4COSCO11C.2 Web Design and Development

HTML Forms

Week 4

UNIVERSITY OF WESTMINSTER

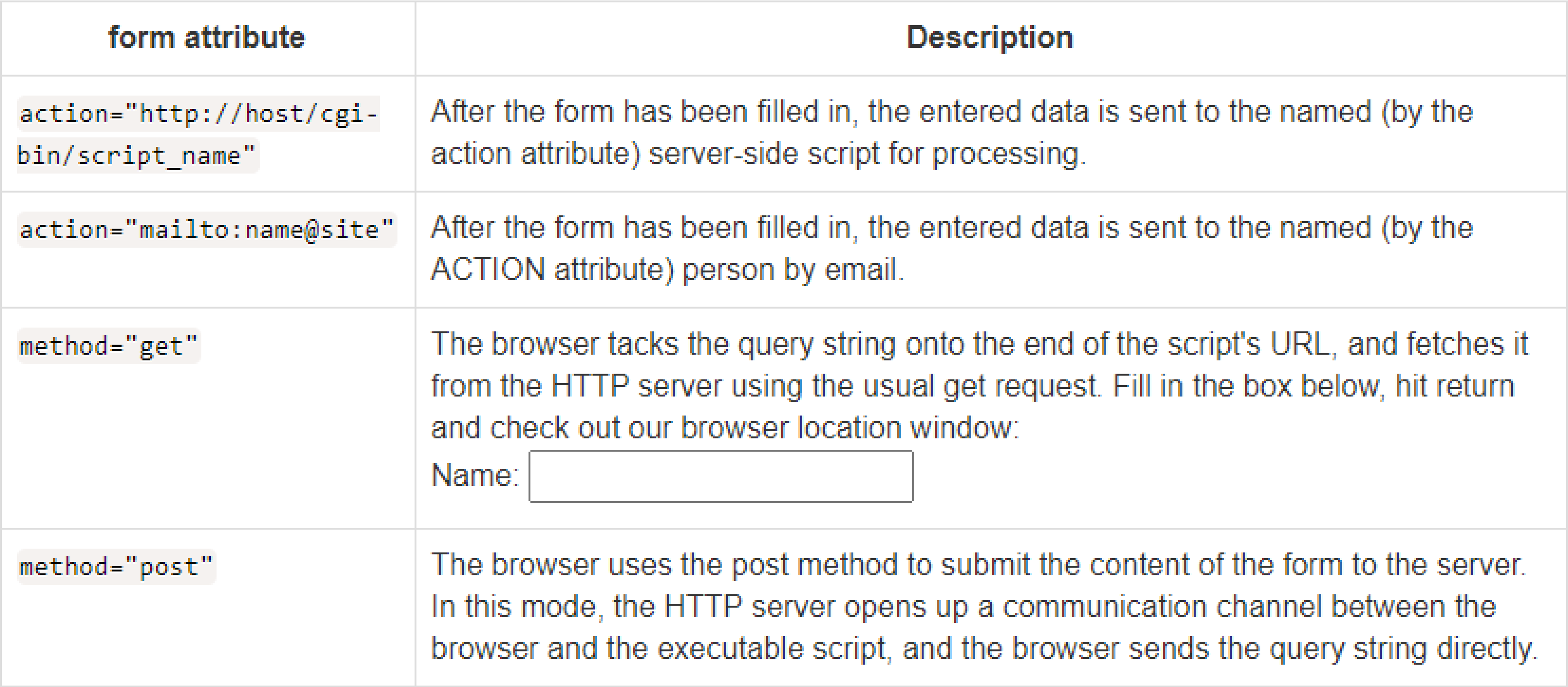
INFORMATICS INSTITUTE OF TECHNOLOGY

Form filling

- It is possible to create forms as Web Pages, to request a user to enter some information.
- When he/she has finished filling in the form, the entered data can be sent to a server-side script or to someone by email for processing.
- The scripts resides and runs on the machine running the web server from which web pages are delivered.
- A form is introduced by the tag <form> and terminated by the inverse tag </form>.

HTML <form>

- The attributes of the <form> tag include:

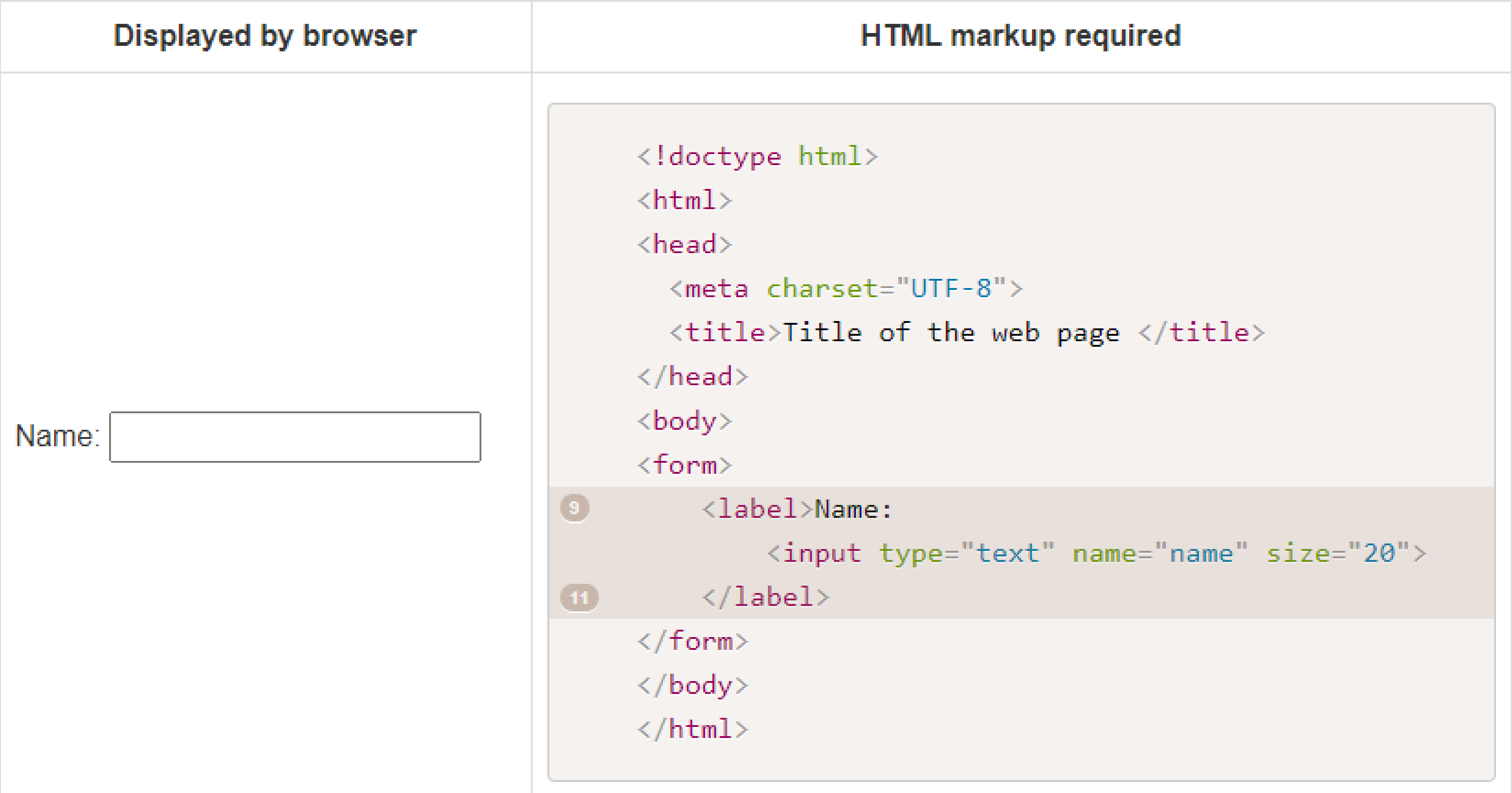


form attribute	Description
`action="http://host/cgi-bin/script_name"`	After the form has been filled in, the entered data is sent to the named (by the action attribute) server-side script for processing.
`action="mailto:name@site"`	After the form has been filled in, the entered data is sent to the named (by the ACTION attribute) person by email.
`method="get"`	The browser tacks the query string onto the end of the script's URL, and fetches it from the HTTP server using the usual get request. Fill in the box below, hit return and check out our browser location window: Name:
`method="post"`	The browser uses the post method to submit the content of the form to the server. In this mode, the HTTP server opens up a communication channel between the browser and the executable script, and the browser sends the query string directly.

- **Review Question** - which is the default method?
- **Please note**: server-side scripts will be covered next year.

Text Input

- A form to request the user to enter some text is shown below.



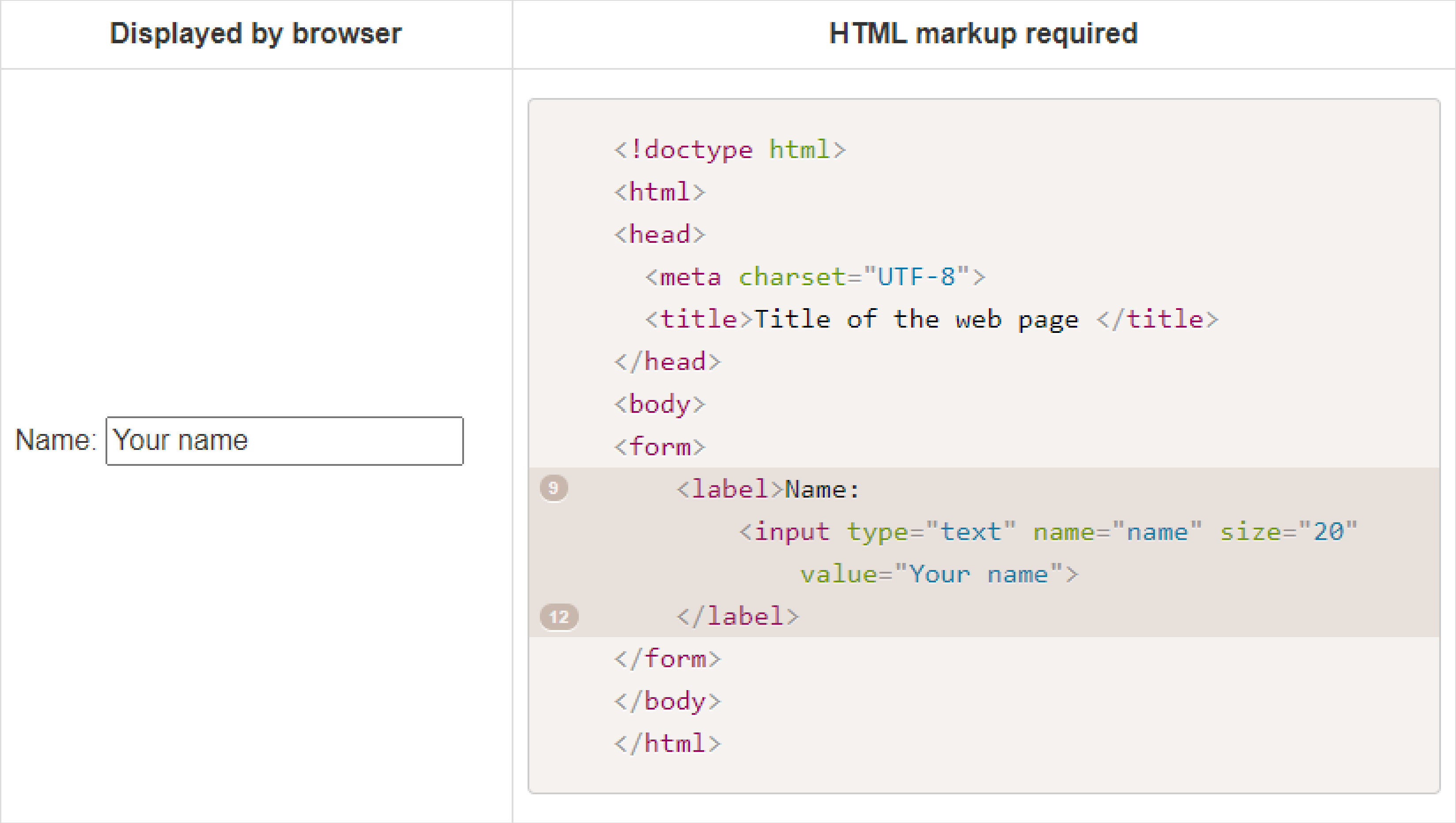
Displayed by browser	HTML markup required

Name:

```
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Title of the web page </title>
</head>
<body>
<form>
    <label>Name:
        <input type="text" name="name" size="20">
    </label>
</form>
</body>
</html>
```

Contd

- The data in the input field can be initialized to a particular value using the value attribute:



Displayed by browser

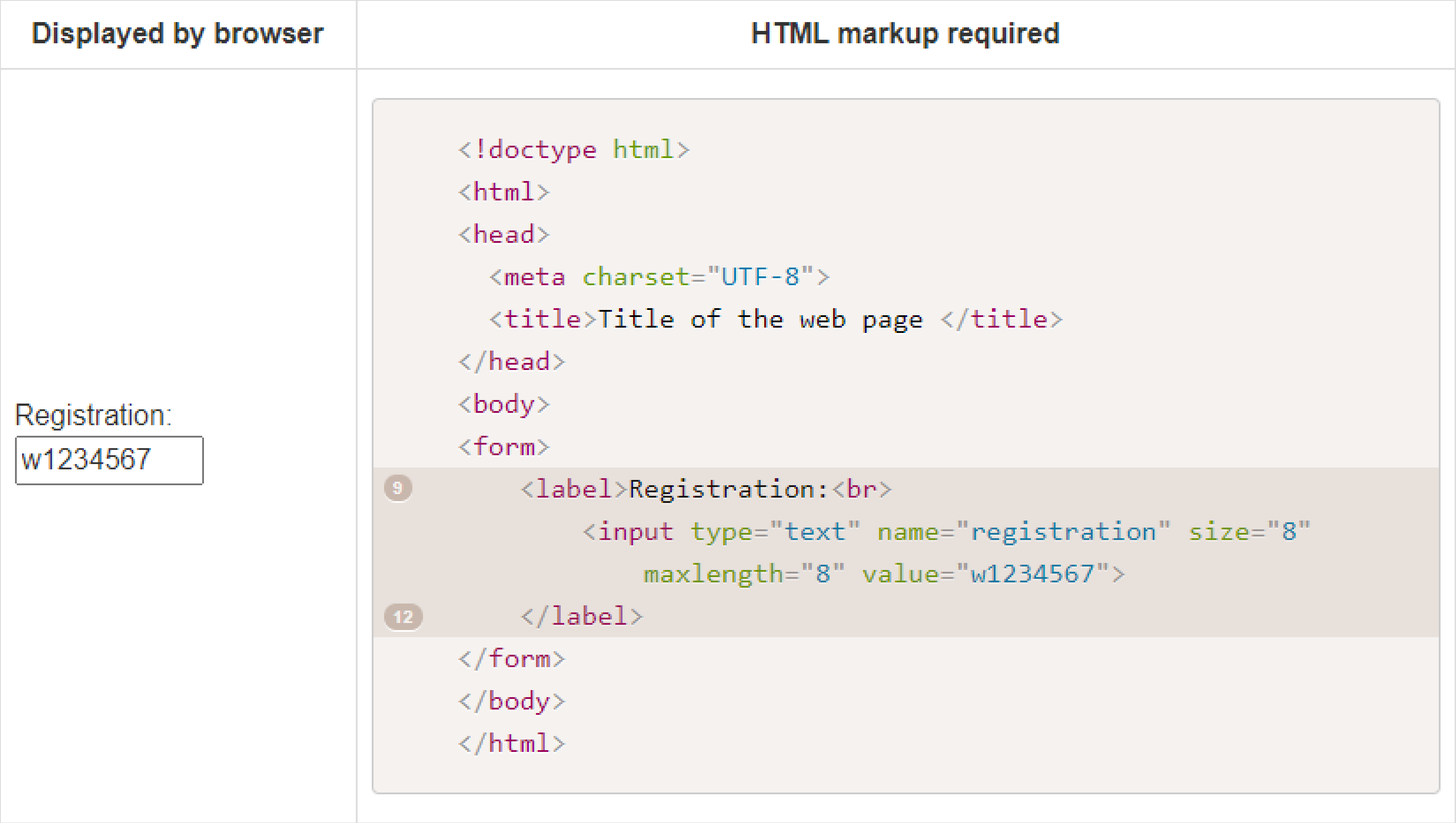
Name: Your name

HTML markup required

```
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Title of the web page </title>
</head>
<body>
<form>
    <label>Name:
        <input type="text" name="name" size="20"
            value="Your name">
    </label>
</form>
</body>
</html>
```

Contd

- It is also possible to specify the **maximum length** for a text to be entered:

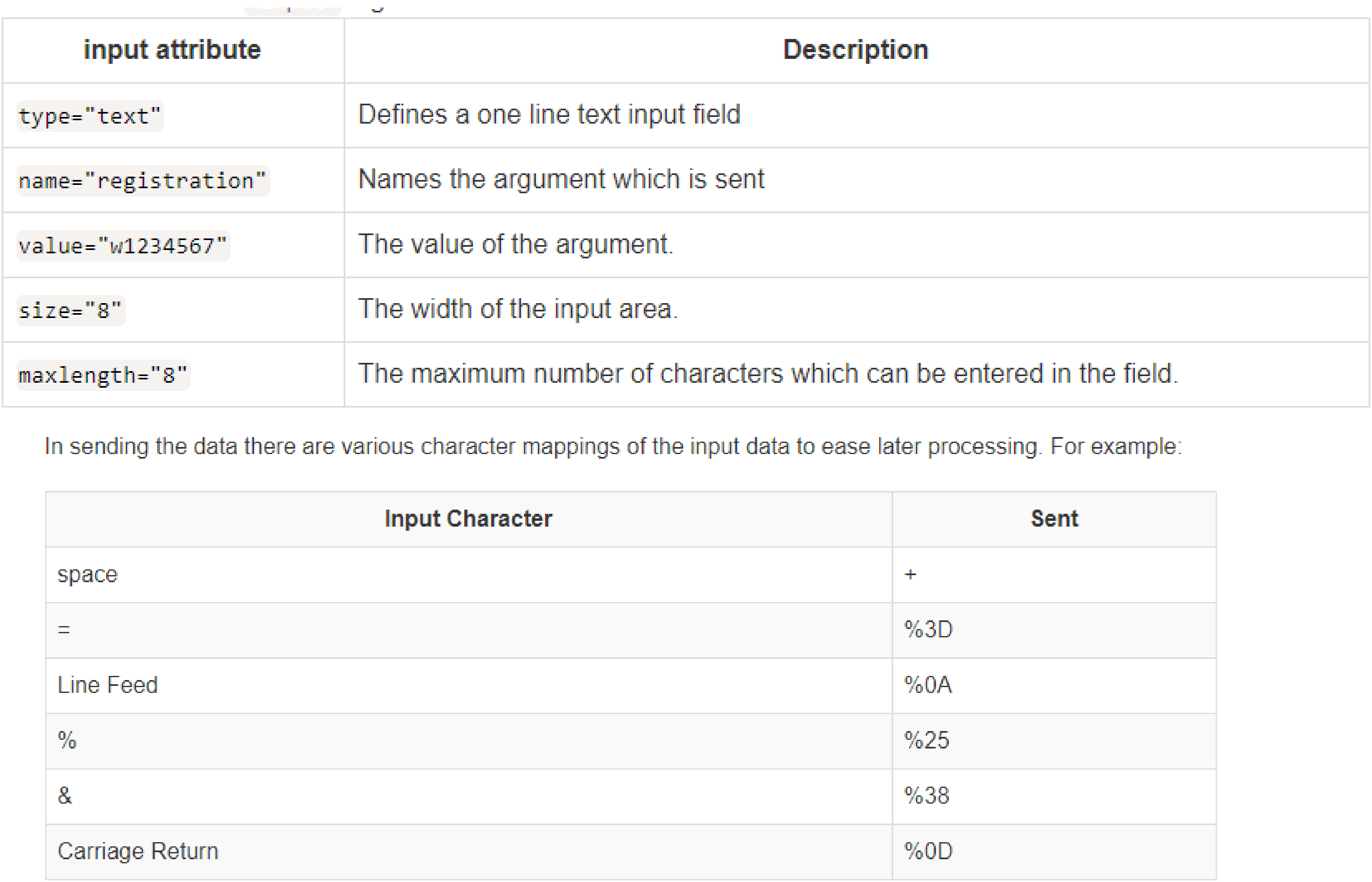


Displayed by browser | **HTML markup required**

Registration:
w1234567

```
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Title of the web page </title>
</head>
<body>
<form>
    <label>Registration:<br>
        <input type="text" name="registration" size="8"
            maxlength="8" value="w1234567">
    </label>
</form>
</body>
</html>
```

<Input> tag attributes



input attribute	Description
`type="text"`	Defines a one line text input field
`name="registration"`	Names the argument which is sent
`value="w1234567"`	The value of the argument.
`size="8"`	The width of the input area.
`maxlength="8"`	The maximum number of characters which can be entered in the field.

