FORMS

INFOCUS

Forms are used on web pages as a means of interacting with users. They provide a way to collect information, such as comments and registration information, and to guide the user about the type of information that is required.

Forms can be created in HTML, but they require a program on the web server to process the information. The programming side is beyond the scope of this course, but if you build a form, you will need to talk to your hosting provider or a Web developer about how to collect and store the information, and what to do with it once it is collected.

In this session you will:

- ✓ learn how to create an ***HTML*** form
- ✓ learn how to create text input fields
- ✓ learn how to set autofocus and placeholder text
- ✓ learn how to create radio buttons
- ✓ learn how to create checkboxes
- ✓ learn how to create a drop-down list
- ✓ learn how to create a scrollable list
- ✓ learn how to create a text area
- ✓ learn how to create reset and submit buttons
- ✓ learn how to validate form inputs
- ✓ gain an understanding of more form controls.

CREATING A FORM

Forms are defined in HTML using the ***<form>*** tag. Within these tags you can place a range of different elements to create the fields on the form. A form is used to collect information from visitors to a web page, and requires a program on the web server to process the information. In this exercise we will just be creating a form in HTML without worrying about processing its information.

Try This Yourself:

Open File

Before starting this exercise you MUST open the file Forms_1.html...

1. In your text editor, type the code for the form as shown

   The action attribute specifies the page that this form's information will be sent to for processing. Usually this would be a program on your web server but for this exercise we are just sending it to the Thanks.html page…

2. Save the changes to the file
3. Open the file in your web browser to display the form

   There's nothing to see here yet as a form needs input fields before it is useful

```
<!-- Created for Polished Pools by PP Web Design -->
<!DOCTYPE html>
<html>
<head>
  <title>Polished Pools Help Questionnaire</title>
</head>
<body >
  <h1>Quest for Help Questionnaire</h1>
  <h3>Let us in on your darkest pool problems...</h3>
  <hr />
  <form action="Thanks.html">

  </form>
</body>
</html>
```

1

3

For Your Reference…

To ***create*** a ***form***:

1. Add the opening ***<form>*** tag
2. Add the ***action*** attribute with the address of the program that will process the form's information
3. Add the closing ***</form>*** tag

Handy to Know…

- You can use the ***method*** attribute to specify the way the form's information is sent to the web server. There are two options: GET and POST. The GET option sends the information in the URL while the POST option hides the information and should be used for most forms.

CREATING TEXT INPUT FIELDS

Within the <form> tags you can place a range of different elements to create the fields on the form. A text input field is simply a text box that is used to enter text. It is ideal for names, addresses, and any other field where you want the user to enter information on a single line. Each text input field is given a label, ID and name.

Try This Yourself:

Open File

Continue using the previous file with this exercise or open the file Forms_2.html in the text editor and browser...

1. In your text editor, type the code for the labels and text input fields as shown

   Labels are not strictly necessary, but they help with usability (when a label is clicked its associated control might be selected) and accessibility (as screen readers will be able to tell which label describes which control)…

2. Save the changes to the file

3. Switch to your web browser, then click on ***Refresh*** to display the text input fields

   *The first four fields have been created – either on new lines using
 or on the same line as in the case of Suburb/Town and Postcode*

```
<h1>Quest for Help Questionnaire</h1>
<h3>Let us in on your darkest pool problems...</h3>
<hr />
<form action="Thanks.html">
  <h4>Name and Address Details</h4>
  <label for="name-input">Name</label>
  <input type="text" id="name-input" name="Name" />
  <br />
  <label for="suburb-input">Suburb/Town</label>
  <input type="text" id="suburb-input" name="Suburb" />
  <label for="postcode-input">Postcode</label>
  <input type="text" id="postcode-input" name="Postcode" />
</form>
```

1

3

For Your Reference…

To ***create*** a ***text input field***:

1. Add the ***<input>*** tag
2. Add the attribute ***type="text"***
3. Add the ***id*** attribute
4. Add the ***name*** attribute

Handy to Know…

- The ***name*** attribute is used when the data is processed on the web server.
- You can use the ***maxlength*** attribute to control the maximum number of characters that can be entered in the field.

AUTOFOCUS AND PLACEHOLDER TEXT

You can make it easier for your users to fill in form data by making use of the ***autofocus*** and ***placeholder*** attributes on your text inputs. The autofocus attribute can be set to make an input active when the user opens the page, enabling them to start typing immediately. The placeholder attribute can be used to display a short hint to help the user know what type of data is expected.

