Glossary: HTML form elements

The **<form>** element in HTML is an important and useful element. The following sheet provides an overview of the **<form>** constituent elements and their common attributes with simple examples for quick reference.

**<input>**

It is used to create interactive controls, for example, buttons and various types of text fields and so on, to accept input or data from the user. The key attribute of this element is **type**. Some common values for the **type** include: **button, checkbox, date, email, number, password, submit, text,** and **url**. These values dictate the appearance of the element. For example, this code:

</form>

  <br>

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

  <br><br>

  <br>

  <label for="pwd">Password:</label>

  <label for="uname">Username:</label>

  <br>

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

<form action="my\_action\_page">





Results in the following output:



Note how the type **password** hides the user input.

**<label>**

Defines a label for an element. It has an attribute "for", the value of which should be equal to the id attribute of the element it is associated with. Note how in the example above, the **<label>** is associated with the **<input>** using its id value.

**<select>**

Defines a drop-down list of options presented to the user. It has a couple of attributes:

* Form, the id of the form in which the drop-down appears
* Name specifies the name of the control
* Multiple Boolean attribute, when specified, indicates if a user can select multiple options out of the list
* Required indicates if the user is required to select an option before submitting a form
* Size mentions the number of visible options in a drop-down list

The options in a drop-down list are defined using the **<option>** element inside **<select>**. Note the example in the **<option>** description below.

**<textarea>**

Defines a multi-line input field, typically to allow the user to input longer textual data. The common attributes for this element include:

* **cols** defines the width of the text area, the default value is 20
* **form** the form element the text area is associated with
* **maxlength** when specified, limits the maximum number of characters that can be entered in the text area
* **minlength** the minimum number of characters that the user should enter
* **readonly** once set, the user cannot modify the contents
* **rows** defines the number of visible text lines for the text area

The following line of code defines a text area of 10 visible lines and nearly 30 characters wide where the user can input a maximum of 200 characters:

<textarea name="response" rows="10" cols="30" maxlength=”200”>

</textarea>





**<button>**

Defines a clickable button. The **onclick** attribute defines the behavior when the button is clicked by the user. For example, in the code below, an alert message is shown to the user.

<button type="button" onclick="alert('You just clicked!')">Click Me!

</button>





**<fieldset>**

Used to group related input elements in a form. For instance, elements related to the user’s personal information and educational qualification can be grouped separately in two field sets.

**<legend>**

Defines a caption for the **<fieldset>** element. For example:

<fieldset>

  <legend>Personal Info</legend>

  <label for="fname">First name:</label><br>

  <input type="text" id="fname" name="fname" value="John"><br>

  <label for="lname">Last name:</label><br>

  <input type="text" id="lname" name="lname" value="Doe"><br>

</fieldset>

<fieldset>

  <legend>Qualificaiton</legend>

  <label for="pdegree">Primary degree:</label><br>

  <input type="text" id="pdegree" name="degree" value="Masters"><br>

  <label for="fos">Last name:</label><br>

  <input type="text" id="fos" name="field" value="Psychology"><br>

</fieldset>





**<datalist>**

Specifies a list of pre-defined options for an input element. It differs from **<select>** since the user can still provide textual or numeric input other than the listed options.

<form action="/my\_action\_page">

  <label for="flowers">Favourite flower:</label><br>

  <input list="flowers" name="flowers">

  <datalist id="flowers">

    <option value="Rose">

    <option value="Lily">

    <option value="Tulip">

    <option value="Daffodil">

    <option value="Orchid">

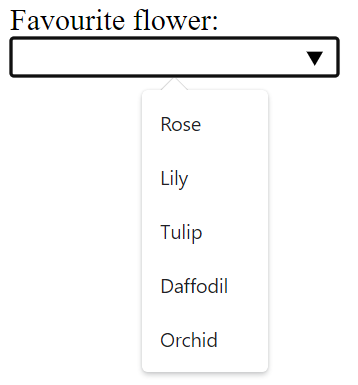
  </datalist>

  ...

</form>







**<output>**

Represents the result of a calculation (typically the output of a script) or the outcome of the user action.

**<option>**

Defines an option for the drop-down list. The following code example demonstrates how a simple list can be defined, with the rendered view below the code block.