In sending the data there are various character mappings of the input data to ease later processing. For example:

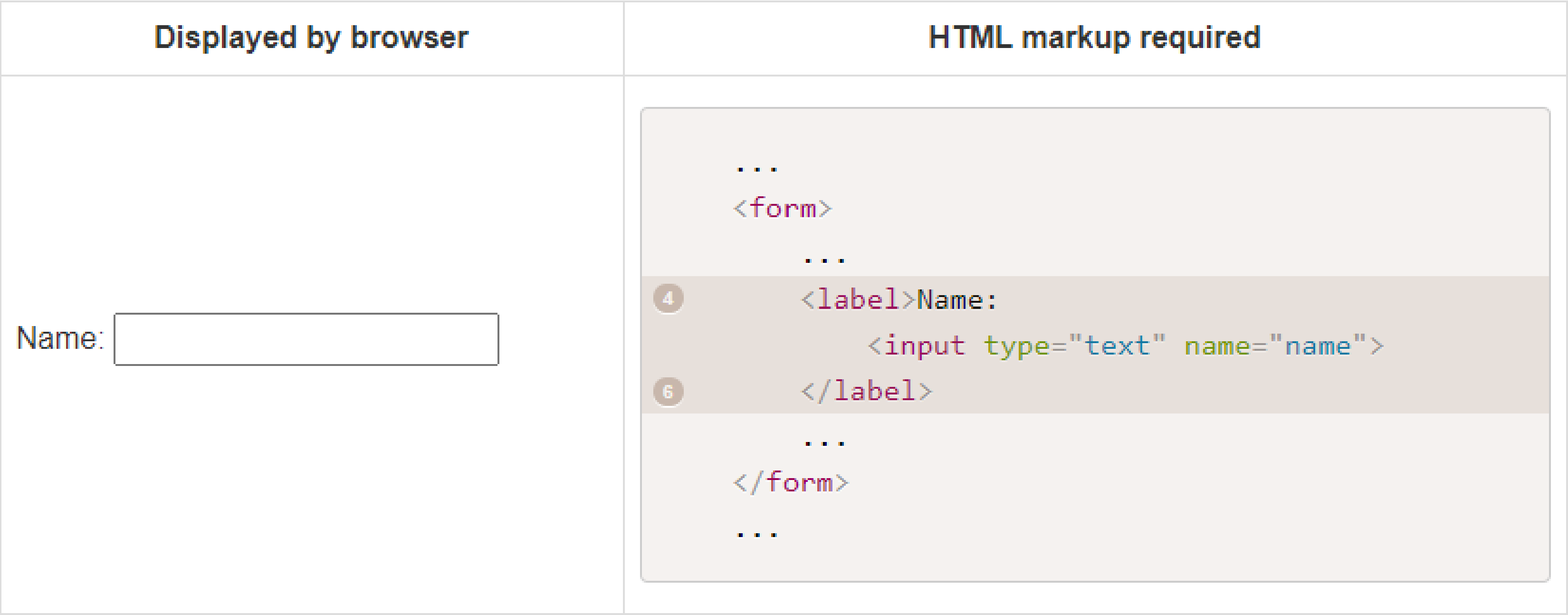
Input Character	Sent
space	+
=	%3D
Line Feed	%0A
%	%25
&	%38
Carriage Return	%0D

Labels

- The <label> element represents a caption for the form element it is associated with. The <label> can also add as an accessibility aid to forms:
    - they can widen the element clickable area
    - they help screen readers identify fields on a web page

Implicit Labels

- Implicit labels are wrapped around a form control. Implicit labels are not well suported in older screen readers, and are harder to style with CSS

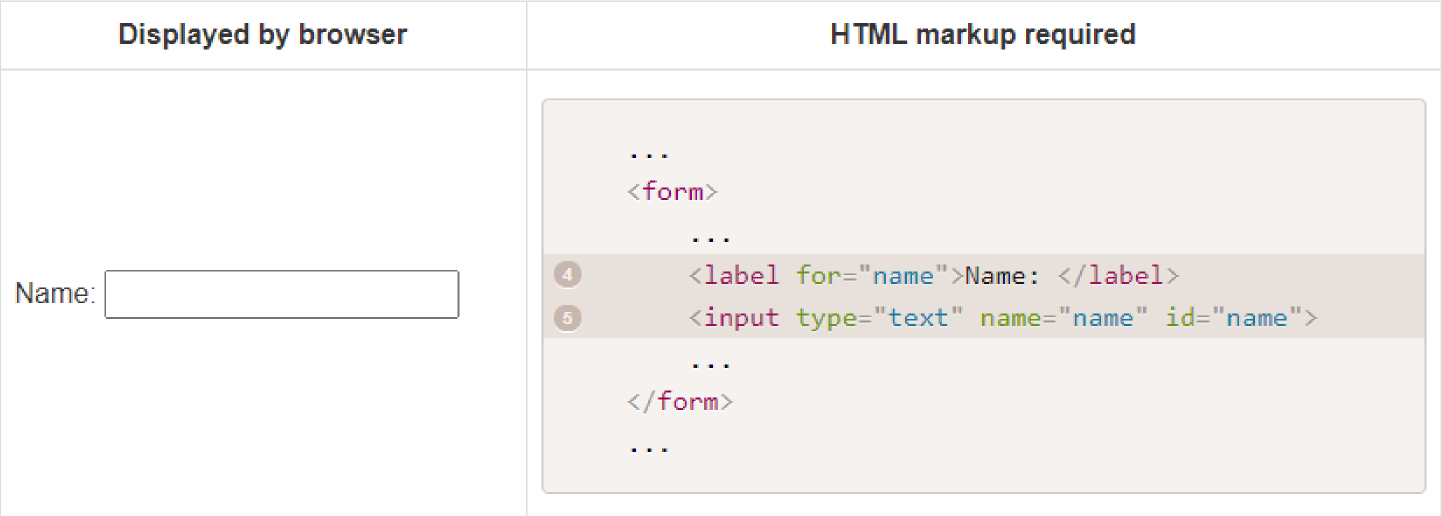


Displayed by browser	HTML markup required
Name:	see code below

```
...
<form>
    ...
    <label>Name:
        <input type="text" name="name">
    </label>
    ...
</form>
...
```

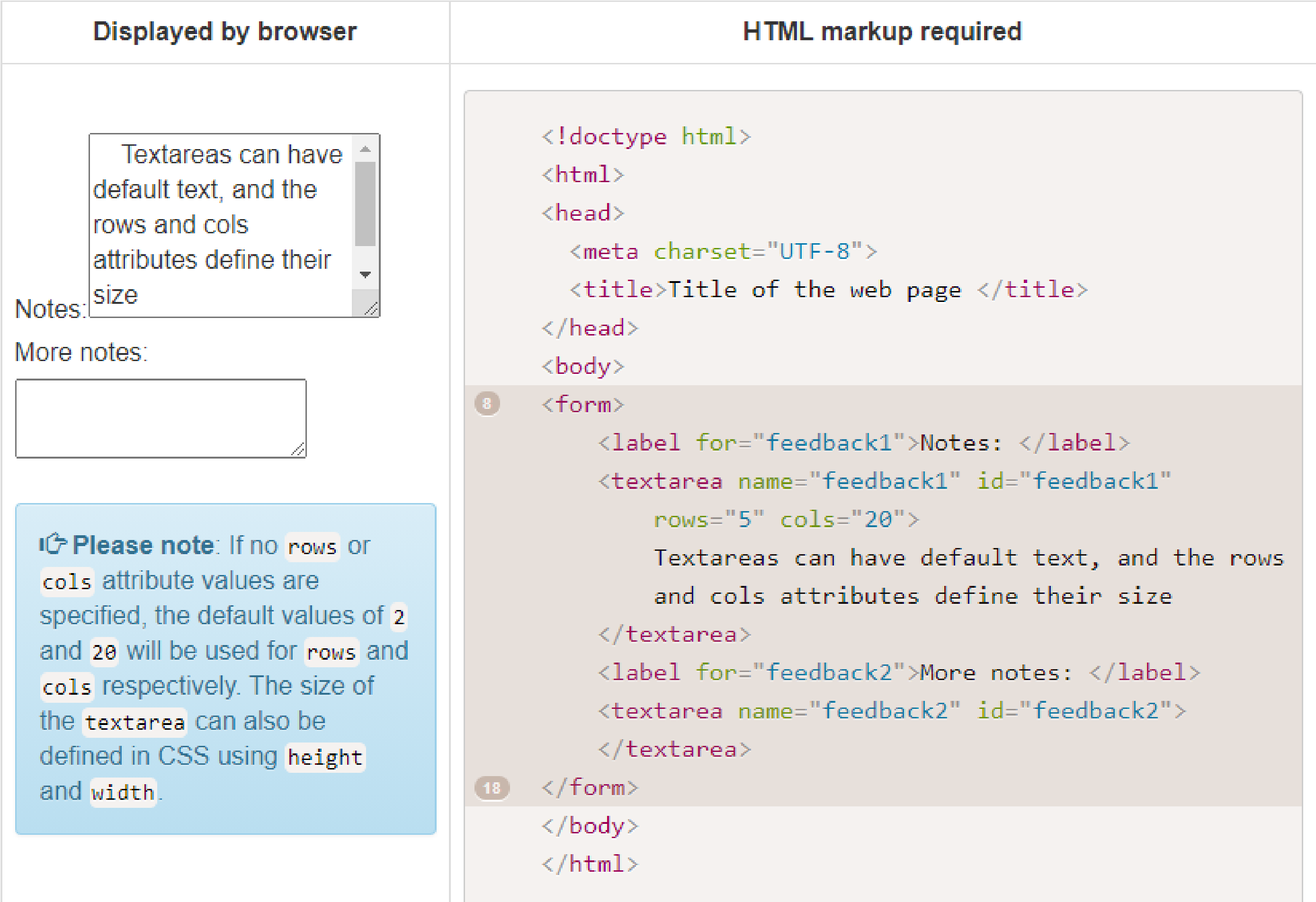
Explicit Labels

- Explicit labels are separate from the form control, and linked to their form control via the for attribute.
- The for attribute of the label has to match the id attribute of the form control it is linked to.
- Explicit labels are well supported by screen readers and are easy to style using CSS. We will be using explicit labels in the notes.



Displayed by browser	HTML markup required
Name:	... <form> ... <label for="name">Name: </label> <input type="text" name="name" id="name"> ... </form> ...

Multiple lines of text input - Textarea



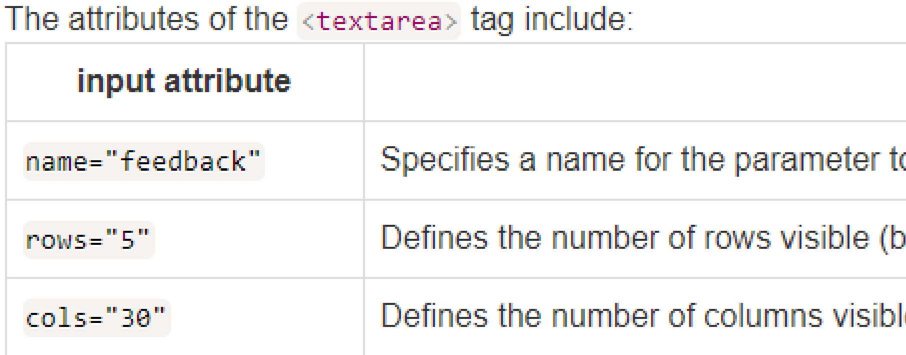
Displayed by browser	HTML markup required

Notes: Textareas can have default text, and the rows and cols attributes define their size

More notes:

☞ **Please note**: If no `rows` or `cols` attribute values are specified, the default values of `2` and `20` will be used for `rows` and `cols` respectively. The size of the `textarea` can also be defined in CSS using `height` and `width`.

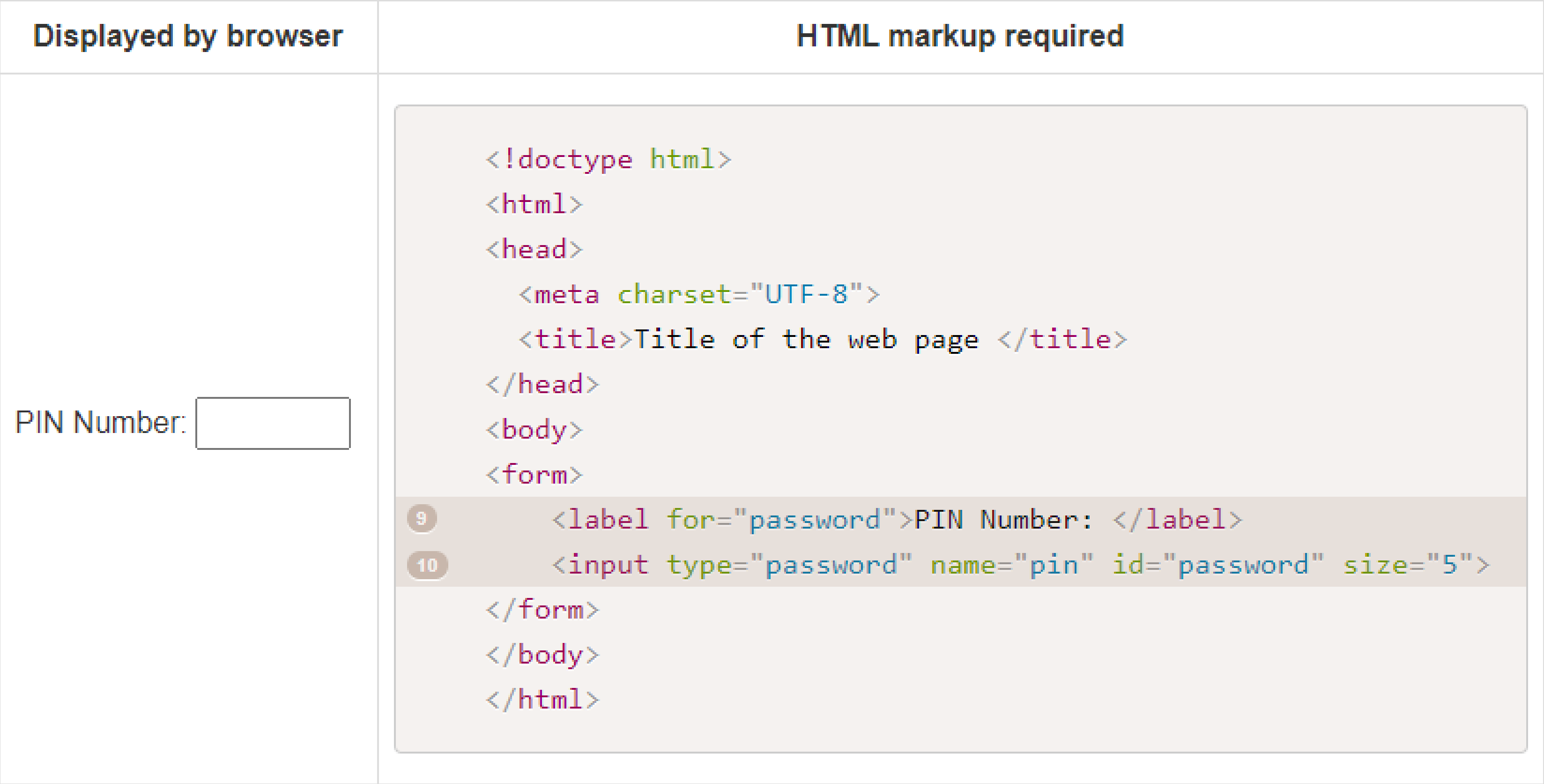
```
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Title of the web page </title>
</head>
<body>
<form>
    <label for="feedback1">Notes: </label>
    <textarea name="feedback1" id="feedback1"
        rows="5" cols="20">
        Textareas can have default text, and the rows
        and cols attributes define their size
    </textarea>
    <label for="feedback2">More notes: </label>
    <textarea name="feedback2" id="feedback2">
    </textarea>
</form>
</body>
</html>
```



The attributes of the `<textarea>` tag include:

input attribute	
`name="feedback"`	Specifies a name for the parameter to
`rows="5"`	Defines the number of rows visible (b
`cols="30"`	Defines the number of columns visibl