Try This Yourself:

Open File

Continue using the previous file with this exercise or open the file Forms_3.html in the text editor and browser...

1. In your text editor, add the ***autofocus*** attribute to the name input as shown
2. Save the changes to the file
3. Switch to your web browser, refresh the page and then type your name

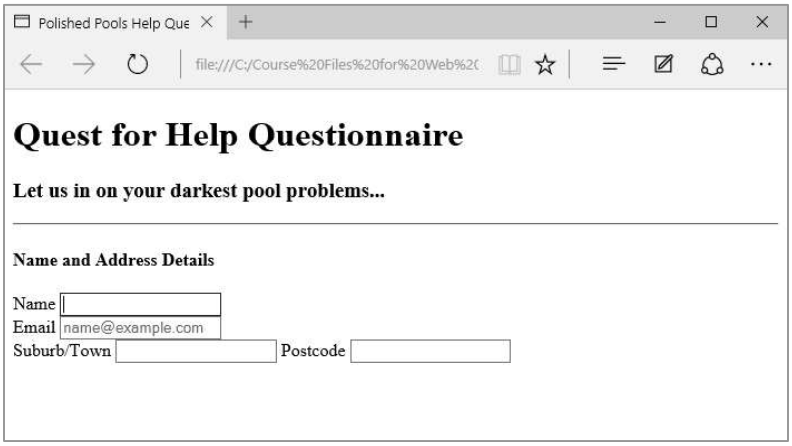
   When you refresh the page the mouse cursor will be positioned in the name text field and so typing your name will put it straight into the field
4. In your text editor, type the code for the email label and text input field as shown
5. Save the changes to the file
6. Switch to your web browser and refresh the page to display the email field with the placeholder text showing the user the required email format

```
<form action="Thanks.html">
  <h4>Name and Address Details</h4>
  <label for="name-input">Name</label>
  <input type="text" id="name-input" name="Name" autofocus />
  <br />
```

1

```
<form action="Thanks.html">
  <h4>Name and Address Details</h4>
  <label for="name-input">Name</label>
  <input type="text" id="name-input" name="Name" required />
  <br />
  <label for="email-input">Email</label>
  <input type="text" id="email-input" name="Email"
placeholder="example@example.com" />
  <br />
```

4



6

For Your Reference…

To ***make*** an ***input active*** when the ***page*** is ***loaded***:

- Add the ***autofocus*** attribute to the input

To ***add placeholder text*** to an ***input***:

- Add the ***placeholder*** attribute to the input with the text you would like to display to the user

Handy to Know…

- The ***autofocus*** attribute should only be set on one input per page.
- The ***placeholder*** attribute can be used to show the required format of the input (e.g. the email or date format), the use for the field (e.g. “Search”) or to show that the field is required.

CREATING RADIO BUTTONS

Radio buttons provide a list of mutually exclusive options for the user to select from. The advantages of using radio buttons are that you force the user to make only one selection, you can control the options listed making the purpose of the question unambiguous, and you avoid the risk of multiple names being typed in for the same type or option.

Try This Yourself:

Same File

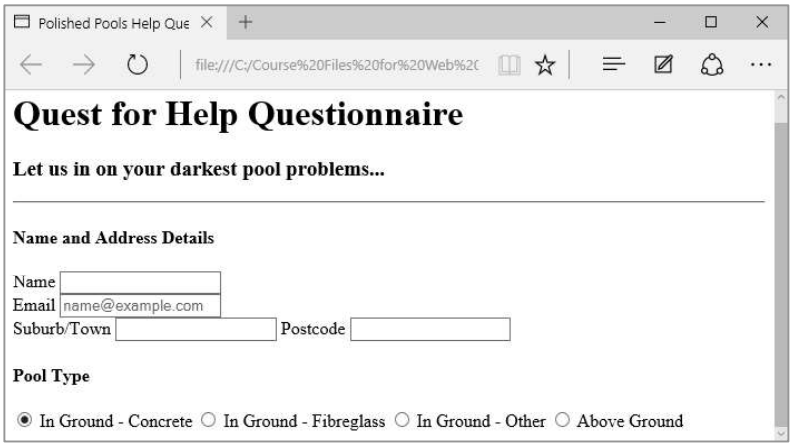
Continue using the previous file with this exercise or open the file Forms_4.html in the text editor and browser...

