**HTML Basics #4: Forms**

Forms is a very important topic in HTML, it allows us to collect user input from the client, and send it to the server to be processed. All forms in HTML are wrapped inside the <form> element like this:

<form>

.

form elements

.

</form>

The <form> element act as a container for different types of input elements, such as text fields, checkboxes, radio buttons, submit buttons, etc. We’ll talk about all of them in this article, but first, let’s start by discussing some attributes that are associated with the <form> element.

Table of Contents

* [Form Attributes](https://www.ericsdevblog.com/index.php/2021/05/html-basics-4-forms/#Form_Attributes)
  + [The action Attribute](https://www.ericsdevblog.com/index.php/2021/05/html-basics-4-forms/#The_action_Attribute)
  + [The target Attribute](https://www.ericsdevblog.com/index.php/2021/05/html-basics-4-forms/#The_target_Attribute)
  + [The method Attribute](https://www.ericsdevblog.com/index.php/2021/05/html-basics-4-forms/#The_method_Attribute)
    - [Difference Between GET and POST](https://www.ericsdevblog.com/index.php/2021/05/html-basics-4-forms/#Difference_Between_GET_and_POST)
  + [The autocomplete Attribute](https://www.ericsdevblog.com/index.php/2021/05/html-basics-4-forms/#The_autocomplete_Attribute)
  + [The novalidate Attribute](https://www.ericsdevblog.com/index.php/2021/05/html-basics-4-forms/#The_novalidate_Attribute)
* [Form Elements](https://www.ericsdevblog.com/index.php/2021/05/html-basics-4-forms/#Form_Elements)
  + [The <input> Element](https://www.ericsdevblog.com/index.php/2021/05/html-basics-4-forms/#The_Element)
  + [The <label> Element](https://www.ericsdevblog.com/index.php/2021/05/html-basics-4-forms/#The_Element-2)
  + [The <select> Element](https://www.ericsdevblog.com/index.php/2021/05/html-basics-4-forms/#The_Element-3)
  + [The <textarea> Element](https://www.ericsdevblog.com/index.php/2021/05/html-basics-4-forms/#The_Element-4)
  + [The <button> Element](https://www.ericsdevblog.com/index.php/2021/05/html-basics-4-forms/#The_Element-5)
  + [The <fieldset> and <legend> Elements](https://www.ericsdevblog.com/index.php/2021/05/html-basics-4-forms/#The_and_Elements)
* [Input Types](https://www.ericsdevblog.com/index.php/2021/05/html-basics-4-forms/#Input_Types)

**FORM ATTRIBUTES**

**The action Attribute**

<form action="/action\_page.php">

.

form elements

.

</form>

The action attribute defines the action to be taken when the form is submitted. Usually, the data in the form will be sent to the server, and that data will be processed by a program. In this example, the program is called “action\_page.php“. After the data is processed, there will usually be a response, sent from the server back to the client.

The will make more sense when we get to the backend part of this web development course.

**The target Attribute**

<form action="/action\_page.php" target="\_blank">

.

form elements

.

</form>

The target attribute specifies where to display the response that was sent from the server. It works just like the target we talked about in links (<a>).

**The method Attribute**

<form action="/action\_page.php" method="get">

.

form elements

.

</form>

The method attribute specifies the HTTP method to be used when submitting the form data. Like we talked about at the very beginning of this course, the two most common methods are GET and POST.

**Difference Between GET and POST**

They are both able to send the form data to the server, the difference is that POST will put the data inside the body of the HTTP request, while GET will append the data to the URL, in name/value pairs. Which makes POST a safer way to send information.

In the example above, the GET method will send the data in URL like this:

www.example.com/action\_page.php?name1=value1&name2=value2

All the data are appended after the question mark (?), and different name/value pairs are joint by the ampersand (&).

As for the POST method, the data is “hidden” in the HTTP request body like this:

POST /action\_page HTTP/1.1

Host: www.example.com

name1=value1&name2=value2

**The autocomplete Attribute**

<form action="/action\_page.php" autocomplete="on">

.

form elements

.

</form>

The autocomplete attribute specifies whether a form should have autocomplete on or off. When autocomplete is on, the browser automatically completes values based on values that the user has entered before.

**The novalidate Attribute**

<form action="/action\_page.php" novalidate>

.

form elements

.

</form>

The novalidate attribute is a Boolean attribute. When present, it specifies that the form-data (input) should not be validated when submitted.

**FORM ELEMENTS**

**The <input> Element**

The <input> element is the most important element in HTML forms. It can be displayed in several different ways depending on its type. We’ll talk about all of them in the next section.

<form>

<input type="text">

</form>

Application

Description automatically generated with medium confidence

**The <label> Element**

The <label> element can be used to define labels for any form elements. This is very useful for screen-reader users, and it improves your website’s accessibility, which is a very important factor in website ranking.

The <label> element has a for attribute, and it should match the id of the form element that it is trying to label.

<form>

<label for="firstname">First Name:</label><br>

<input type="text" id="firstname"><br>

<br>

<label for="lastname">Last Name:</label><br>

<input type="text" id="lastname"><br>

</form>

Table

Description automatically generated with low confidence

**The <select> Element**

The <select> element defines a drop-down list like this:

<label for="fruits">Pick A Fruit:</label>

<select id="fruits" name="fruits">

<option value="Apple">Apple</option>

<option value="123">Banana</option>

<option value="eprag">Grape</option>

</select>

A picture containing diagram

Description automatically generated

In this example, each <option> element defines an option that can be selected, and it has a value attribute and content. The content will be displayed and the value of the selected option will be sent to the backend when the form is submitted. The content and the value do not have to match each other. For instance, if the Apple option is chosen, the value "Apple" will be sent to the server, and if the Banana option is chosen, the value "123" will be sent.