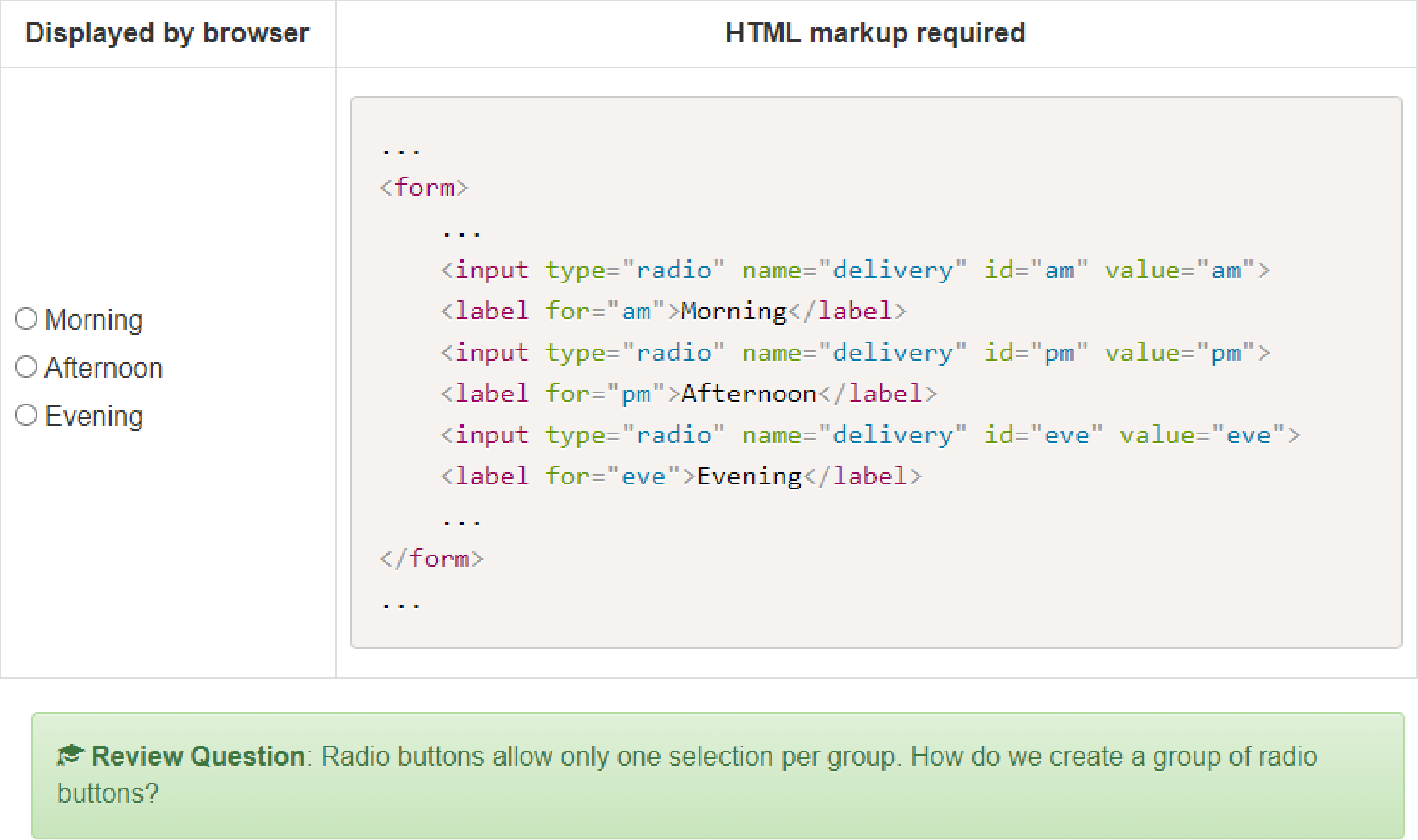
Password



Displayed by browser	HTML markup required

PIN Number:	`<!doctype html>`
	`<html>`
	`<head>`
	`  <meta charset="UTF-8">`
	`  <title>Title of the web page </title>`
	`</head>`
	`<body>`
	`<form>`
	`    <label for="password">PIN Number: </label>`
	`    <input type="password" name="pin" id="password" size="5">`
	`</form>`
	`</body>`
	`</html>`

Radio Buttons

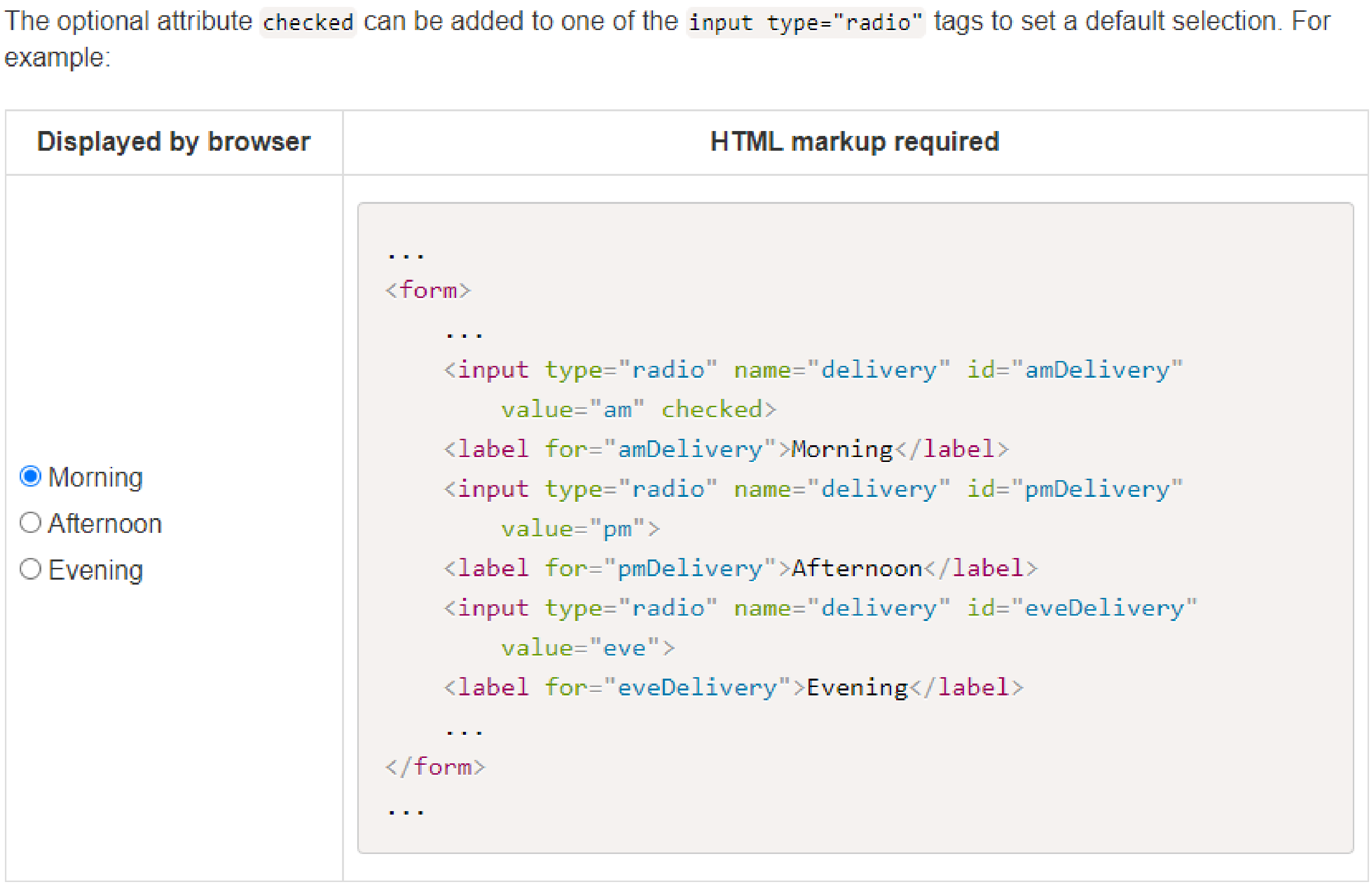


Displayed by browser	HTML markup required

○ Morning
○ Afternoon
○ Evening
	`...`
	`<form>`
	`    ...`
	`    <input type="radio" name="delivery" id="am" value="am">`
	`    <label for="am">Morning</label>`
	`    <input type="radio" name="delivery" id="pm" value="pm">`
	`    <label for="pm">Afternoon</label>`
	`    <input type="radio" name="delivery" id="eve" value="eve">`
	`    <label for="eve">Evening</label>`
	`    ...`
	`</form>`
	`...`

🎓 **Review Question**: Radio buttons allow only one selection per group. How do we create a group of radio buttons?

Contd

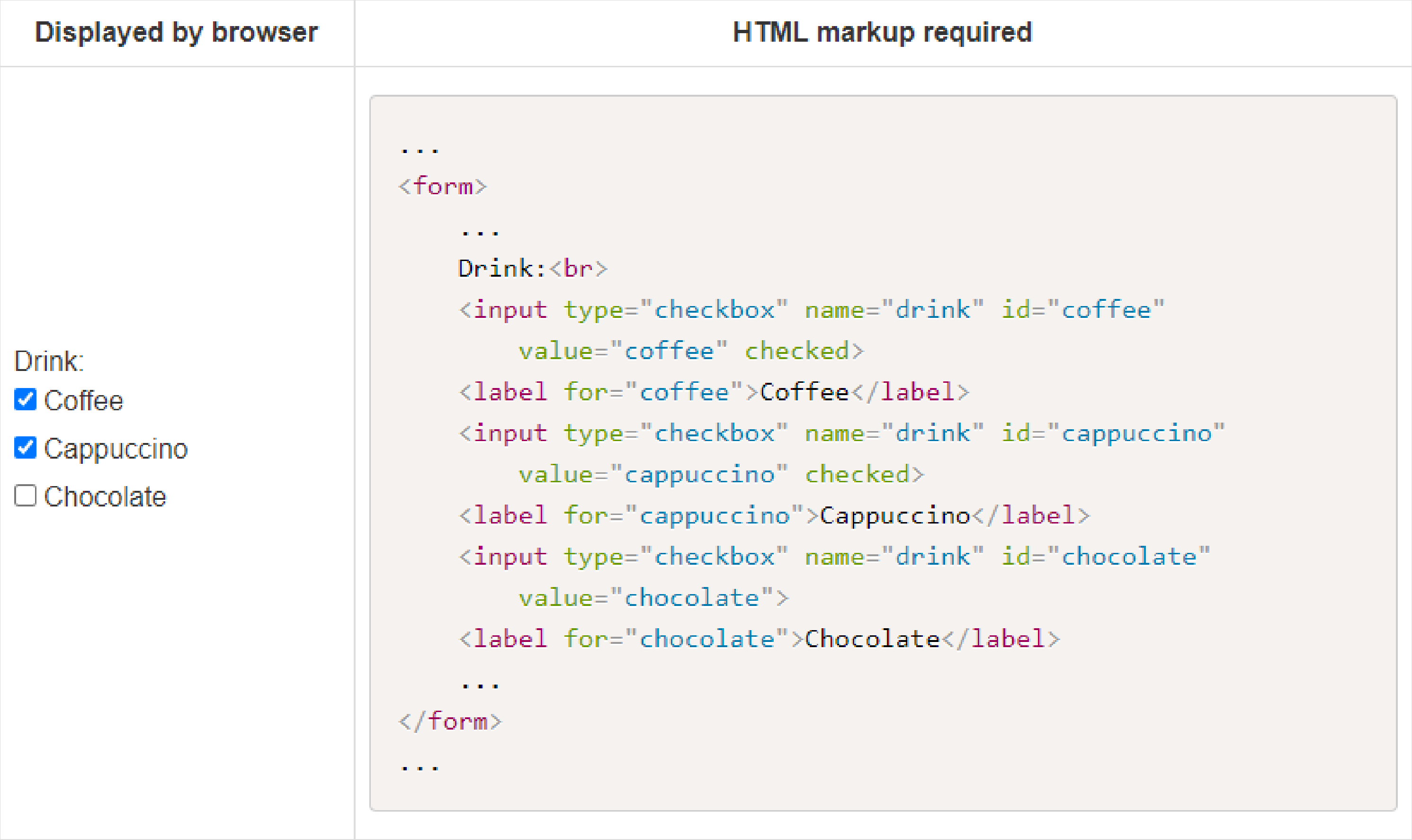


The optional attribute `checked` can be added to one of the `input type="radio"` tags to set a default selection. For example:

Displayed by browser	HTML markup required

● Morning
○ Afternoon
○ Evening
	`...`
	`<form>`
	`    ...`
	`    <input type="radio" name="delivery" id="amDelivery"`
	`        value="am" checked>`
	`    <label for="amDelivery">Morning</label>`
	`    <input type="radio" name="delivery" id="pmDelivery"`
	`        value="pm">`
	`    <label for="pmDelivery">Afternoon</label>`
	`    <input type="radio" name="delivery" id="eveDelivery"`
	`        value="eve">`
	`    <label for="eveDelivery">Evening</label>`
	`    ...`
	`</form>`
	`...`

Checkboxes

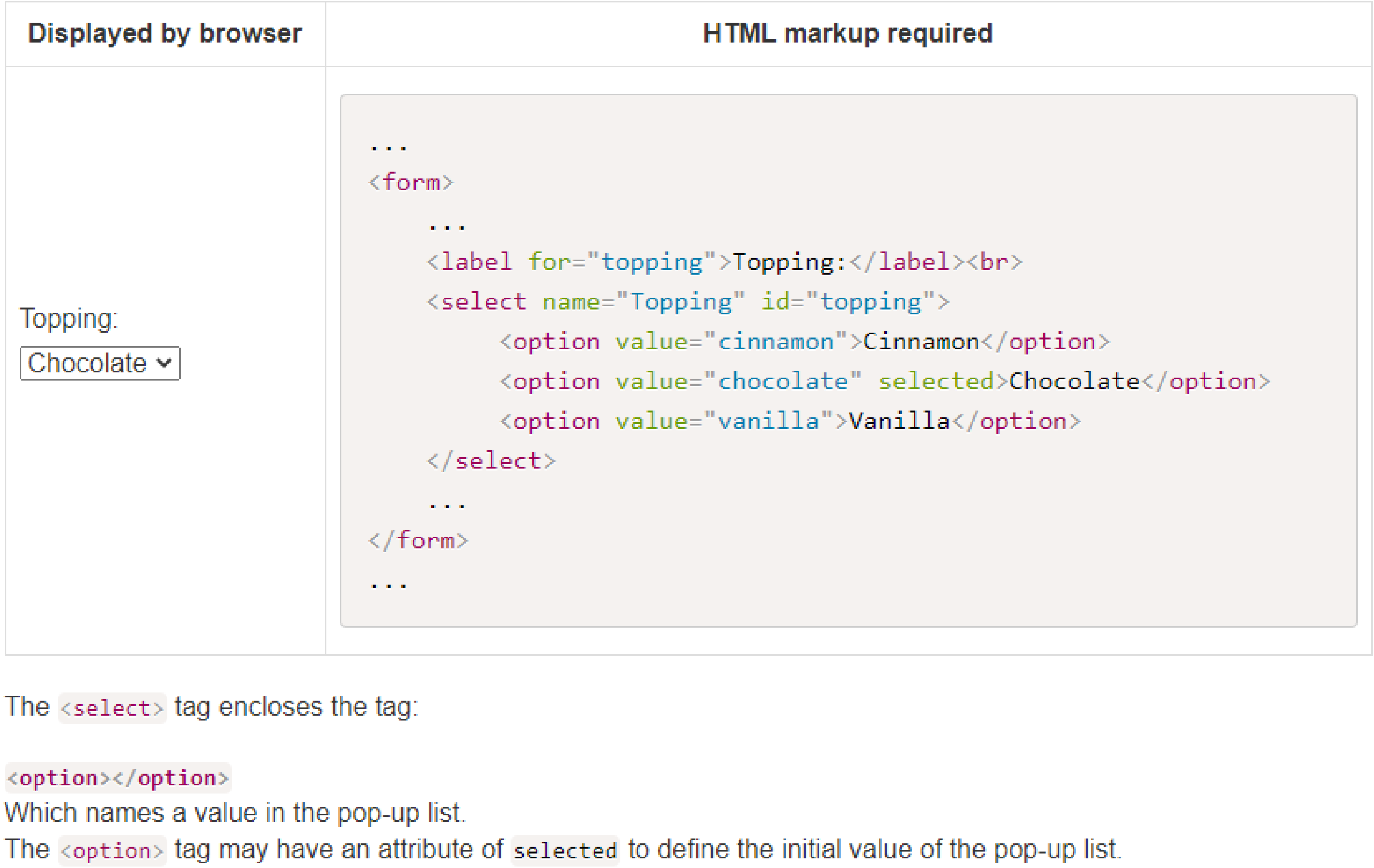


Displayed by browser	HTML markup required
Drink: ☑ Coffee ☑ Cappuccino ☐ Chocolate	

```
...
<form>
    ...
    Drink:<br>
    <input type="checkbox" name="drink" id="coffee"
        value="coffee" checked>
    <label for="coffee">Coffee</label>
    <input type="checkbox" name="drink" id="cappuccino"
        value="cappuccino" checked>
    <label for="cappuccino">Cappuccino</label>
    <input type="checkbox" name="drink" id="chocolate"
        value="chocolate">
    <label for="chocolate">Chocolate</label>
    ...
</form>
...
```

Pop up List

A form to allow the user to select **an item** from a pop-up list uses the <select> tag.
An example of a pop-up list is shown below:



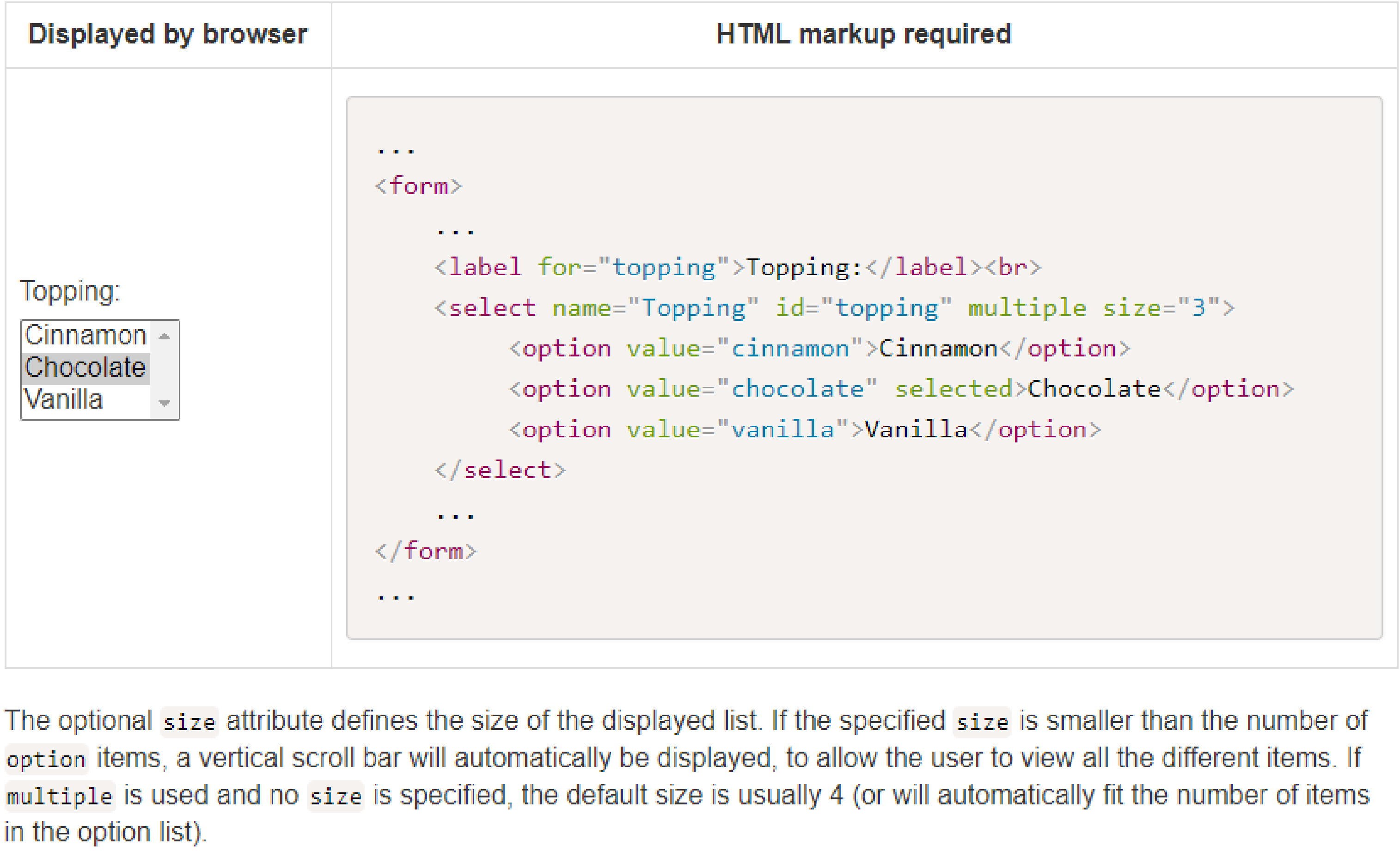
<table>
<tr><th>Displayed by browser</th><th>HTML markup required</th></tr>
<tr><td>Topping:
Chocolate ⌄</td><td><pre>
...
<form>
    ...
    <label for="topping">Topping:</label>
    <select name="Topping" id="topping">
         <option value="cinnamon">Cinnamon</option>
         <option value="chocolate" selected>Chocolate</option>
         <option value="vanilla">Vanilla</option>
    </select>
    ...
</form>
...
</pre></td></tr>
</table>

The `<select>` tag encloses the tag:

`<option></option>`
Which names a value in the pop-up list.
The `<option>` tag may have an attribute of `selected` to define the initial value of the pop-up list.