1. In your text editor, add the lines of code as shown to create the radio buttons
2. Save the changes to the file
3. Switch to your web browser and refresh the page

   The buttons and their text appear on the same line

```
  <label for="postcode-input">Postcode</label>
  <input type="text" id="postcode-input" name="Postcode" />
  <br />
  <h4>Pool Type</h4>
  <input type="radio" id="pool-type-1" name="PoolType"
value="InGroundConcrete" checked />
  <label for="pool-type-1">In Ground - Concrete</label>
  <input type="radio" id="pool-type-2" name="PoolType"
value="InGroundFibreglass" />
  <label for="pool-type-2">In Ground - Fibreglass</label>
  <input type="radio" id="pool-type-3" name="PoolType"
value="InGroundOther" />
  <label for="pool-type-3">In Ground - Other</label>
  <input type="radio" id="pool-type-4" name="PoolType"
value="AboveGround" />
  <label for="pool-type-4">Above Ground</label>
</form>
```

1



3

For Your Reference…

To ***create radio buttons***:

1. Add the ***<input>*** tag
2. Add the attribute ***type="radio"***
3. Add the ***id*** attribute
4. Add the ***name*** attribute
5. Add the attribute ***checked*** if this radio button should be checked by default

Handy to Know…

- The ***name*** attribute is used to group the radio buttons – all radio buttons in the same group should have the same name.
- When ***value*** is used with radio buttons, it is used to specify the value sent to the web server when the element is selected.

CREATING CHECKBOXES

Checkboxes are used to provide a list of options from which the user can select some, all, or none. The advantages of using checkboxes are that you provide the user with a series of defined responses making the question unambiguous, you enable the user to provide varying amounts of information, and you avoid the risk of multiple names being typed in for the same type or option.

Try This Yourself:

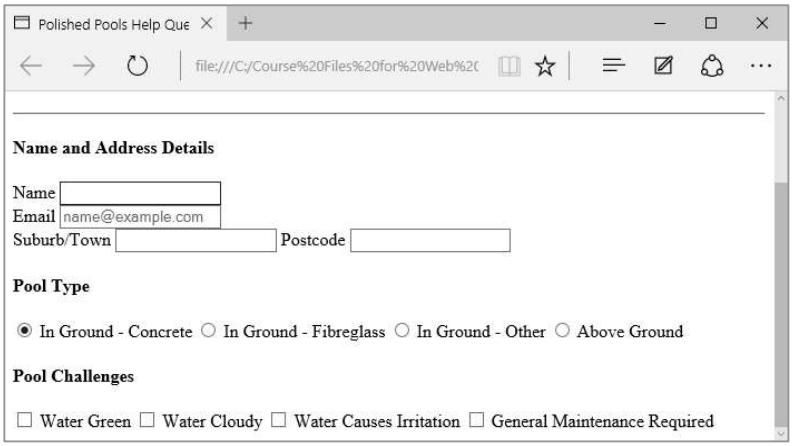
Same File

Continue using the previous file with this exercise or open the file Forms_5.html in the text editor and browser...

1. In your text editor, add the lines of code as shown to create the checkboxes
2. Save the changes to the file
3. Switch to your web browser and refresh the page to display the checkboxes

```
  <h4>Pool Challenges</h4>
  <input type="checkbox" id="challenge-1" value="WaterGreen"
/>
  <label for="challenge-1">Water Green</label>
  <input type="checkbox" id="challenge-2"
value="WaterCloudy" />
  <label for="challenge-2">Water Cloudy</label>
  <input type="checkbox" id="challenge-3"
value="WaterIrritation" />
  <label for="challenge-3">Water Causes Irritation</label>
  <input type="checkbox" id="challenge-4"
value="Maintenance" />
  <label for="challenge-4">General Maintenance
Required</label>
</form>
```

1



3

For Your Reference…

To ***create checkboxes***:

1. Add the ***<input>*** tag
2. Add the attribute ***type="checkbox"***
3. Add the ***id*** attribute
4. Add the ***name*** attribute
5. Add the attribute ***checked*** if this checkbox should be checked by default

Handy to Know…

- When ***value*** is used with checkboxes, it is used to specify the value sent to the web server when the element is selected.

