In order for a form to be usable, each form control should be accompanied by an indication of what it is for or how it should be used. For example: ‘Surname’ or ‘Enter your surname’ next to a field into which you are meant to enter your surname.

A label element represents such an indication (a caption).

A label element can be associated with a specific form control either using its for attribute or by putting the form control inside the label element itself.

```
<label for="first">First name(s): <input name="first" id="f1" type="text"></label>
```

A form is an element that contains form controls, such as text fields, buttons, checkboxes, range controls, or color pickers.

Forms allow users to enter data which can then be sent to a web server for processing.

A form element has several (optional) attributes, including:

- action: URL to use for form submission
- method: HTTP method to use for form submission (get or post)
- novalidate: form is not validated during submission

```
<form action="https://www.cs.liv.ac.uk/COMP/Calendar.pl" method="post" enctype="text/plain">
  <input type="submit">
</form>
```

A form can be submitted and on submission the data entered via the form controls will be sent to the URL in action using the HTTP request method in method with encoding type enctype.

A form is an element that contains form controls, such as text fields, buttons, checkboxes, range controls, or color pickers.

HTML5 introduced a number of additional form controls and attributes for all form controls, but browser support is partial. Always test your forms with a range of browsers (Apple Safari, Google Chrome, Mozilla Firefox, MS IE, MS Edge).

Depending on the value of the type attribute, an input element will have additional attributes that define its behaviour.

```
<input name="studentid" id="sid" type="number" min="190000000" max="999999999">
```

The input element represents a ‘field’ that allows the user to enter data of a certain type.

The type attribute of an input element determine what type of data can be entered and in what form.

```
<table>
<thead>
<tr>
<th>Value of type</th>
<th>Data type</th>
<th>Form control</th>
</tr>
</thead>
<tbody>
<tr>
<td>text</td>
<td>Text with no line breaks</td>
<td>Text field</td>
</tr>
<tr>
<td>tel</td>
<td>Phone number</td>
<td>Text field</td>
</tr>
<tr>
<td>date</td>
<td>Date (Year/Month/Day)</td>
<td>Date control</td>
</tr>
<tr>
<td>email</td>
<td>E-mail address</td>
<td>Text field</td>
</tr>
<tr>
<td>file</td>
<td>Zero or more files</td>
<td>Button and file selector</td>
</tr>
<tr>
<td>color</td>
<td>sRGB color</td>
<td>Button and color picker</td>
</tr>
<tr>
<td>number</td>
<td>Floating-point number</td>
<td>Text field or spinner</td>
</tr>
<tr>
<td>password</td>
<td>Password</td>
<td>Text field with obscured input</td>
</tr>
</tbody>
</table>
```

The input element represents a ‘field’ that allows the user to enter data of a certain type.

The type attribute of an input element determine what type of data can be entered and in what form.

```
<table>
<thead>
<tr>
<th>Value of type</th>
<th>Data type</th>
<th>Form control</th>
</tr>
</thead>
<tbody>
<tr>
<td>checkbox</td>
<td>A set of zero or more values from a predefined list</td>
<td>Checkbox</td>
</tr>
<tr>
<td>radio</td>
<td>An enumerated value</td>
<td>Radio button</td>
</tr>
<tr>
<td>button</td>
<td>Initiates form submission</td>
<td>Button</td>
</tr>
<tr>
<td>reset</td>
<td>Resets form</td>
<td>Button</td>
</tr>
</tbody>
</table>
```

Depending on the value of the type attribute, an input element will have additional attributes that define its behaviour.

Common attributes include:

- id: unique id used to identify the element with the document
- name: (unique) name used by the form processor to access input
- autofocus: automatically focus on this form control when the page is loaded
- disabled: whether the form control is disabled
- required: whether the form control is required to have a non-empty value for form submission
Names versus IDs

- On submission, the web browser will construct pairs `name=value` where `name` is the value of the name attribute of one of the form controls and `value` is the input by the user for that form control
- A string composed of those pairs will be sent to process.php

Example:

Peters, Amy Lee, and 201612345 are entered into the three fields whom are sent to process.php

```
<form action="process.php" method="post">
    <label>Name: <input name="fn" required /></label>
    <label>Eassy: <textarea name="essay" required></textarea></label>
    <input type="submit" name="action" value="Save essay" formnovalidate />
</form>
```

Input: Submit

- An input element with type submit represents a button that, when activated, submits the form it is associated with
- Attributes include:
  - name: replaces the default label of the button
  - value: (unique) name used by the form processor to access input
  - formaction: Overwrite the
  - formmethod: attributes of the form
  - formnovalidate: attributes of the form

```
<form action="process.php" method="post">
    <label>Name: <input name="fn" required /></label>
    <label>Eassy: <textarea name="essay" required></textarea></label>
    <input type="submit" name="action" value="Save essay" formnovalidate />
</form>
```

Input: Submit (Example 1)

```
<form action="process.php" method="post">
    <label>Name: <input name="fn" required /></label>
    <label>Eassy: <textarea name="essay" required></textarea></label>
    <input type="submit" name="action" value="Save essay" formnovalidate />
</form>
```

Input: Submit (Example 2)

```
<form action="process.php" method="post">
    <label>Name: <input name="fn" required /></label>
    <label>Eassy: <textarea name="essay" required></textarea></label>
    <input type="submit" name="action" value="Save essay" formnovalidate />
</form>
```
Input: Number