Scrolling List

- A form to allow the user to **select one or more item** from a list uses the <select> tag with the multiple attribute.



Displayed by browser

Topping:

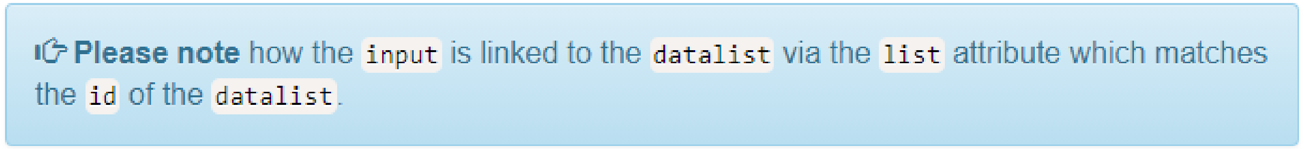
Cinnamon
Chocolate
Vanilla

HTML markup required

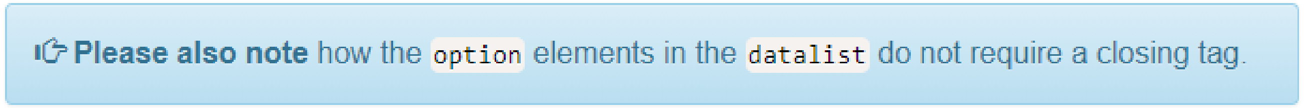
```
...
<form>
    ...
    <label for="topping">Topping:</label><br>
    <select name="Topping" id="topping" multiple size="3">
        <option value="cinnamon">Cinnamon</option>
        <option value="chocolate" selected>Chocolate</option>
        <option value="vanilla">Vanilla</option>
    </select>
    ...
</form>
...
```

The optional `size` attribute defines the size of the displayed list. If the specified `size` is smaller than the number of `option` items, a vertical scroll bar will automatically be displayed, to allow the user to view all the different items. If `multiple` is used and no `size` is specified, the default size is usually 4 (or will automatically fit the number of items in the option list).

Data List

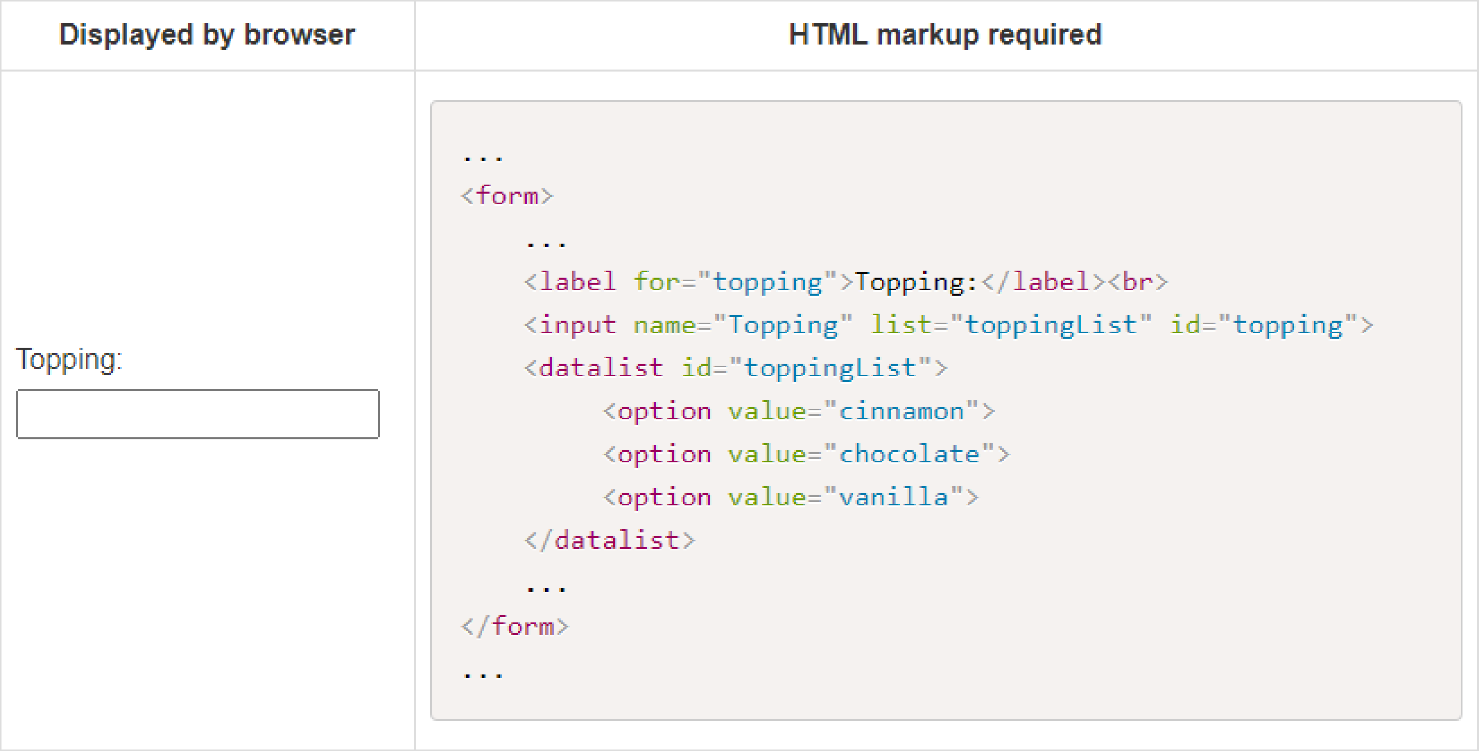


☞ **Please note** how the `input` is linked to the `datalist` via the `list` attribute which matches the `id` of the `datalist`.



☞ **Please also note** how the `option` elements in the `datalist` do not require a closing tag.

- datalist is an HTML5 element.
- The datalist represents a list of options available for its parent element.
- The datalist can be paired up with an <input type="text" list="idOfList“>

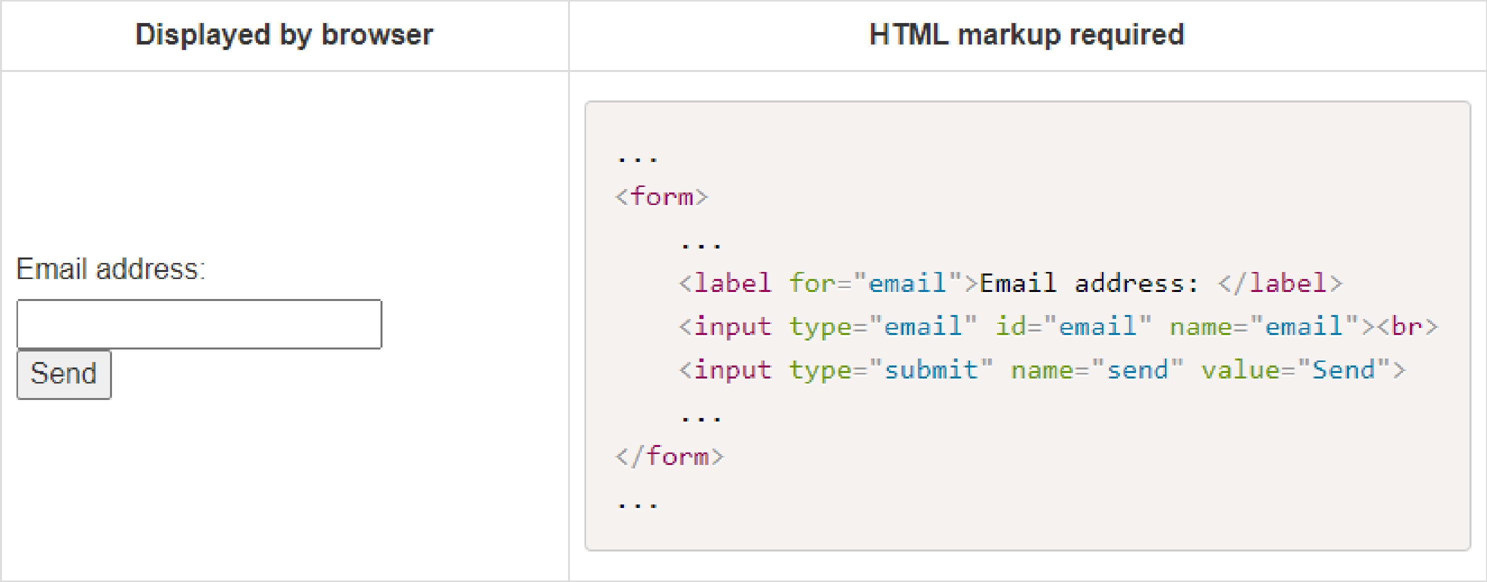


Displayed by browser	HTML markup required

Topping:
□
	...
	<form>
	    ...
	    <label for="topping">Topping:</label>
	    <input name="Topping" list="toppingList" id="topping">
	    <datalist id="toppingList">
	        <option value="cinnamon">
	        <option value="chocolate">
	        <option value="vanilla">
	    </datalist>
	    ...
	</form>
	...

Email input field

- email is an HTML5 input type.
- Browsers that support this input type will automatically check the validity of the format of the data entered and alert the user if the format is invalid.
- If the browser doesn't support this input type, it will automatically fall back to the type="text"

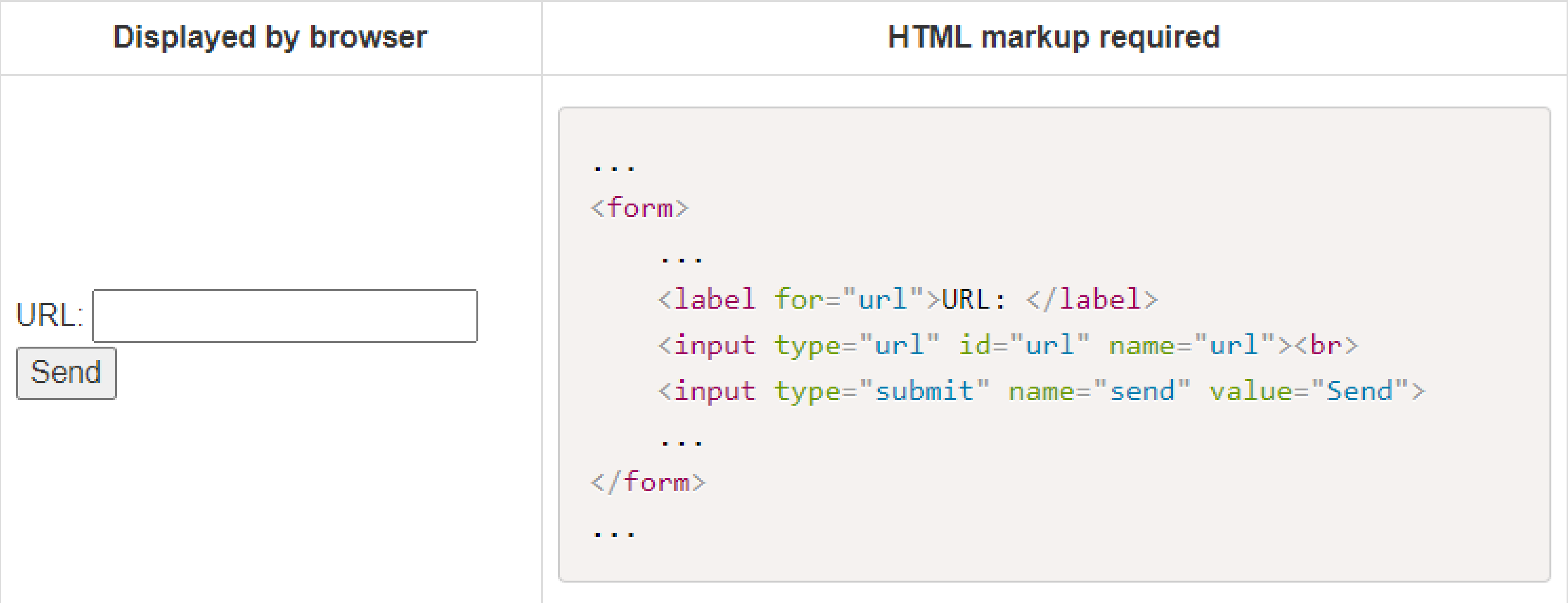


Displayed by browser	HTML markup required

```
...
<form>
    ...
    <label for="email">Email address: </label>
    <input type="email" id="email" name="email"><br>
    <input type="submit" name="send" value="Send">
    ...
</form>
...
```

URL input

- URL is an HTML5 input type.
- Browsers that support this input type will automatically check the validity of the format of the data entered and alert the user if the format is invalid.
- If the browser doesn't support this input type, it will automatically fall back to the type="text

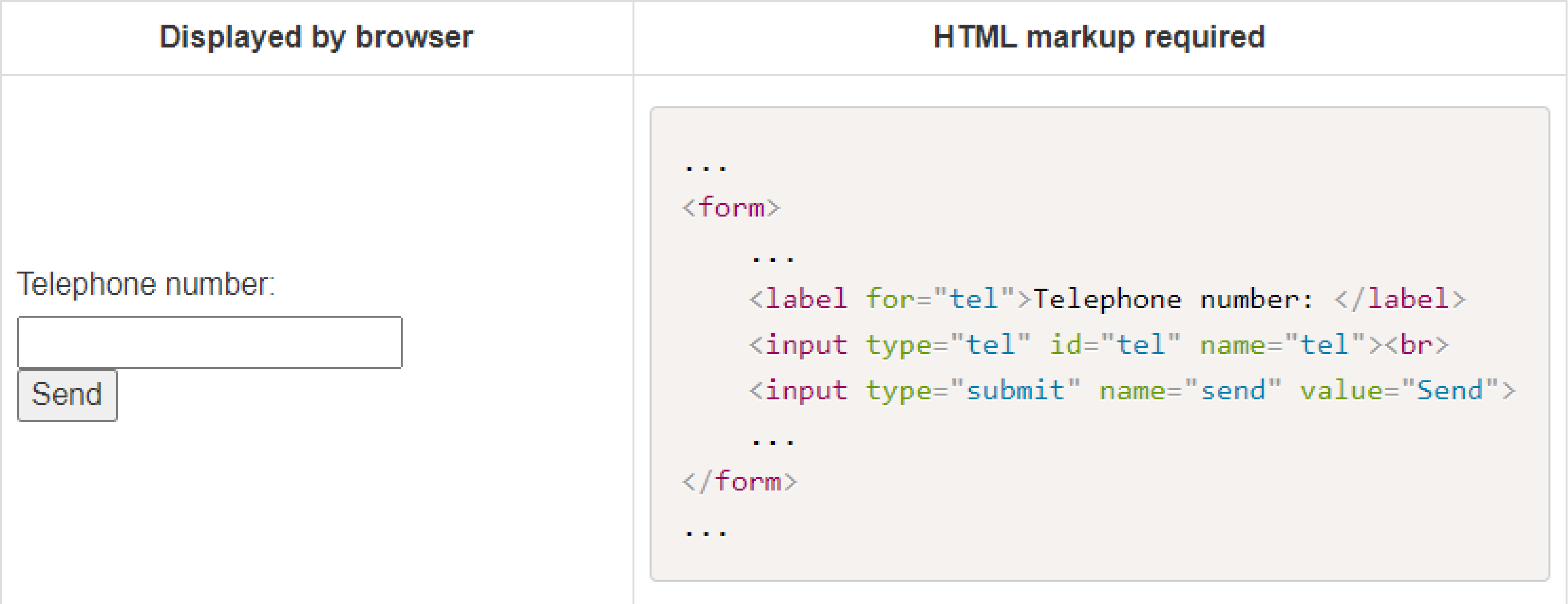


Displayed by browser	HTML markup required

```
...
<form>
    ...
    <label for="url">URL: </label>
    <input type="url" id="url" name="url"><br>
    <input type="submit" name="send" value="Send">
    ...
</form>
...
```

Tel input field

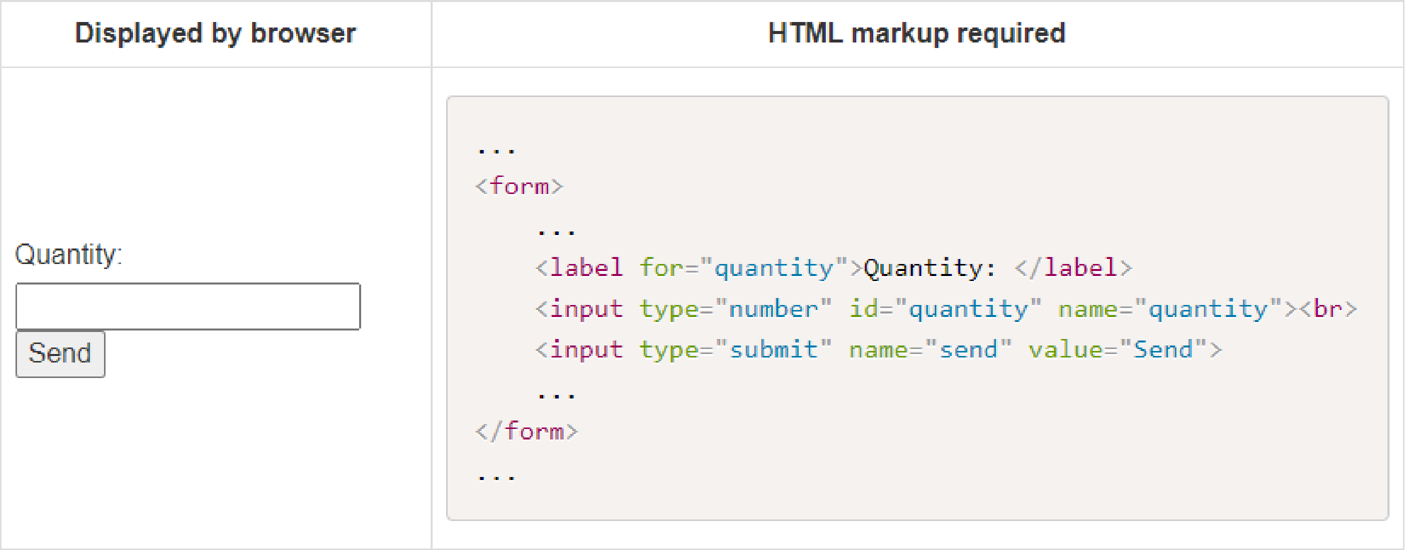
- Tel is an HTML5 input type.
- Browsers that support this input type will automatically check the validity of the format of the data entered and alert the user if the format is invalid.
- If the browser doesn't support this input type, it will automatically fall back to the type="text