CREATING A DROP-DOWN LIST

A ***drop-down list*** is a list of items that appears when you click on a drop arrow. The advantages of using a drop-down list are that the list presents a series of mutually exclusive options for the user to select from and it doesn't take up a large amount of space on the form to do it. A drop-down list is created using the ***<select>*** and ***<option>*** tags.

Try This Yourself:

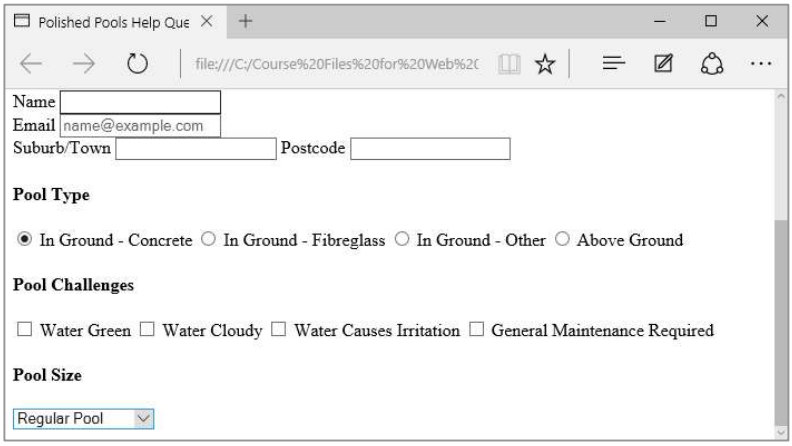
Same File

Continue using the previous file with this exercise or open the file Forms_6.html in the text editor and browser...

1. In your text editor, add the lines of code as shown to create the drop-down list
2. Save the changes to the file
3. Switch to your web browser and refresh the page to display the list
4. Click on the drop arrow to display the list of options

```
    <h4>Pool Size</h4>
    <select name="PoolSize">
      <option value="Regular Spa">Regular Spa</option>
      <option value="Large Spa">Large Spa</option>
      <option value="Small Pool">Small Pool</option>
      <option value="Regular Pool" selected>Regular
Pool</option>
      <option value="Large Pool">Large Pool</option>
      <option value="Commercial Pool">Commercial
Pool</option>
    </select>
```

1



3

For Your Reference…

To ***create*** a ***drop-down list***:

1. Add the opening ***<select>*** tag
2. Add the ***name*** attribute
3. Add an ***<option>*** tag for each option
4. Add the attribute ***selected*** if this option should be selected by default
5. Add the closing ***<select>*** tag

Handy to Know…

- <option> can be used between the <select> tags only.

CREATING A SCROLLABLE LIST

A ***scrollable list*** is very similar to a drop-down list except that more than one option is visible at a time. The number of options that are visible depends on the value assigned to the ***size*** attribute of the <select> tag. The advantage of using a scrollable list is that you can see more of the options, and can use the ***multiple*** attribute to enable the selection of several items at once.

Try This Yourself:

Same File

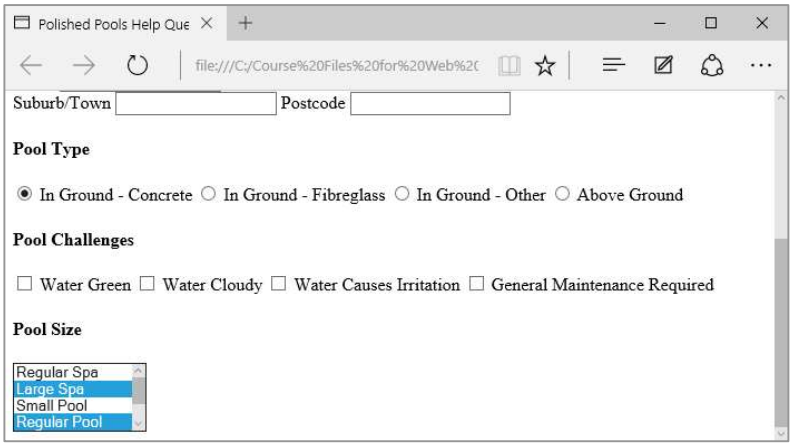
Continue using the previous file with this exercise or open the file Forms_7.html in the text editor and browser...