<label for="course">Choose a course:</label><br>

<select id="course" name="courselist">

  <option value="html">HTML Introduction</option>

  <option value="css">Styling with CSS</option>

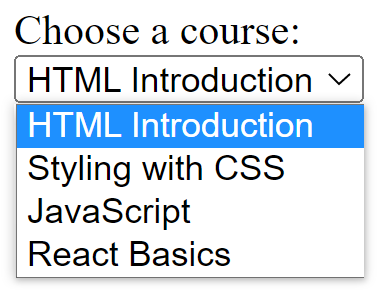
  <option value="js">JavaScript</option>

  <option value="react">React Basics</option>

</select>







By default, the first item in the drop-down list is selected. To define a pre-selected option, add the selected attribute to the option.

**<optgroup>**

Defines a group of related options in a drop-down list. Its attribute label names the group.

# Create a complex form (solution)

The following code is an example solution for the previous exercise.

This code is placed inside the <main> element of index.html.

<form>

    <div>

        <label for="email\_address">Email Address</label>

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

    </div>

    <div>

        <label for="booking\_date">Date of Booking</label>

        <input type="date" id="booking\_date" name="booking\_date">

    </div>

    <div>

        <label for="people">Number of people</label>

        <input type="number" id="people" min="1" max="8" name="people">

    </div>

    <div>

        <label>

            <input type="checkbox" id="agree" name="agree" required>

            I agree to the cancellation policy

        </label>

    </div>

    <button type="submit">Book Now</button>

</form>

While reviewing the code, note the following:

* The *type* attribute is set to *email* for the email address field. This will enable client-side validation to ensure that the user enters a correctly formatted email address.
* The *type* attribute is set to *date* for the booking date field. This will enable the browser's built-in date picker for the field.
* The *type* attribute is set to *number* for the number of people field. This will use the browser's built-in number picker for the field.
* The *min* attribute is added to the people input element to set the minimum value to 1.
* The *max* attribute is added to the people input element to set the maximum value to 8.
* The checkbox input element is contained inside the label element. For mobile devices, this will improve the user experience so that the user can touch either the checkbox or the text "I agree to the cancellation policy" to toggle the checkbox.
* The checkbox input element has a *required* attribute. This will require that the user agrees to the cancellation policy before being able to book a table.

# Images

This lesson will help refresh your knowledge of the **<img>** tag and how you can use it to embed images in webpages. The **<img>** tag is used to add an image to a web page. The image’s address is specified using the src attribute. For example, if you wanted to embed an image file named photo.png, you can do that with the following HTML. **<img src="photo.png">** You can also specify the width and height of the image using the width and height attributes. For example, if the width of the photo is 640 pixels and the height of the photo is 480 pixels, you can use the following HTML. **<img src="photo.png" width="640" height="480">** It is important to note that you can set the image to a larger or smaller size and the web browser will automatically scale the image. For example, you can update the previous HTML to half the width and height and the image would shrink by 50%. **<img src="photo.png" width="320" height="240">** One useful feature of rendering images in the web browser is that the web browser can automatically keep the correct ratio of width to height. So for example, if you want to scale the image by 50%, you can simply set the width attribute and the web browser will automatically calculate the height. **<img src="photo.png" width="320">** But what happens if the photo doesn’t load? Perhaps the file was accidentally deleted, or you mistyped the file name. Luckily, the web browser has a way to display some text when the image fails to load. This is done using the alt attribute. For example, you can display the text My Profile Photo using the alt attribute in the previous HTML. **<img src="photo.png" width="320" alt="My Profile Photo">** It is important to ensure that screen reader accessibility software can interpret images displayed in the web browser. To support this, the <img> tag is combined with the **<figure>** and **<figcaption>** tags to provide a description of the image. The **<img>** tag is added inside the **<figure>** tag and the **<figcaption>** is specified after it.

<figure>

   <img src="photo.png" width="320" alt="My Profile Photo">

   <figcaption>A photo of myself on a beach in 2015</figcaption>

</figure>





One last thing to note is that like videos and audio, the web browser only supports specific file types. These file types are:

* .APNG – Animated Portable Network Graphics
* .AVIF – AV1 Image Format
* .GIF – Graphics Interchange Format
* .JPEG / .JPG – Joint Photographic Expert Group image format
* .PNG – Portable Network Graphics
* .SVG – Scalable Vector Graphics
* .WEBP – Web Picture Format

Images will be important as you build websites and ensuring they are accessible will provide a better user experience for all visitors.