<table>
<tr><th>Displayed by browser</th><th>HTML markup required</th></tr>
<tr><td>Telephone number:
Send</td><td><pre>...
<form>
    ...
    <label for="tel">Telephone number: </label>
    <input type="tel" id="tel" name="tel">
    <input type="submit" name="send" value="Send">
    ...
</form>
...</pre></td></tr>
</table>

Number input field

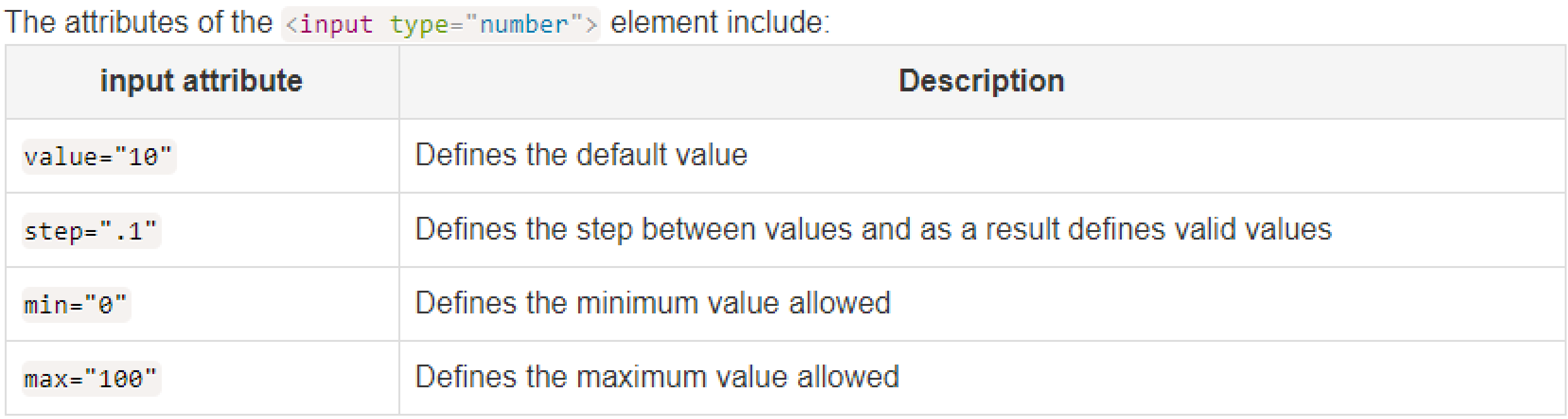
- number is an HTML5 input type.
- This input type is designed to let the user enter a number.
- Browsers that support this input type will automatically check the validity of the format of the data entered and alert the user if the format is invalid.
- The browser may also display up and down arrows for the user to click on to increase and decrease the value entered.
- If the browser doesn't support this input type, it will automatically fall back to the type="text"



Displayed by browser	HTML markup required

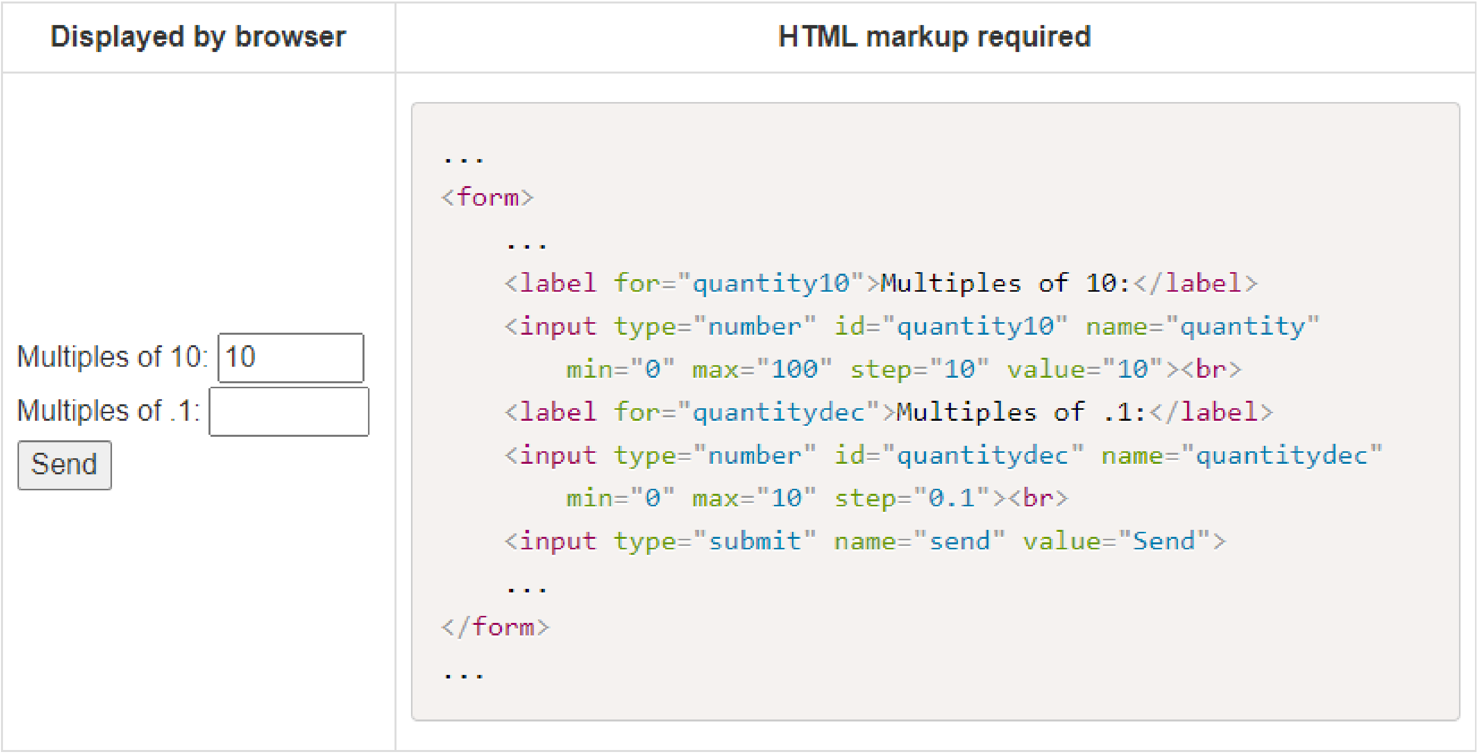
```
...
<form>
    ...
    <label for="quantity">Quantity: </label>
    <input type="number" id="quantity" name="quantity"><br>
    <input type="submit" name="send" value="Send">
    ...
</form>
...
```

Contc



The attributes of the `<input type="number">` element include:

input attribute	Description
`value="10"`	Defines the default value
`step=".1"`	Defines the step between values and as a result defines valid values
`min="0"`	Defines the minimum value allowed
`max="100"`	Defines the maximum value allowed

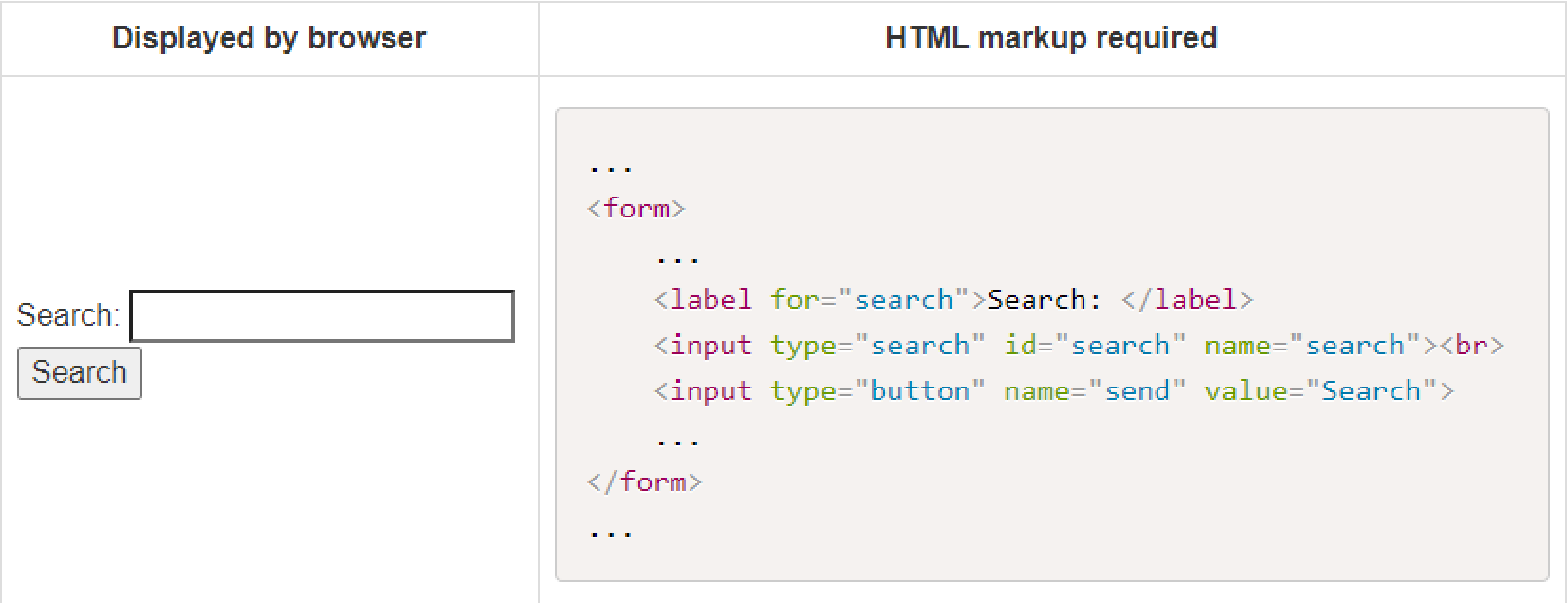


Displayed by browser	HTML markup required
Multiples of 10: 10 Multiples of .1: Send	see below

```
...
<form>
    ...
    <label for="quantity10">Multiples of 10:</label>
    <input type="number" id="quantity10" name="quantity"
        min="0" max="100" step="10" value="10"><br>
    <label for="quantitydec">Multiples of .1:</label>
    <input type="number" id="quantitydec" name="quantitydec"
        min="0" max="10" step="0.1"><br>
    <input type="submit" name="send" value="Send">
    ...
</form>
...
```

Search Input field

- search is an HTML5 input type.
- This input type is designed to let the user enter a search query. Browsers that support this input type may style it differently from the text input and may automatically offer the possibility to clear the input field.
- If the browser doesn't support this input type, it will automatically fall back to the type="text"



Displayed by browser | **HTML markup required**

Search:

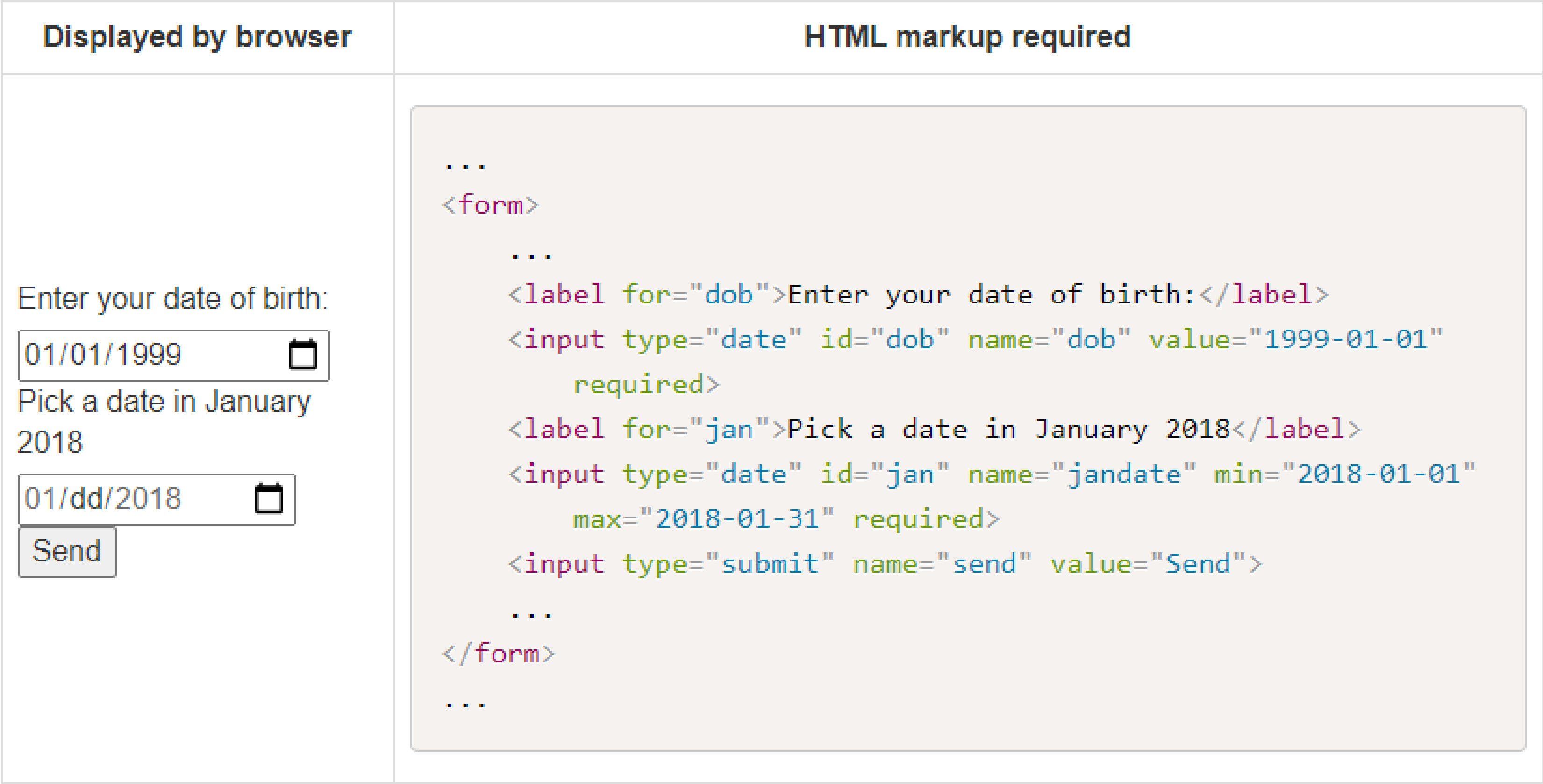
Search

```
...
<form>
    ...
    <label for="search">Search: </label>
    <input type="search" id="search" name="search"><br>
    <input type="button" name="send" value="Search">
    ...
</form>
...
```

Date picker

- There are a number of attributes that will create a date picker using the input element:
- date is an HTML5 input type.
- This input type is designed to let the user enter a date. Browsers that support this input type will automatically check the validity of the format of the data entered and alert the user if the format is invalid.
- The browser may also display a date picker. If the browser doesn't support this input type, it will automatically fall back to the type="text".
- You can provide a default value in the format yyyy-mm-dd, and you may use min and max date values if you want to retrict the valid date.
- Current Browser Support for date and time input

Date



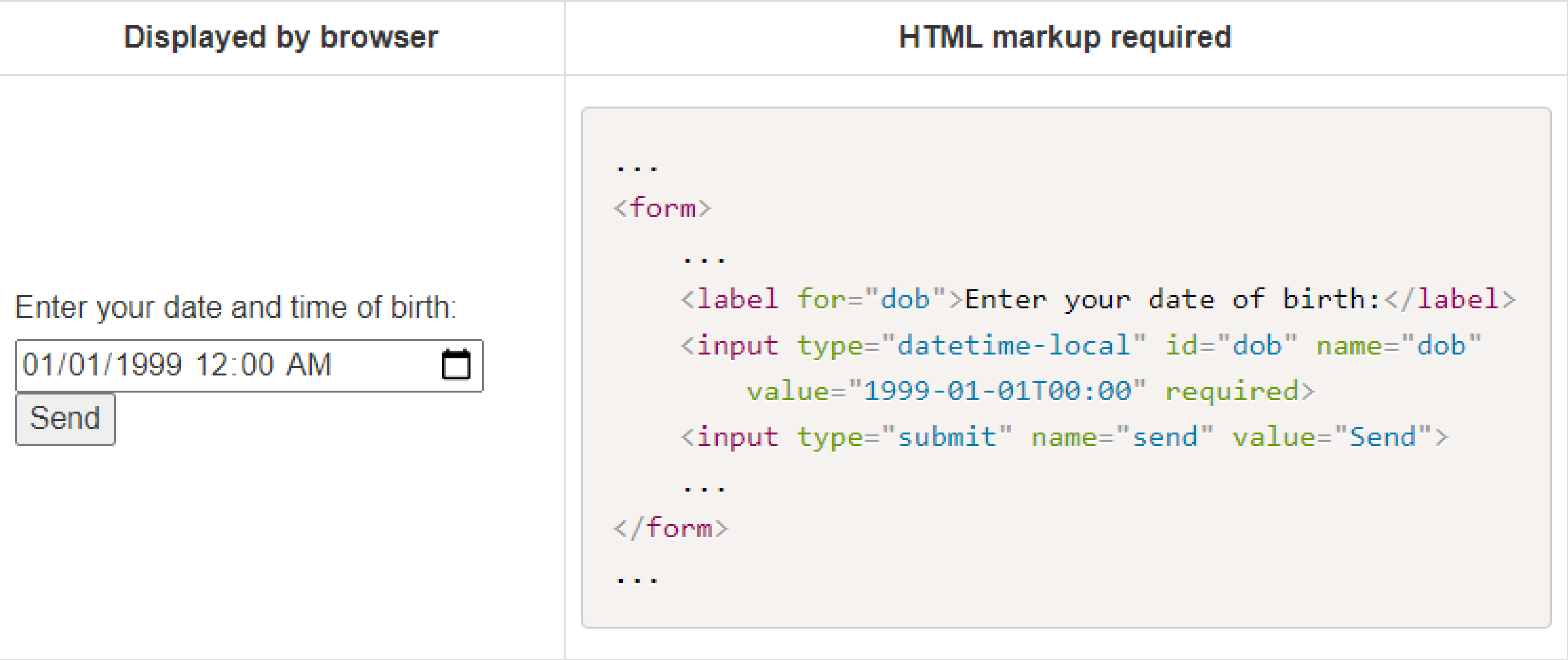
Displayed by browser	HTML markup required
Enter your date of birth: 01/01/1999 Pick a date in January 2018 01/dd/2018 Send	

```
...
<form>
    ...
    <label for="dob">Enter your date of birth:</label>
    <input type="date" id="dob" name="dob" value="1999-01-01"
        required>
    <label for="jan">Pick a date in January 2018</label>
    <input type="date" id="jan" name="jandate" min="2018-01-01"
        max="2018-01-31" required>
    <input type="submit" name="send" value="Send">
    ...
</form>
...
```

Datetime-local

- datetime-local is an HTML5 input type.
- This input type is designed to let the user enter a date and time in the user's local time zone.
- Browsers that support this input type will automatically check the validity of the format of the data entered and alert the user if the format is invalid.
- The browser may also display a date and time picker. If the browser doesn't support this input type, it will automatically fall back to the type="text".
- You can provide a default value in the format yyyy-mm-ddThh:mm, and you may use min and max date and time values if you want to restrict the valid date/time.