1. In your text editor, add the additional attributes to change the single choice drop-down list into a multiple choice scrollable list
2. Save the changes to the file
3. Switch to your web browser and refresh the page to redisplay the list
4. Scroll down the list to see all of the options, then click on ***Large Spa***
5. Hold down Ctrl and click on ***Regular Pool*** so that both appear selected

If you were to submit this form, both of these selections would be sent to the web server

```
  <h4>Pool Size</h4>
  <select name="PoolSize" multiple size="4">
    <option value="Regular Spa">Regular Spa</option>
    <option value="Large Spa">Large Spa</option>
    <option value="Small Pool">Small Pool</option>
    <option value="Regular Pool" selected>Regular
Pool</option>
    <option value="Large Pool">Large Pool</option>
    <option value="Commercial Pool">Commercial Pool</option>
  </select>
</form>
```

1



5

For Your Reference…

To ***create*** a ***scrollable list***:

1. Add the opening ***<select>*** tag
2. Add the ***name, multiple*** and ***size*** attributes
3. Add an ***<option>*** tag for each option
4. Add the attribute ***selected*** if this option should be selected by default
5. Add the closing ***<select>*** tag

Handy to Know…

- You can select a series of consecutive items by clicking on the first item, then holding down Shift and clicking on the last item.

CREATING A TEXT AREA

A ***text area*** is a multiple line text box designed to enable users to provide detailed information as requested in a form. It is defined using the ***<textarea>*** tag. The size of the text box on the screen can be set with the attributes ***cols***, which is the width of the box in characters, and ***rows***, which is the number of rows. The text will wrap automatically.

Try This Yourself:

Same File

Continue using the previous file with this exercise or open the file Forms_8.html...

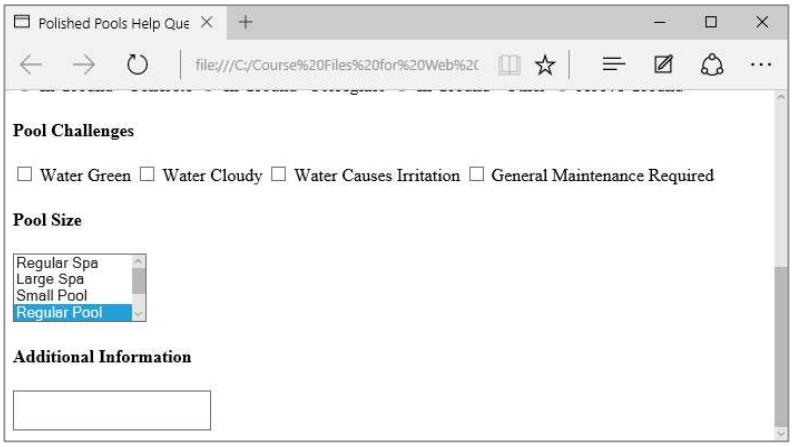
1. In your text editor, add the lines of code as shown to create the comment text box

   <textarea> is used instead of <input> when more than one line of text is required…

2. Save the changes to the file
3. Switch to your web browser and refresh the page to display the text area

```
  <h4>Pool Size</h4>
  <select name="PoolSize">
    <option value="Regular Spa">Regular Spa</option>
    <option value="Large Spa">Large Spa</option>
    <option value="Small Pool">Small Pool</option>
    <option value="Regular Pool" selected>Regular
Pool</option>
    <option value="Large Pool">Large Pool</option>
    <option value="Commercial Pool">Commercial Pool</option>
  </select>
  <h4>Additional Information</h4>
  <textarea name="Comments"></textarea>
</form>
```

1



3

For Your Reference…

To ***create*** a ***text area***:

1. Add the opening ***<textarea>*** tag
2. Add the ***name*** attribute
3. Add the closing ***</textarea>*** tag

Handy to Know…

- If you want to include some default text in the textbox, place the text between the <textarea> tags. You cannot format this text in any way – the tags you use will be displayed as well as the text.
- The <textarea> tag does not self-close – it needs a closing tag

CREATING RESET AND SUBMIT BUTTONS

A ***reset*** button is a button that, when clicked, will cause the fields in a form to be reset to their default values. A ***submit*** button, when clicked, will send the contents of the form to a specified program on a server or to an email address. Both of these buttons are specified using the ***<input>*** tag. The buttons are formatted automatically in your browser.

Try This Yourself:

Same File

Continue using the previous file with this exercise or open the file Forms_9.html in the text editor and browser...