By default, the first option of the list is chosen, you can change that by adding the selected attribute to another option:

<select id="fruits" name="fruits">

<option value="Apple">Apple</option>

<option value="123">Banana</option>

<option value="eprag" selected>Grape</option>

</select>

You can also use the size attribute to specify the number of visible values:

<select id="fruits" name="fruits" size="2">

<option value="Apple">Apple</option>

<option value="123">Banana</option>

<option value="eprag">Grape</option>

</select>

A picture containing graphical user interface

Description automatically generated

And now, because we can see multiple options, we can choose multiple options using the multiple attribute:

<select id="fruits" name="fruits" size="2" multiple>

<option value="Apple">Apple</option>

<option value="123">Banana</option>

<option value="eprag">Grape</option>

</select>

**The <textarea> Element**

The <textarea> element defines a input box that allows you to have multiple lines:

<textarea name="message" rows="10" cols="30">

Lorem ipsum dolor sit amet...

</textarea>

Text

Description automatically generated

The rows and cols attribute specify the size of the input box.

**The <button> Element**

The <button> element defines a clickable button:

<button type="button">Click Me!</button>

This button has the type button, which means when you click on it, it doesn’t do anything, but don’t worry, we’ll use it to do some interesting stuff once we get to the JavaScript part of this course.

**The <fieldset> and <legend> Elements**

The <fieldset> can group different input fields together, and <legend> creates a name for that group:

<form>

<fieldset>

<legend>Personalia:</legend>

<label for="firstname">First Name:</label><br>

<input type="text" id="firstname"><br>

<br>

<label for="lastname">Last Name:</label><br>

<input type="text" id="lastname"><br>

</fieldset>

<fieldset>

<legend>Add to Shopping Cart:</legend>

<label for="fruits">Pick A Fruit:</label>

<select id="fruits" name="fruits">

<option value="Apple">Apple</option>

<option value="123">Banana</option>

<option value="eprag">Grape</option>

</select><br>

<br>

<label for="vegetables">Pick A Vegetable:</label>

<select id="vegetables" name="vegetables">

<option value="098">Ginger</option>

<option value="987">Kale</option>

<option value="876">Eggplant</option>

</select>

</fieldset>

<br>

<button type="button">Click Me!</button>

</form>

Graphical user interface, text, application

Description automatically generated

**INPUT TYPES**

Like we just said, there are many different types for the <input> element, as shown in this example:

<form>

<fieldset>

<legend>Input Type Text:</legend>

<label for="name">Name:</label>

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

</fieldset>

<fieldset>

<legend>Input Type Email:</legend>

<label for="email">Enter your email:</label>

<input type="email" id="email" name="email">

</fieldset>

<fieldset>

<legend>Input Type Password:</legend>

<label for="pswd">Password:</label>

<input type="password" id="pswd">

</fieldset>

<fieldset>

<legend>Input Type Radio:</legend>

<input type="radio" id="male" name="gender" value="male">

<label for="male">Male</label><br>

<input type="radio" id="female" name="gender" value="female">

<label for="female">Female</label><br>

<input type="radio" id="other" name="gender" value="other">

<label for="other">Other</label>

</fieldset>

<fieldset>

<legend>Input Type Checkbox:</legend>

<input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">

<label for="vehicle1"> I have a bike</label><br>

<input type="checkbox" id="vehicle2" name="vehicle2" value="Car">

<label for="vehicle2"> I have a car</label><br>

<input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">

<label for="vehicle3"> I have a boat</label>

</fieldset>

<fieldset>

<legend>Input Type Color:</legend>

<label for="favcolor">Select your favorite color:</label>

<input type="color" id="favcolor" name="favcolor">

</fieldset>

<fieldset>

<legend>Input Type Date and Date Time:</legend>

<label for="birthday">Birthday:</label>

<input type="date" id="birthday" name="birthday"><br><br>

<label for="birthdaytime">Birthday (date and time):</label>

<input type="datetime-local" id="birthdaytime" name="birthdaytime">

</fieldset>

<fieldset>

<legend>Input Type File:</legend>

<label for="myfile">Select a file:</label>

<input type="file" id="myfile" name="myfile">

</fieldset>

<fieldset>

<legend>Input Type Number:</legend>

<label for="quantity">Quantity (between 1 and 5):</label>

<input type="number" id="quantity" name="quantity" min="1" max="5">

</fieldset>

<fieldset>

<legend>Input Type Submit and Reset:</legend>

<input type="submit">

<input type="reset">

</fieldset>

</form>

Graphical user interface, application

Description automatically generated

| **Type** | **Description** |
| --- | --- |
| Text | Defines a single-line text input field. |
| Email | Defines an input field that should contain an e-mail address. The e-mail address can be automatically validated when the form is submitted. |
| Password | Defines a password field. |
| Radio | Defines a radio button. |
| Checkbox | Defines a checkbox. |
| Color | Defines a input field that should contain a color. When you click on the field, a color picker should show up. |
| Date and Date Time | Used for selecting dates and times. |
| File | Defines a file-select field and a “Browse” button for file uploads |
| Number | Defines a numeric input field. |
| Submit | Defines a button for submitting form data. |
| Reset | Defines a reset button that will reset all form values to their default values. |