Datetime-local

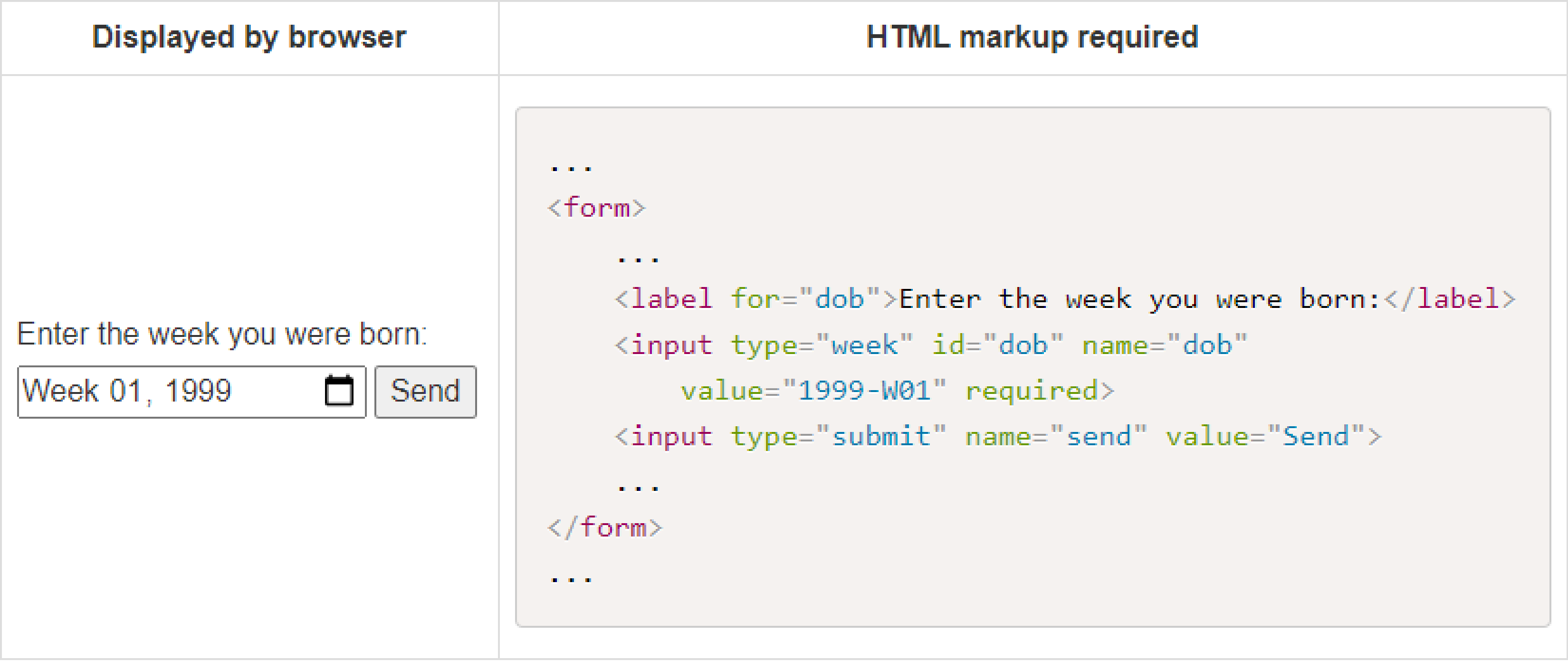


Displayed by browser	HTML markup required
Enter your date and time of birth: 01/01/1999 12:00 AM Send	see markup below

```
...
<form>
    ...
    <label for="dob">Enter your date of birth:</label>
    <input type="datetime-local" id="dob" name="dob"
        value="1999-01-01T00:00" required>
    <input type="submit" name="send" value="Send">
    ...
</form>
...
```

Week

- week is an HTML5 input type.
- You can provide a default value in the format yyyy-Www, and you may use min and max date and time values if you want to restrict the valid week.



Displayed by browser

Enter the week you were born:

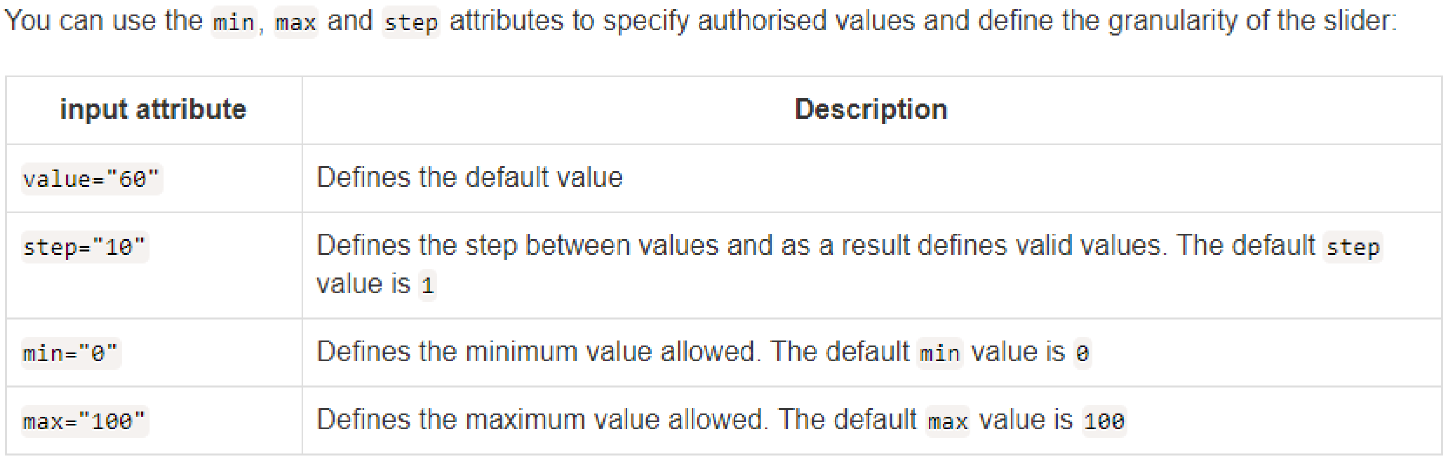
Week 01, 1999 | Send

HTML markup required

```
...
<form>
    ...
    <label for="dob">Enter the week you were born:</label>
    <input type="week" id="dob" name="dob"
        value="1999-W01" required>
    <input type="submit" name="send" value="Send">
    ...
</form>
...
```

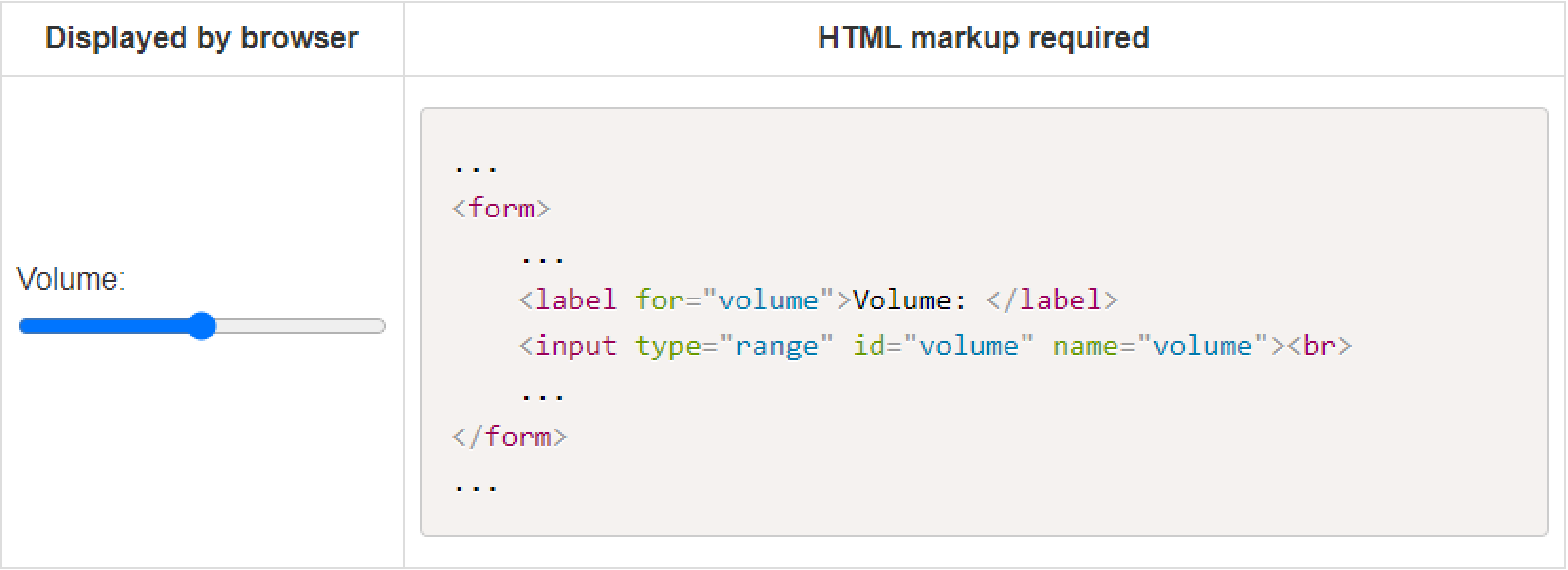
Range input fie

- range is an HTML5 input type



You can use the `min`, `max` and `step` attributes to specify authorised values and define the granularity of the slider:

input attribute	Description
`value="60"`	Defines the default value
`step="10"`	Defines the step between values and as a result defines valid values. The default `step` value is `1`
`min="0"`	Defines the minimum value allowed. The default `min` value is `0`
`max="100"`	Defines the maximum value allowed. The default `max` value is `100`

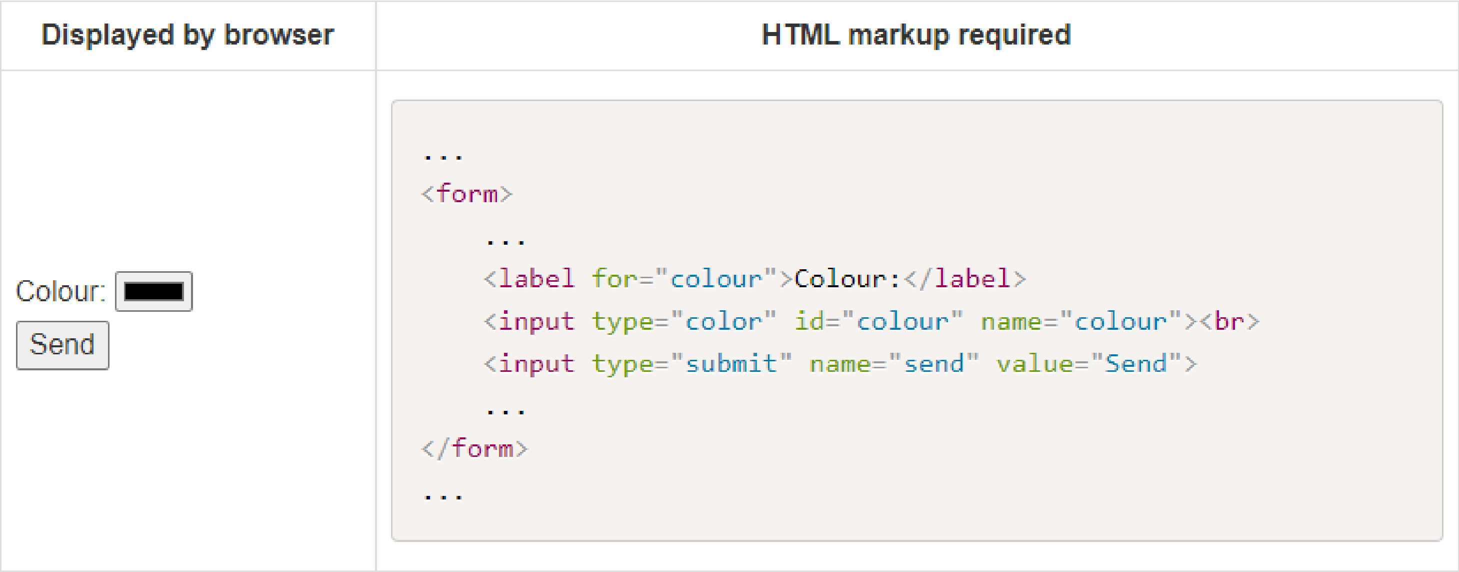


Displayed by browser	HTML markup required

Volume:	`...`
	`<form>`
	`    ...`
	`    <label for="volume">Volume: </label>`
	` <input type="range" id="volume" name="volume">
	`
	`    ...`
	`</form>`
	`...`

Color input field

- color is an HTML5 input type.
- This input type is designed to let the user enter a color in hexadecimal format.
- Browsers that support this input type may style it differently from the text input and may automatically present a colour picker.
- If no default value is specified, the default #000000 is used.

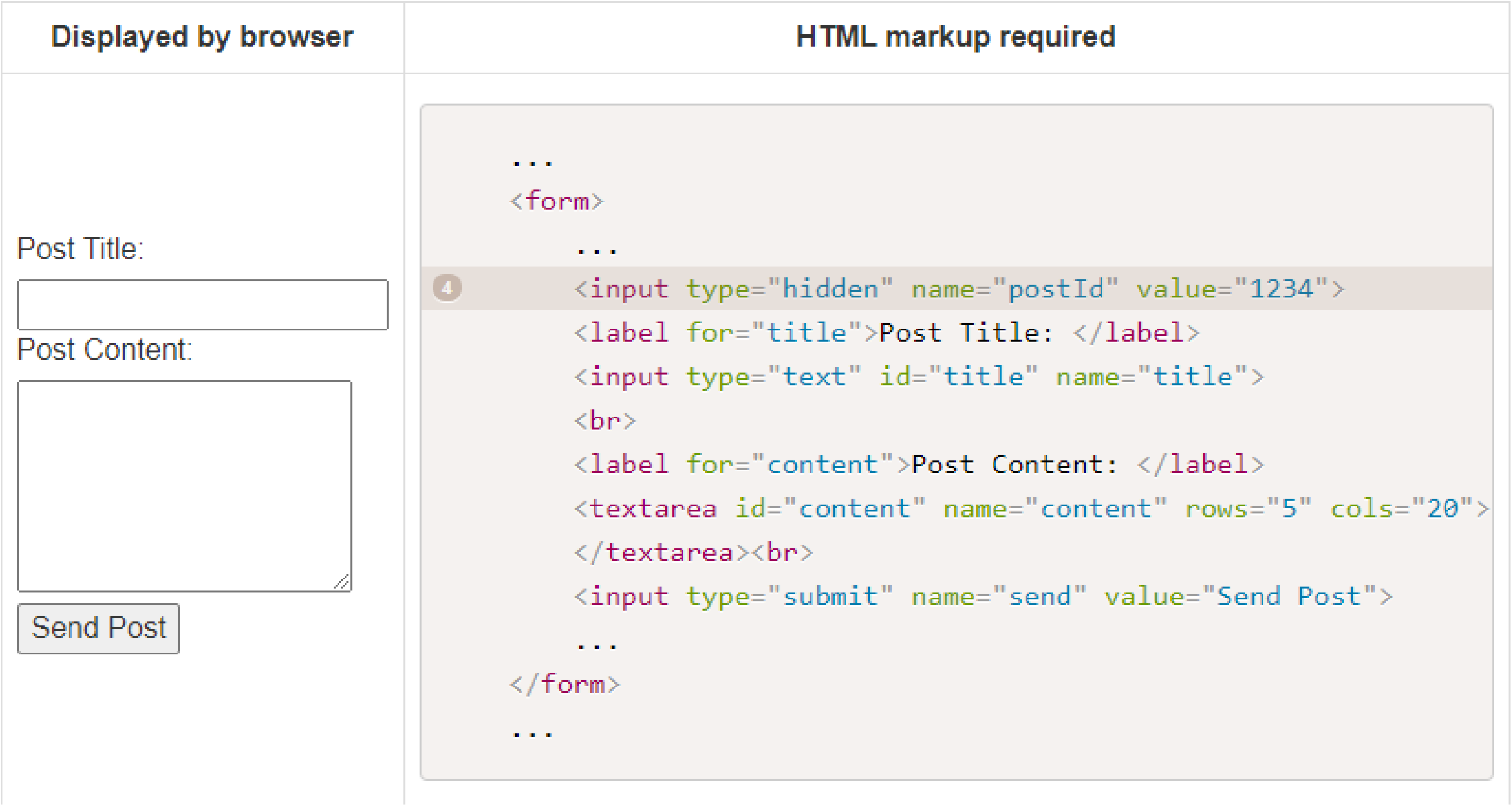


Displayed by browser	HTML markup required

Colour: ■
Send
	`...`
	`<form>`
	`    ...`
	`    <label for="colour">Colour:</label>`
	` <input type="color" id="colour" name="colour">
	`
	`    <input type="submit" name="send" value="Send">`
	`    ...`
	`</form>`
	`...`

Hidden field in a form

- Hidden fields are used to send parameters you don't want to appear in the displayed form.
- The name and value of the hidden field are incorporated into the parameter list sent, but do not have any visible counterpart on the form.