1. In your text editor, add the lines of code as shown to create the ***reset*** and ***submit*** buttons

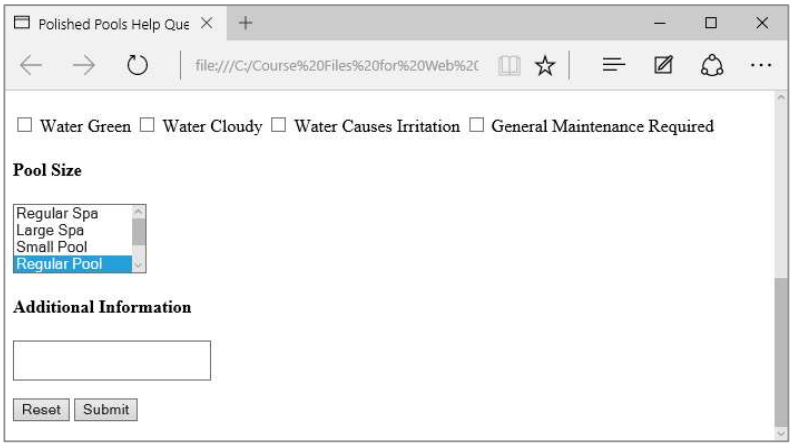
   The submit button sends the form to the location specified in the action attribute of the <form> tag…

2. Save the changes to the file
3. Switch to your web browser, refresh the page and then scroll down to see the buttons
4. Fill in some information on the form and make some selections, then click on **[Submit]**

   Your browser should take you to the Thanks.html page. You should be able to see the form information in the address bar as part of the information being passed to the server

```
  <h4>Pool Size</h4>
  <select name="PoolSize" multiple size="4">
    <option value="Regular Spa">Regular Spa</option>
    <option value="Large Spa">Large Spa</option>
    <option value="Small Pool">Small Pool</option>
    <option value="Regular Pool" selected>Regular
Pool</option>
    <option value="Large Pool">Large Pool</option>
    <option value="Commercial Pool">Commercial Pool</option>
  </select>
  <h4>Additional Information</h4>
  <textarea name="Comments"></textarea>
  <p>
    <input type="reset" value="Reset" />
    <input type="submit" value="Submit" />
  </p>
</form>
```

1



3

For Your Reference…

To ***create*** a ***reset button***:

- Add ***<input type="reset" value="Reset" />***

To ***create*** a ***submit button***:

- Add ***<input type="submit" value="Submit" />***

Handy to Know…

- The ***action*** attribute in <form> may also be accompanied by the attribute setting ***method="post"***. This provides a more secure data transmission as the information is not passed in the URL and visible in the address bar.

FORM VALIDATION

In previous versions of HTML, making input fields required and restricting the type of information they accept required writing code. HTML5 includes new attributes that you can use to validate your fields, making this a much easier task for you and a more consistent experience for web surfers. However, at the time of writing they are not supported in all browsers, so use them with care!

Try This Yourself:

Same File

Continue using the previous file with this exercise or open the file Forms_10.html in the text editor and browser...

1. In your text editor, add the ***required*** attribute to the name and email input fields

   We can't respond to enquiries if we don't know who they are and how to contact them!

```
<form action="Thanks.html">
  <h4>Name and Address Details</h4>
  <label for="name-input">Name</label>
  <input type="text" id="name-input" name="Name" autofocus
required />
  <br />
  <label for="email-input">Email</label>
  <input type="text" id="email-input" name="Email"
placeholder="name@example.com" required />
  <br />
```

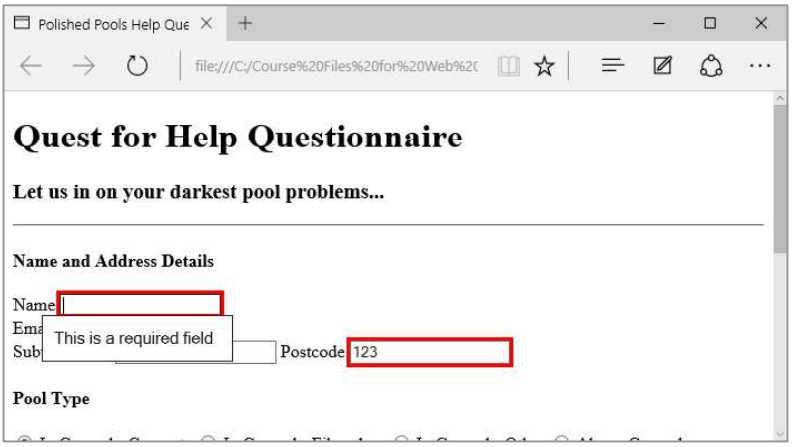
2. Add the ***pattern*** attribute to the post code input as shown

