allows for simple plain text entry as an input
		- “email” – value that only accepts properly formatted email addresses as input
		- “password” – value that only accepts passwords; input is automatically masked
		- “date” – value that only accepts dd/mm/yyyy-style dates as input
		- “color” – value that provides a color picker as input
		- “range” – value provides a spectrum-like slider as input
		- “month” or “week” – value provides a calendar as input
		- “radio” – value defines radio button input (good for single-choice selection)
			* Add the “checked” attribute inside <input> to auto-check an option
		- “checkbox” – value defines checkbox input (good for multiple-choice selection)
			* Add the “checked” attribute inside <input> to auto-check an option
		- “submit” – creates a button that, when clicked, submits form data to web server
	+ “name” attribute: defines the name of the input field; it is important to name each input field because the “name” attribute is required for any JavaScript you might want to add
	+ “value” attribute: allows you to set default text to appear in the input field until the user interacts with it, thereby replacing the text with his/her own
	+ “size” attribute: defines the width (in characters, rather than pixels) of the input line
	+ “autofocus” attribute: defines a single element that should be the user’s primary focus by automatically highlighting the important input field, then highlighting any other input field that is selected thereafter
* <textarea> - defines a multi-line input field
	+ “name” attribute: defines the name of the input field
	+ “value” attribute: allows you to set default text to appear in the input field until the user interacts with it, thereby replacing the text with his/her own
* <button> - allows the user to send their data; there are 3 types:
	+ *<button> tags are optional – the same effect can be achieved with the “type=’submit’” attribute of the <input> tag!*
	+ “type” attribute: extremely important; defines the way the <input> element behaves
		- “submit” – sends the form’s data to the page defined by the “action” attribute
		- “reset” – resets all the form input fields to their default values immediately
		- “button” – does nothing… until teamed up with JavaScript
* <fieldset> - groups related data inside a form box
* <legend> - adds a title within the form box generated by the <fieldset> tag
* <div> - can be placed between <input>s in order to conveniently and neatly structure the code within your forms to make it easier to style (via CSS) later on. Of course, classes/ids may also be added inside each div if you would like to customize each individual section of your form.

***The Explanation***

* All forms are defined by the <form></form> tag, which is a container tag just like <ul></ul> or <ol></ol> tags used to define a list or <table></table> tags used to define a table.
	+ Though all attributes associated with the <form> tag are optional, it is best practise to define at least the “action” and “method” attributes if you plan on linking your form up with a web server (which, in our case, we are not)
* Each separate section of your form is called a “field” or “input field.” This is where your user will input, or type, his/her information or answers. For short, single-word answers (like a name or email address) it is best to use <input> tags to define the field, since it allows for space-saving single-line input. For long, multi-word answers (like a message or idea) it is best to use <textarea> tags to define the field, since it allows for larger, multi-line input.
* The <button> tag typically comes at the bottom of the form and is used to create a button that your user can click on to submit their field input responses
* You can use divs and classes throughout your form’s html code to produce specific elements, which you can later style (just like any other html element) using CSS.

[Bonus Material: How to make a drop-down list form](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_elem_select)

***The Example***

<form action=”/my-form-handling-page” method=”post”>

<fieldset class=”favorites”>

<legend>Favorites</legend>

<div>

<p>Favorite Color:</p>

<input type="text" name="color" value="Mine is purple!"><br>

 </div>

<div>

 <p>Favorite Food:</p>

 <input type="text" name="food" value="Mine is poutine!"><br>

 </div>

 <div>

 <p>Explain your choices in the space below</p>

 <textarea name="explanation"></textarea><br><br>

 </div>

 <div>

</fieldset>

<fieldset class=”Extras”>

<legend>Extras</legend>

 <p>Gender:</p>

 <input type="radio" name="gender">Female<br>

 <input type="radio" name="gender">Male<br>

<input type="radio" name="gender">Other<br>

 </div>

 <div>

 <p>Types of Pets:</p>

 <input type="checkbox" checked name="pets">Dog(s)<br>

 <input type="checkbox" name="pets">Cat(s)<br>

 </div>

</fieldset>

 <div>

 <input type="submit">

 </div>

</form>