Displayed by browser

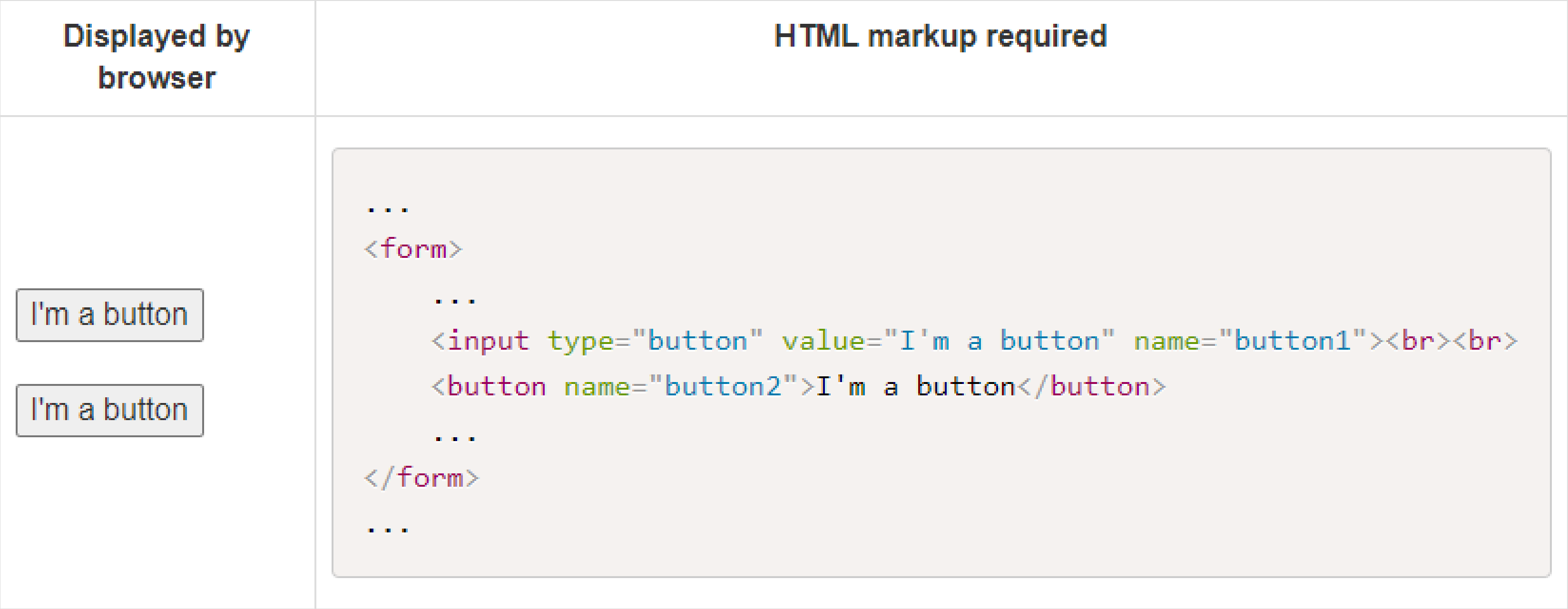
HTML markup required

```
...
<form>
    ...
    <input type="hidden" name="postId" value="1234">
    <label for="title">Post Title: </label>
    <input type="text" id="title" name="title">
    <br>
    <label for="content">Post Content: </label>
    <textarea id="content" name="content" rows="5" cols="20">
    </textarea><br>
    <input type="submit" name="send" value="Send Post">
    ...
</form>
...
```

Buttons

- There are several ways to create a button on a form, and there are several types of buttons available.
- You can create a button by adding the type="button" attribute to the input element, or by using the button element.
- These two types of button will have no default behaviour, other than being a clickable button, and you will need to add some JavaScript to add event listeners to these buttons and add behaviour (for example calculating the total price of an order).
- Please see in the last section of this document for details on Submit buttons and Reset buttons.

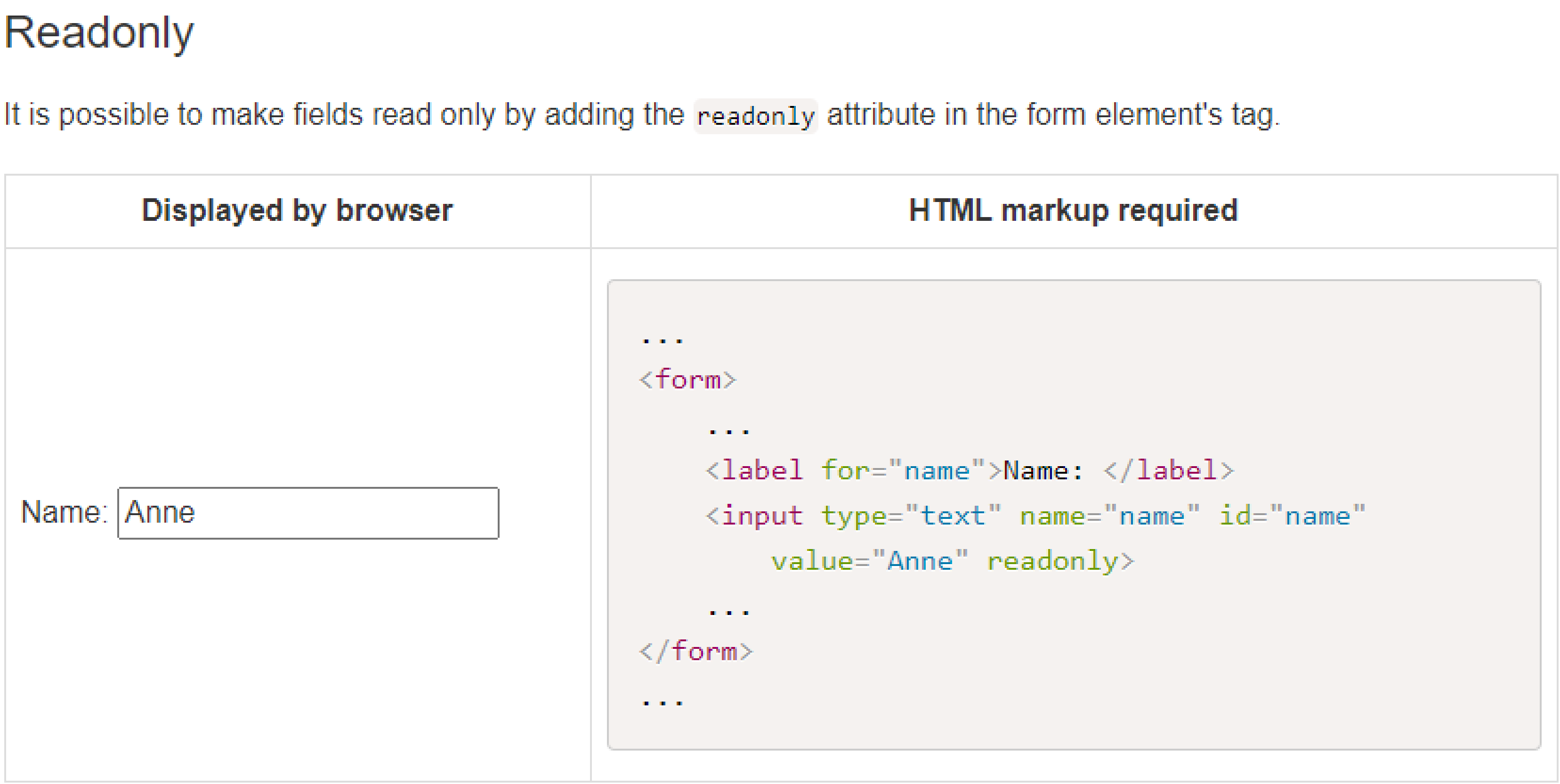
Buttons



Displayed by browser	HTML markup required

I'm a button
I'm a button
	...
	<form>
	    ...
	    <input type="button" value="I'm a button" name="button1">
	
	    <button name="button2">I'm a button</button>
	    ...
	</form>
	...

Useful attributes



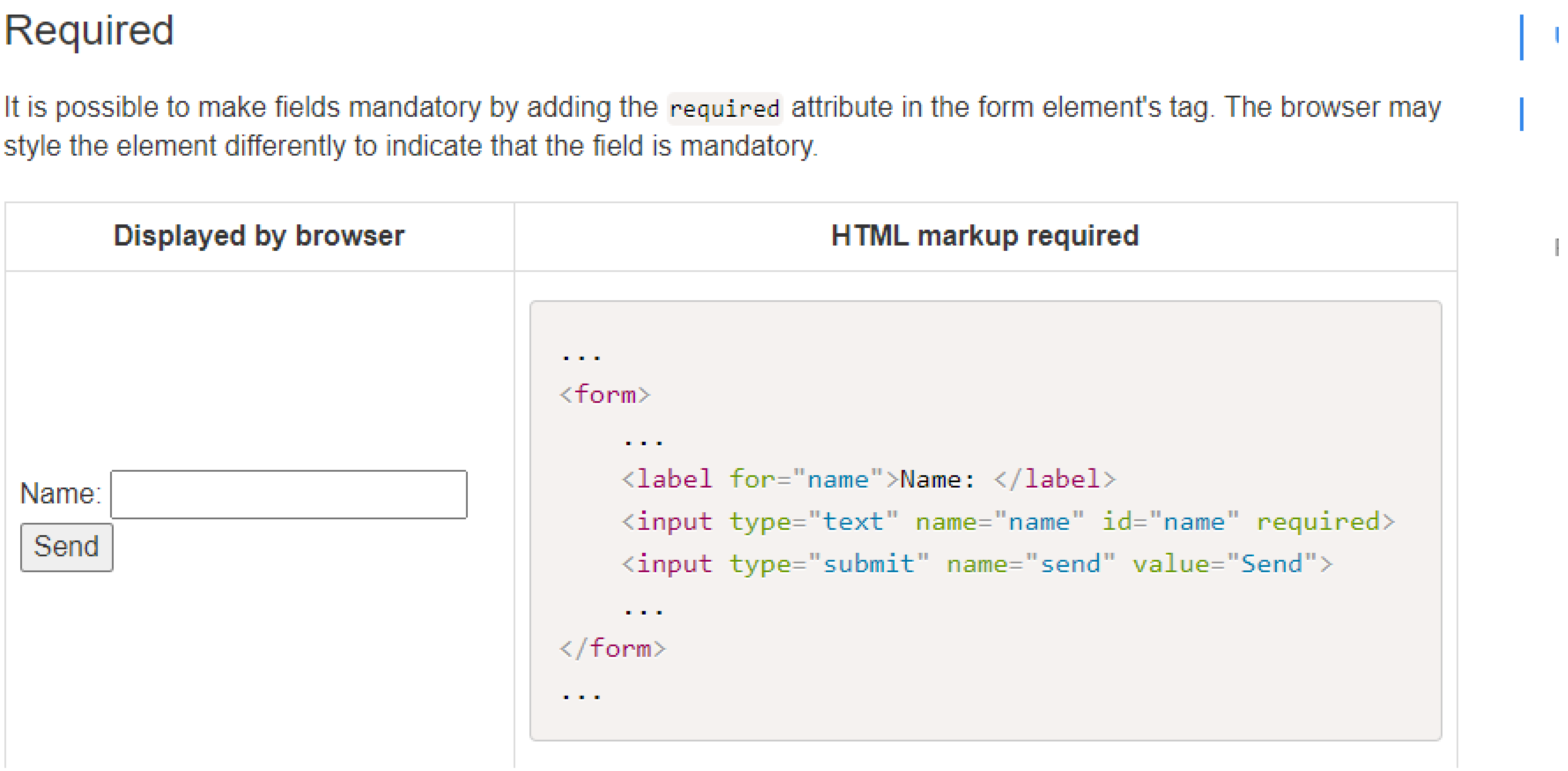
Readonly

It is possible to make fields read only by adding the `readonly` attribute in the form element's tag.

Displayed by browser	HTML markup required

Name: Anne	...
	<form>
	    ...
	    <label for="name">Name: </label>
	    <input type="text" name="name" id="name"
	        value="Anne" readonly>
	    ...
	</form>
	...

Useful Attributes



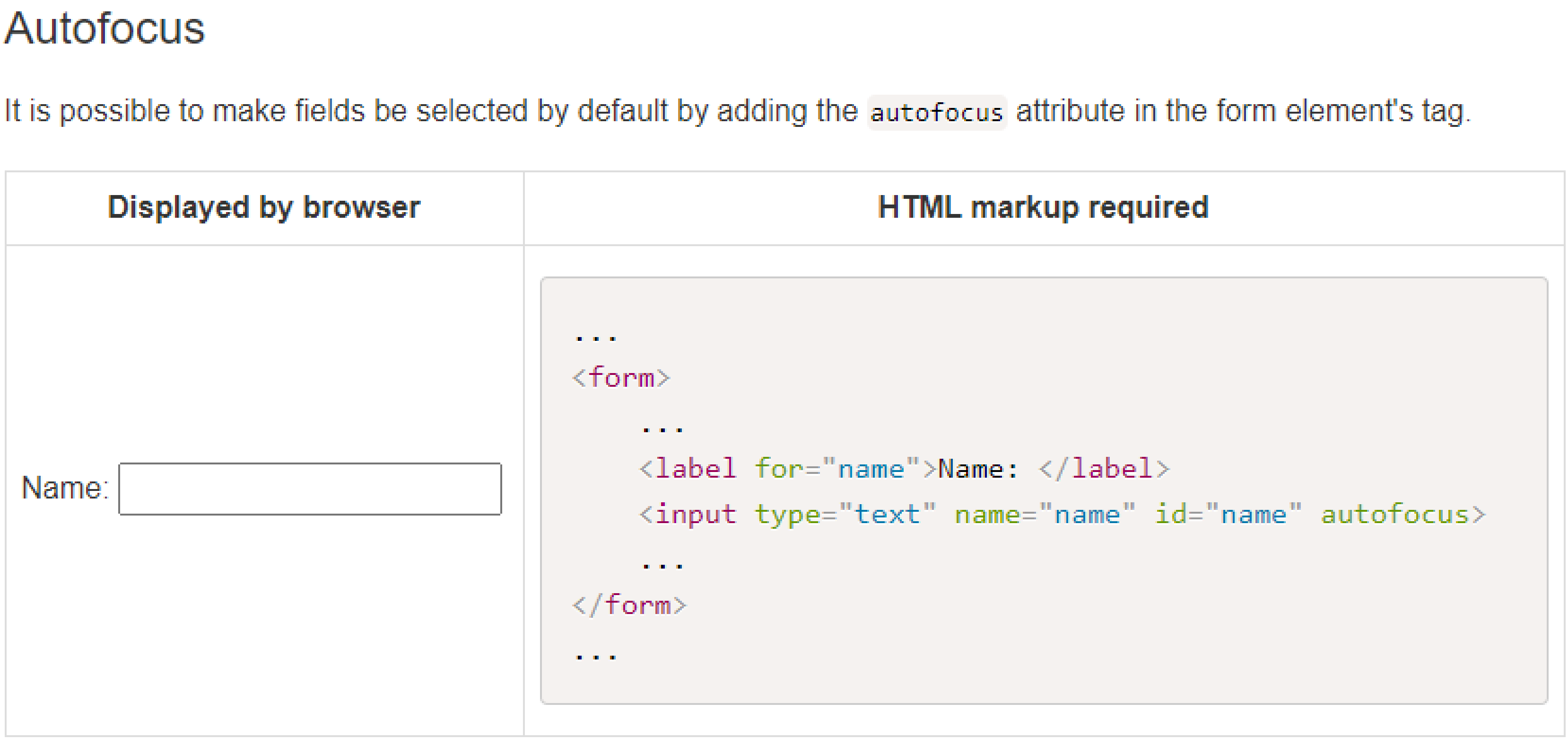
Required

It is possible to make fields mandatory by adding the `required` attribute in the form element's tag. The browser may style the element differently to indicate that the field is mandatory.

Displayed by browser	HTML markup required

Name:
 Send
	`...`
	`<form>`
	`    ...`
	`    <label for="name">Name: </label>`
	`    <input type="text" name="name" id="name" required>`
	`    <input type="submit" name="send" value="Send">`
	`    ...`
	`</form>`
	`...`

Useful Attributes



Autofocus

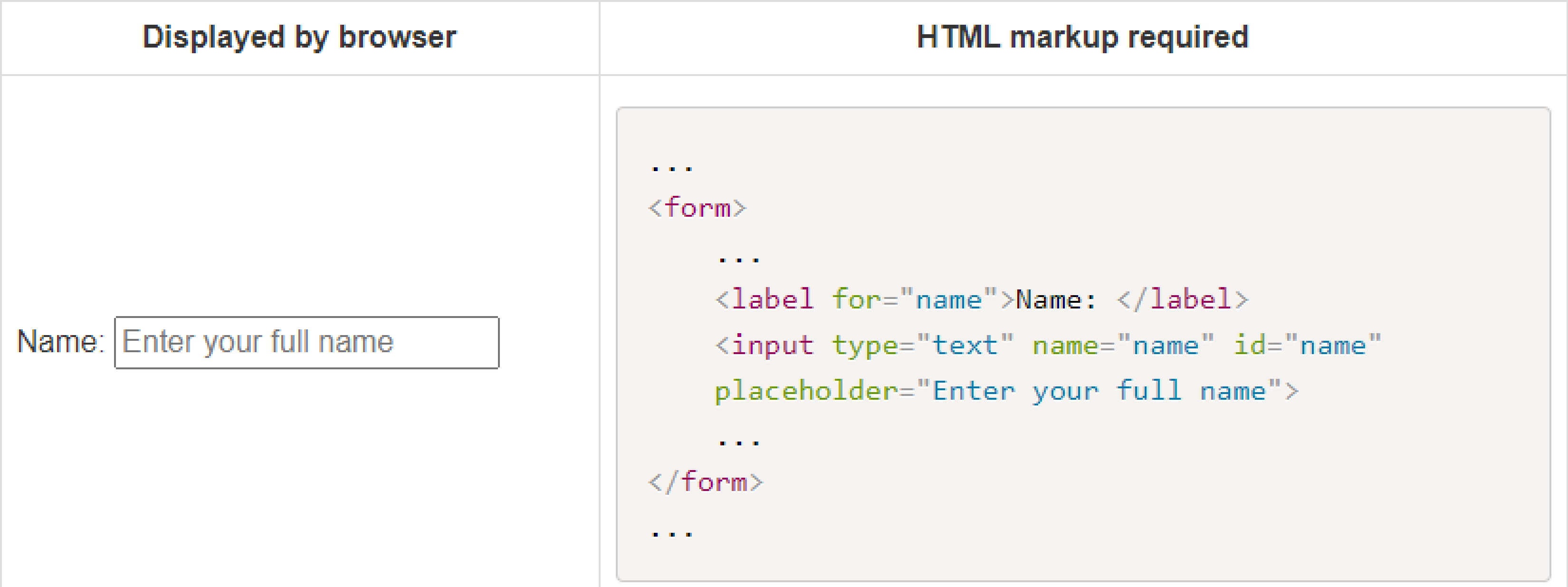
It is possible to make fields be selected by default by adding the `autofocus` attribute in the form element's tag.