   *The value of the pattern attribute is a **regular expression**, which is a way of specifying the required format. This expression states that the post code must consist of four numbers. Regular expressions are outside the scope of this course and can get very complicated…*

```
  <label for="postcode-input">Postcode</label>
  <input type="text" id="postcode-input" name="Postcode"
pattern="\d{4}" />
  <br />
```

3. Save the changes to the file

4. Switch to your web browser, refresh the page and then press the **[Submit]** button without entering any values to see validation in action



For Your Reference…

To ***make*** an ***input field required***:

- Add the required attribute to the field

To ***make*** an ***input*** field's ***value match*** a ***pattern***:

- Add the ***pattern*** attribute to the field with a regular expression describing the pattern that the input value must match

Handy to Know…

- You should always validate the user's input on the server as well as in the browser, as there are many ways that malicious users can circumvent your form inputs and send the information directly to the server.

MORE FORM CONTROLS

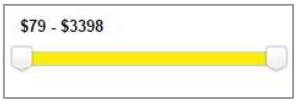
In addition to the standard form controls of textboxes, radio buttons, checkboxes and lists, HTML5 includes a number of form controls intended to make it easier to collect information such as passwords and email addresses as well as making data input much easier for mobile phone users. Some of these controls are described below.

Input Types

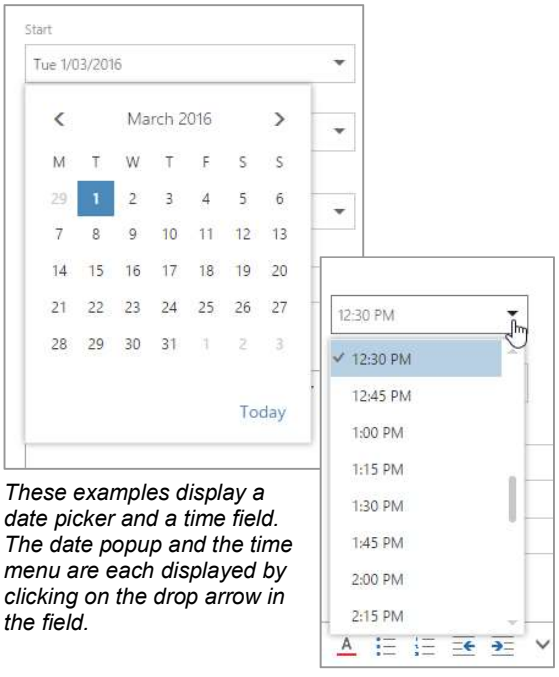
password	Password inputs display a textbox that hides the password as it's typed in. This might be implemented by replacing characters with stars or black circles
email	Email inputs display a textbox but on smartphones the keyboard that is shown will be tailored for emails, usually with an "@" key and ".com" key
number	Number inputs restrict input to numbers. On desktop browsers these generally appear as a box with up and down arrows while on smartphones the keyboard that is shown will only contain numbers
range	Ranges display a slider that the user can drag to select a value
date	Date inputs restrict input to dates and, depending on the browser, may include a popup to select the date from
time	Time inputs restrict input to times and, depending on the browser, may include a popup to select the time from
datetime	Date/time inputs function as a combination of date inputs and time inputs
tel	Telephone inputs restrict input to phone numbers and will trigger some smartphones to show a special keyboard for entering phone numbers
color	Colour inputs restrict input to colours and, depending on the browser, may include a popup to select the colour from
url	URL inputs will trigger some smartphones to show a special keyboard for entering URLs, usually with a ".com" key

Some of these input types may not be implemented on all browsers. You can still use them, however, as in browsers where they are not implemented they will just behave as input textboxes.

The above example displays an email text field and a password field. When the fields are filled in, the characters in the password are displayed as dots in order to maintain security.



The above example displays a range slider. These are common on retail websites as they allow you to filter products by price.



These examples display a date picker and a time field. The date popup and the time menu are each displayed by clicking on the drop arrow in the field.