**HTML Form Element**

Forms in HTML allow for users/clients to interact with your application/website by inputting data. We create form elements using <form></form> tags and then fill the form with a selection of different types of control elements.

The following list references the HTML Forms Guide, excellent documentation written by the Mozilla Developer Network. Teaching and remembering every aspect of HTML would be both difficult and fairly unnecessary. Instead we should get used to being able to quickly search correct HTML syntax and practice reading online documentation.

The table contains links to references about the most popular HTML form elements, their most popular attributes and some of those attribute's most popular values. Read through the docs as well as the notes below and then use both to complete tonight's exercise.

| **Popular Elements** | **Popular Attributes** | **Popular Values** |
| --- | --- | --- |
| [Form](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/form) | [action](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/form#attr-action), [method](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/form#attr-method) | Action takes a URI like "/example.com" and Method takes a HTTP request method of either GET or POST. |
| [Input](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input) | [type](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input#attr-type) | button, checkbox, color, date, email, hidden, number, password, radio, submit, text, url |
| [Input](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input) | [checked](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input#checked), [name](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input#name), [placeholder](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input#placeholder), [value](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input#value) | Checked is either true or false. Name, placeholder, and value will be equal to a string. |
| [Label](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/label) | [for](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/label#for) | The for attribute takes a string corresponding to the id attribute of an element being labeled. |
| [Select](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/select) | name, disabled | Disabled is either true or false. Name will be equal to a string. |
| [Option](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/option) | value, selected, disabled | Disabled/Selected is either true or false. Value is equal to a string. |
| [Textarea](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/textarea) | maxlength, minlength, name, rows, cols, wrap, spellcheck | These values depend on the attribute but are usually just a string of text or numbers. |

**Notes**

Some Notes on these popular HTML Elements:

* Input Elements are by far the most common and special element of a form because they can take on completely different functionalities depending on their attribute type.
* Labels can be written two ways, by wrapping the input element or by using the for attribute and applying a corresponding id attribute to the input.
* Submitting forms makes a default post request with parameters made up by the provided values for each name attribute.
* By giving each radio input element the same name attribute value, the radio buttons will only allow the user to select one per radio of that name per form.
* Option elements are defined inside a select element and the value attributes for each option display what will be inside the dropdown.

**Form Example**

Below is an example of a form that would be used on a typical signup page for a website:

<form action="http://example.com/signup" method="POST">

<label for="username">Username</label>

<input type="text" id="username" name="username">

<br />

<label>Password

<input type="password" name="password">

</label>

<br />

<input type="submit" value="Sign Up!">

</form>

The action of the form is a URL where it will be submitted. At this point, we won't go over everything involved in a HTTP request, but in short, the information provided by the inputs will be sent in a request to the URL provided.

We can see two styles of using a label, in the first example the for attribute is equal to the id of the corresponding input, and in the second example the label is wrapped around the input. Both are valid ways of labelling an input. The purpose of the label is just to associate text to the input field. Also, if a user clicks on the label, the corresponding input will be selected.

The above example shows three types of inputs: text, password, and submit. Text inputs will show up as a box which a user can type in. A password input is the same as a text input except the text is hidden by black dots. A submit input looks like a button, which the user can not type into, and the value will be the text on the button. There are many other types of inputs, as listed in the table, and they all have their own differences. I recommend making a form and testing them out for yourself!