- An input element with type number represents a one line plain text field for the element’s value
- Attributes include:
  - value: shown as ‘default’ value in the text field
  - readonly: the value cannot be changed
  - min: minimum value allowed to be entered
  - max: maximum value allowed to be entered
  - step: granularity that is expected (and required) of the value
  - placeholder: a short hint describing the expected value of a text area

```html
<form action="process.php">
  <input type="number" name="quantity" min="0" max="14" step="2" value="2">
</form>
```

Input: Password

- An input element with type password represents a one line plain text field for the element’s value
- Same as an input element with type text, except that the web browser should obscure the value that is being entered
- Has the same attributes as input element with type text

```html
<form action="process.php">
  <input type="password" name="password" size="16">
</form>
```

Input: Radio

- A set of input elements with type radio represents a radio button group in which only one form control can be selected/set to true
- For input elements to belong to the same radio button group:
  - they must be associated with the same form
  - their name-attributes must have the same value
- Attributes include:
  - value: replaces the ‘default’ return value on
  - checked: this element is selected by default

```html
<form action="process.php">
  <p>Please select your preferred contact method:</p>
  <input type="radio" id="cc1" name="contact" value="email">
  <label for="cc1">Email</label>
  <input type="radio" id="cc2" name="contact" value="phone">
  <label for="cc2">Phone</label>
  <input type="radio" id="cc3" name="contact" value="mail">
  <label for="cc3">Mail</label>
  <input type="submit">
</form>
```

Input: Text

- An input element with type text represents a one line plain text field for the element’s value
- Attributes include:
  - value: shown as ‘default’ value in the text field
  - readonly: the value cannot be changed
  - size: visible width of the field in characters
  - minlength: minimum number of characters allowed to be entered
  - maxlength: maximum number of characters allowed to be entered
  - placeholder: a short hint describing the expected value of a text area
  - pattern: a regular expression that the value must match

```html
<form action="process.php">
  <input type="text" name="name" size="100" required>
  <input type="text" name="department" value="Computer Science" readonly>
  <input type="text" name="studentid" minlength="9" maxlength="9">
</form>
```

Input: Checkbox

- An input element with type checkbox represents a two-state control (checkbox)
- The return value of a checked checkbox is on, an unchecked checkbox returns nothing
- Attributes include:
  - value: replaces the ‘default’ return value on
  - checked: this element is selected by default

```html
<form action="process.php">
  <input type="checkbox" name="fruit" value="o">
  <input type="checkbox" name="fruit" value="a">
</form>
```

Textarea

- A textarea element represents a multi-line text input control
- A textarea element can have several attributes, including:
  - id: unique id used to identify the element with the document
  - name: (unique) name used by the form processor to access input
  - cols: the visible width of a text area (number)
  - rows: the visible height of a text area (number)
  - maxlength: maximum number of characters that can be entered
  - placeholder: a short hint describing the expected value of a text area

```html
<textarea name="problem" id="problem" cols="40" rows="10">
What fruit do you like?
</textarea>
```

- wrap: specifies whether newlines are preserved ("hard" is YES, "soft" is NO)
- required: specifies that a text area must be filled out

```html
<form action="process.php">
  <textarea name="problem" id="problem" cols="40" rows="10" wrap="soft" required wrap="hard">
  Describe your problem
</textarea>
</form>
```
**Select**

- A *select* element represents a drop-down menu with pre-defined options between which the user must select.

- The content of a *select* element consists of a list of *option* elements that represent those options.

- A *select* element can have several attributes, including:
  - id: unique id used to identify the element with the document
  - name: (unique) name used by the form processor to access input
  - multiple: allow multiple options to be selected
  - required: an option must be selected that has a non-empty value
  - disabled: the current selection can not be changed
  - size: number of options to show to the user

- An *option* element can have several attributes, including:
  - label: the label used in the drop-down menu
  - value: the value returned for the option
  - selected: the option is selected by default
  - disabled: the option is shown but cannot be selected

```html
<form action='process.php'>
  <label for='module'>Select a module: </label>
  <select name='module'>
    <option value='COMP517'>COMP517: Object Oriented Software Development</option>
    <option value='COMP518'>COMP518: Database and Information Systems</option>
    <option value='COMP519'>COMP519: Web Programming</option>
  </select>
  <input type='submit'>
</form>
```

By default, the first option is selected.

If the selection is not changed and the user activates the submit button, then module=COMP517 is sent to process.php.

In general, the value associated with the selected option will be send.

- Adding the attribute `selected` to the second option, makes it the option that is selected by default.

```html
<form action='process.php'>
  <label for='module'>Select a module: </label>
  <select name='module'>
    <option value='COMP517'>COMP517: Object Oriented Software Development</option>
    <option value='COMP518' selected>COMP518: Database and Information Systems</option>
    <option value='COMP519'>COMP519: Web Programming</option>
  </select>
  <input type='submit'>
</form>
```

- That an option with a non-empty value is pre-selected is often not desirable — the user does not need to make a conscious choice.

- Adding a default option with empty value and adding the attribute required to the select element forces the user to make a conscious choice.

```html
<form action='process.php'>
  <label for='module'>Your choice: </label>
  <select name='module' required>
    <option value=''>Select a module</option>
    <option value='COMP517'>COMP517: Object Oriented Software Development</option>
    <option value='COMP518'>COMP518: Database and Information Systems</option>
    <option value='COMP519'>COMP519: Web Programming</option>
  </select>
  <input type='submit'>
</form>
```

- Revision and Further Reading

Read:
- Chapter 9: Forms
- Chapter 19: More CSS Techniques (Styling Forms)

E-book [https://library.liv.ac.uk/record=b5647021](https://library.liv.ac.uk/record=b5647021)