Displayed by browser	HTML markup required

Name:	`...`
	`<form>`
	`    ...`
	`    <label for="name">Name: </label>`
	`    <input type="text" name="name" id="name" autofocus>`
	`    ...`
	`</form>`
	`...`

Useful Attributes

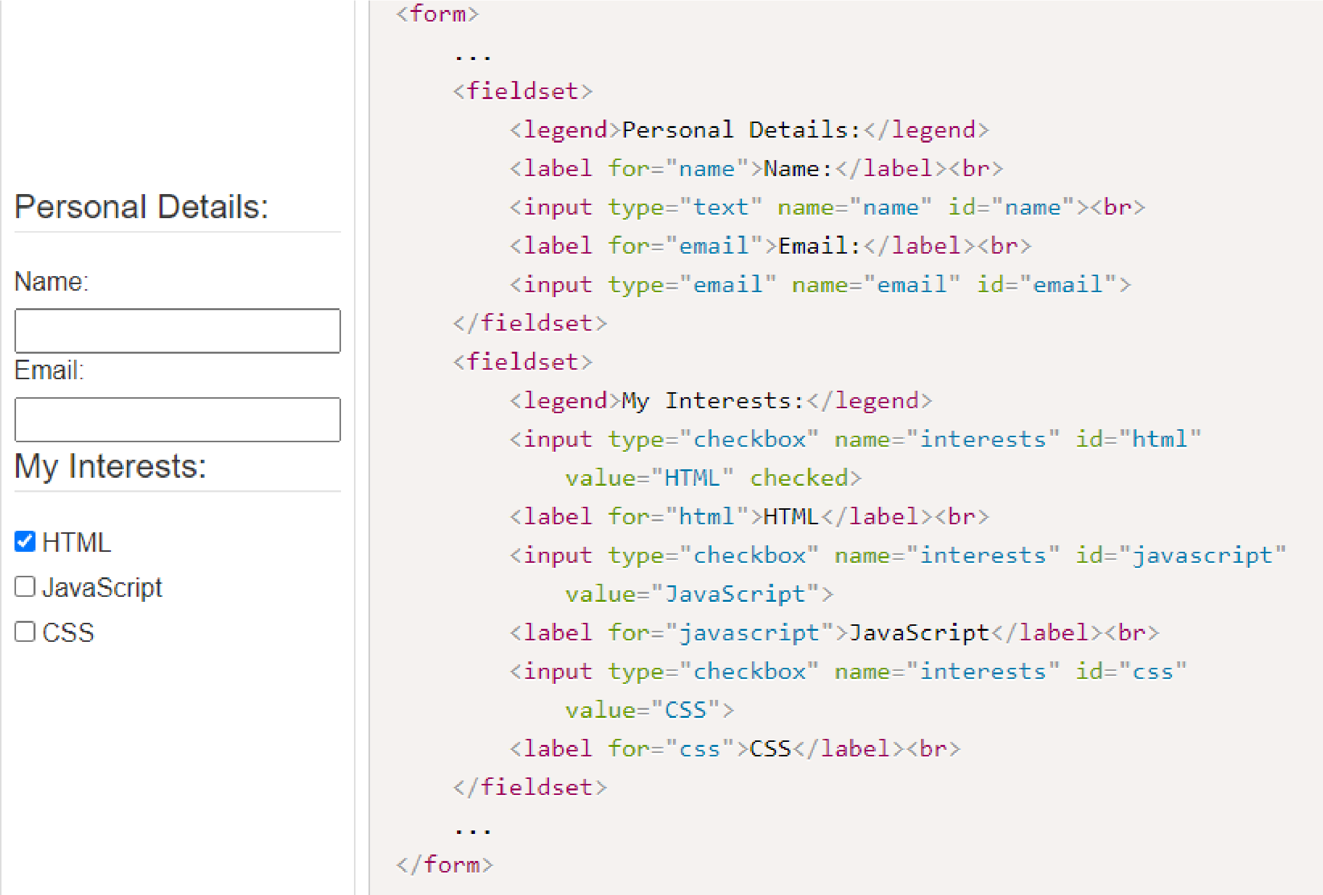
- It is possible to add a placeholder in the input field by adding the placeholder attribute and spacifying a value in the form element's tag.
- The advantage of the placeholder over a default value, is that the placeholder automatically disappears when the user starts entering text, and will be displayed again if the user deletes what they entered, hence the information is never lost, and never needs additional user interaction to be removed.



Displayed by browser	HTML markup required
Name: Enter your full name	... <form> ... <label for="name">Name: </label> <input type="text" name="name" id="name" placeholder="Enter your full name"> ... </form> ...

Fieldset and Legend

- The fieldset element togehter with its corresponding legend are used to group several form controls and labels together.



Personal Details:

Name:

Email:

My Interests:

☑ HTML

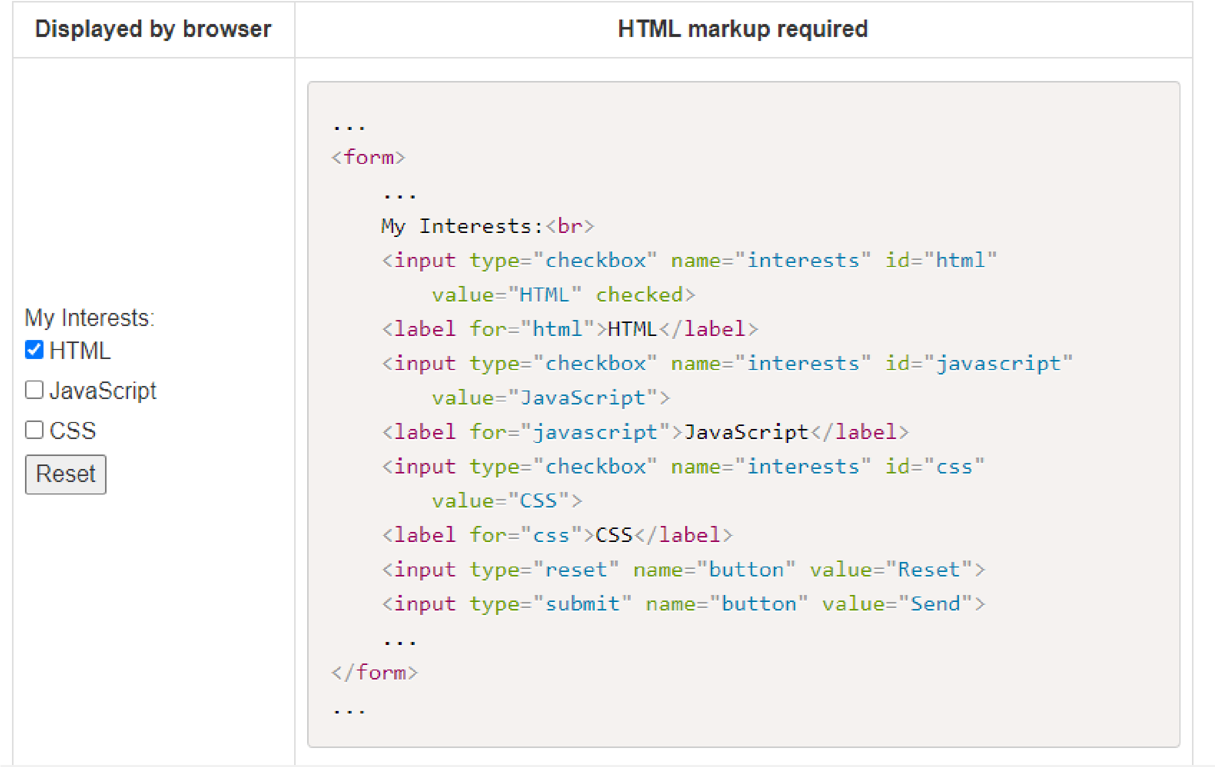
☐ JavaScript

☐ CSS

```
<form>
    ...
    <fieldset>
        <legend>Personal Details:</legend>
        <label for="name">Name:</label><br>
        <input type="text" name="name" id="name"><br>
        <label for="email">Email:</label><br>
        <input type="email" name="email" id="email">
    </fieldset>
    <fieldset>
        <legend>My Interests:</legend>
        <input type="checkbox" name="interests" id="html"
            value="HTML" checked>
        <label for="html">HTML</label><br>
        <input type="checkbox" name="interests" id="javascript"
            value="JavaScript">
        <label for="javascript">JavaScript</label><br>
        <input type="checkbox" name="interests" id="css"
            value="CSS">
        <label for="css">CSS</label><br>
    </fieldset>
    ...
</form>
```

Reset

- The <input> tag with an attribute of type="reset" is used to reset the values in a form back to their default value.
- For example, the following form may be reset to its initial values by pressing the "reset" button.

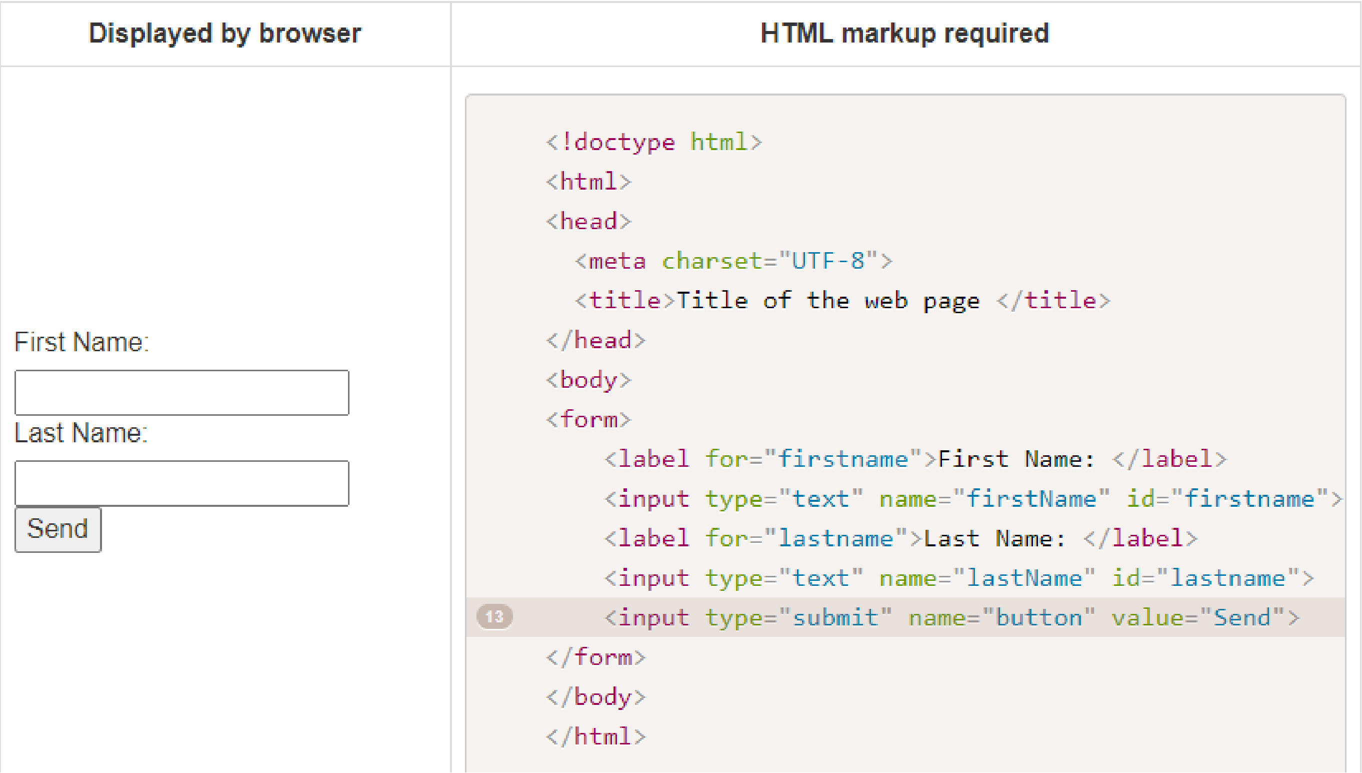


Displayed by browser	HTML markup required
My Interests: ☑ HTML □ JavaScript □ CSS [Reset]	

```
...
<form>
    ...
    My Interests:<br>
    <input type="checkbox" name="interests" id="html"
        value="HTML" checked>
    <label for="html">HTML</label>
    <input type="checkbox" name="interests" id="javascript"
        value="JavaScript">
    <label for="javascript">JavaScript</label>
    <input type="checkbox" name="interests" id="css"
        value="CSS">
    <label for="css">CSS</label>
    <input type="reset" name="button" value="Reset">
    <input type="submit" name="button" value="Send">
    ...
</form>
...
```

Submit Button

- When a form contains a single entry field, the information is sent as soon as the user presses the "return" key.
- However, most forms will contain multiple form elements and therefore require a "submit button", generated by an additional tag <input type="submit"> to cause the submission of the input data:

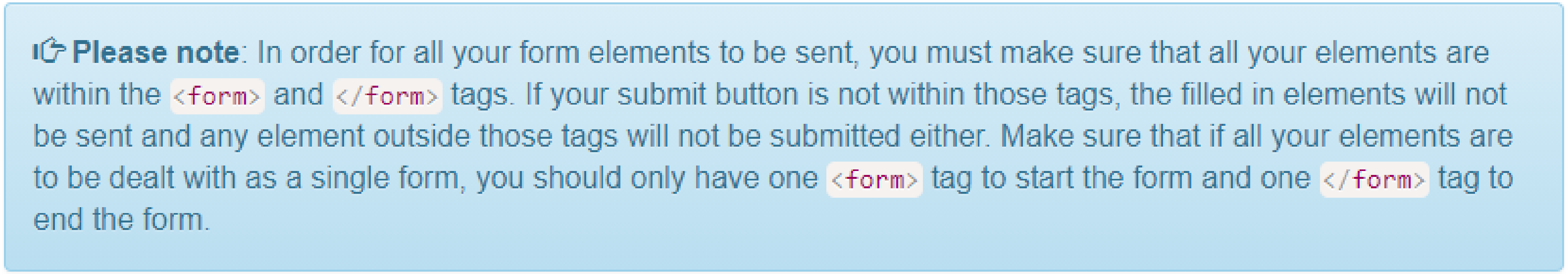


Displayed by browser | HTML markup required

```
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Title of the web page </title>
</head>
<body>
<form>
    <label for="firstname">First Name: </label>
    <input type="text" name="firstName" id="firstname">
    <label for="lastname">Last Name: </label>
    <input type="text" name="lastName" id="lastname">
    <input type="submit" name="button" value="Send">
</form>
</body>
</html>
```

Form submission

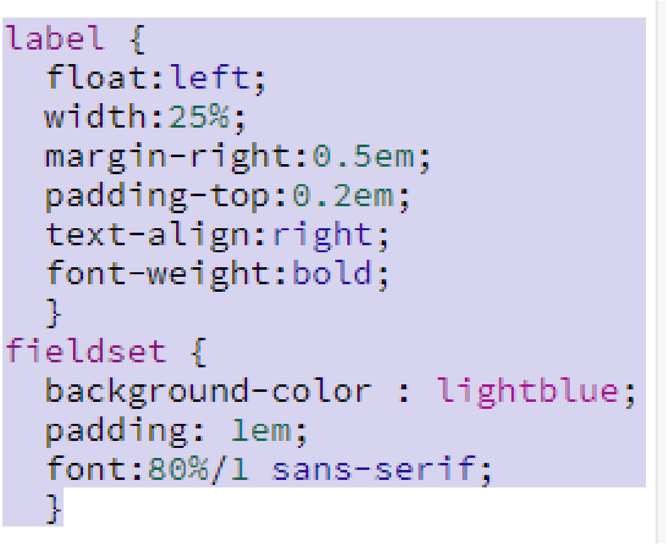
- When there are several elements in a form the data sent to the server-side script is composed of the individual elements concatenated together with an &.
- For example, when the Send button is pressed and the name Jane Smith is entered, then the following information will be sent: firstName=Jane&lastName=Smith&button=Send
- https://www.w3schools.com/tags/tryit.asp?filename=tryhtml_form_method



☞ **Please note**: In order for all your form elements to be sent, you must make sure that all your elements are within the `<form>` and `</form>` tags. If your submit button is not within those tags, the filled in elements will not be sent and any element outside those tags will not be submitted either. Make sure that if all your elements are to be dealt with as a single form, you should only have one `<form>` tag to start the form and one `</form>` tag to end the form.

If you only want to style a specific input type, you can use attribute selectors:

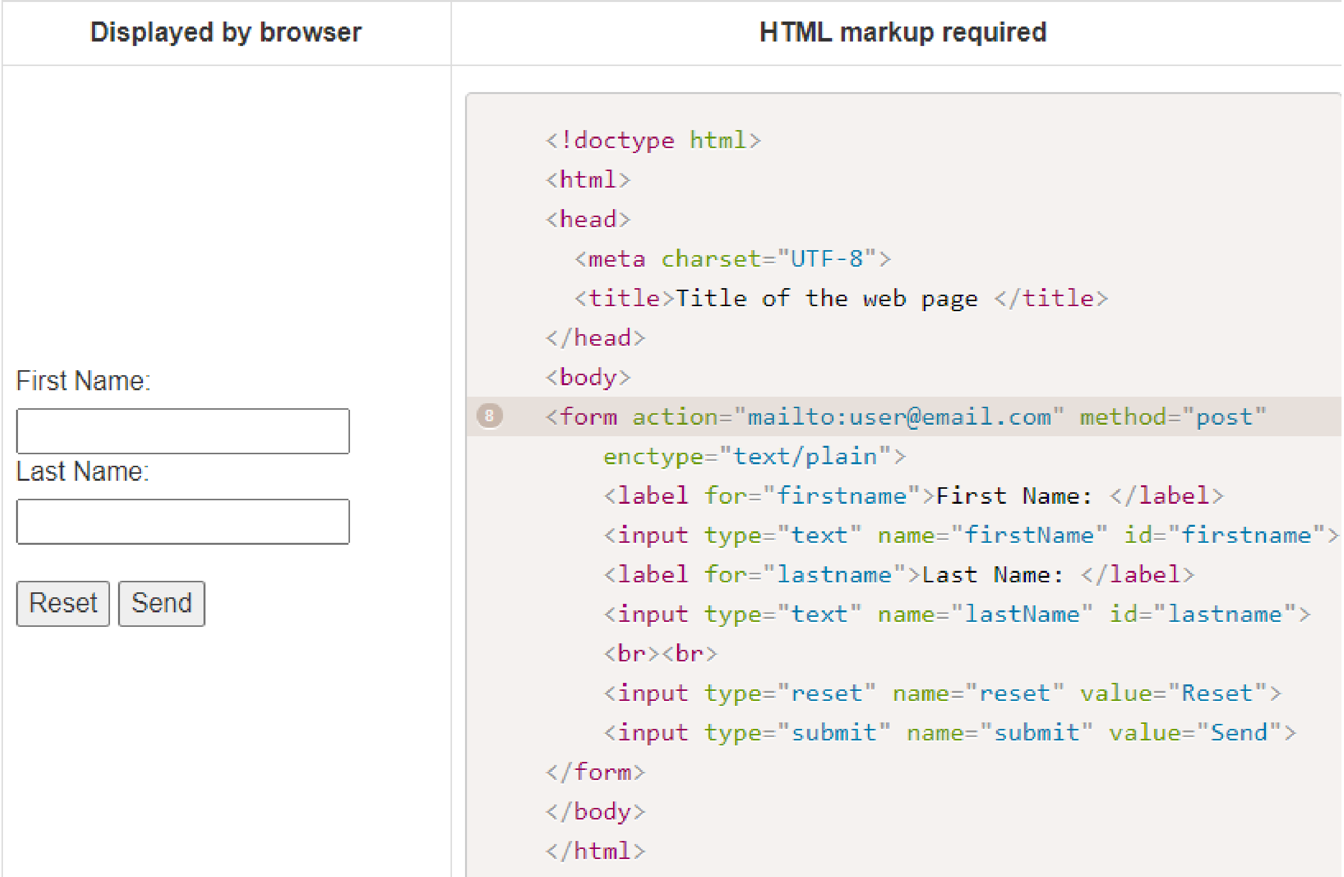
- input[type=text] - will only select text fields
- input[type=password] - will only select password fields
- input[type=number] - will only select number fields
- etc..



```
label {
  float:left;
  width:25%;
  margin-right:0.5em;
  padding-top:0.2em;
  text-align:right;
  font-weight:bold;
  }
fieldset {
  background-color : lightblue;
  padding: 1em;
  font:80%/1 sans-serif;
  }
```

Emailing the content of a form

- By adding the following attributes: method="post" action="mailto:Your e-mail address" enctype="text/plain" to your form element.
- When a user fills in and submits the form, the content of the form is posted to your e-mail address. Of course for this to work, the mail preferences must be set up in the users browser.



Displayed by browser

HTML markup required

```html
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Title of the web page </title>
</head>
<body>
<form action="mailto:user@email.com" method="post"
    enctype="text/plain">
    <label for="firstname">First Name: </label>
    <input type="text" name="firstName" id="firstname">
    <label for="lastname">Last Name: </label>
    <input type="text" name="lastName" id="lastname">
    <br><br>
    <input type="reset" name="reset" value="Reset">
    <input type="submit" name="submit" value="Send">
</form>
</body>
</html>
```