INTRODUCING JAVASCRIPT

INFOCUS

JavaScript is used to program your HTML pages to add interactivity like expanding sections, popups, and so on. Alongside HTML and CSS, it is one of the three essential technologies of Web development.

The syntax, structure and use of JavaScript is an enormous topic, but the first step is in learning how to incorporate it into your Web pages.

In this session you will:

- ✓ gain an understanding of ***JavaScript***
- ✓ gain an understanding of ***JavaScript*** events
- ✓ learn how to create a ***JavaScript*** event
- ✓ learn how to manipulate an element using ***JavaScript***
- ✓ gain an understanding of ***JavaScript*** syntax
- ✓ learn how to create an embedded style
- ✓ learn how to create and apply an external script file.

UNDERSTANDING JAVASCRIPT

Alongside HTML and CSS, JavaScript is one of the three essential technologies of Web development. Where HTML provides the structure of the document and CSS provides the presentation rules, JavaScript is used to provide interactivity, changing your pages from static pages that load once to dynamic pages that respond to the user's actions.

A Programming Language

JavaScript is a programming language, and has support for many powerful features such as variables (for storing information), functions (for encapsulating functionality to be used repeatedly), objects (for grouping variables and functions) and arrays (for making lists of variables).

Back in the early days of the web, JavaScript was clunky and slow, but due to its expanding ubiquity and some powerful backers like Google and Mozilla, it has become powerful and extremely fast. Now JavaScript is responsible for sophisticated, rich web applications, taking us from the initial web of static pages to today's web of interactive web applications like Gmail and Facebook.

In the context of Web development, JavaScript is usually used as a client-side technology, meaning that it is downloaded and executed by the user's browser. This means that it is executed straight away, rather than having to wait for a response from the Web server.

Uses For JavaScript

Among other things, JavaScript can be used:

- To change the content of an HTML element, such as a heading or paragraph
- To change the attribute of an HTML element, such as changing its colour
- To popup images and videos over a web page
- To validate the values provided by a user in a form
- To retrieve information from the Web server in the background without having to interrupt the user, such as what happens when Gmail requests new email and shows an updated mail count
- To create sophisticated animations
- To create more powerful interactions for the user, such as dragging and dropping elements.

JavaScript Events

JavaScript can be run in response to the user interacting with a HTML element in some way, such as clicking on a button or selecting an option from a drop-down list, or in response to the browser completing an operation, such as loading the page. These instances are called ***events***.

Element Events

Some of the events you are likely to encounter with elements are:

onclick	Occurs when the user clicks an element
onmouseover	Occurs when the user moves over the element with their mouse
onmouseenter and ***onmouseleave***	Occurs when the user moves their mouse over or off the element
onkeypress	Occurs when the user press a key on their keyboard

Page Events

Events also occur in response to the page being changed:

onload	Occurs when the page has been loaded
onresize	Occurs when the browser window has been resized
onscroll	Occurs when the page is scrolled with the scrollbar or mousewheel

Form Events

There are also some handy events to deal with form elements:

onchange	Occurs when the value of a textbox, the checked state of a checkbox or the selected value of a list is changed
onfocus	Occurs when the element is focused, generally by the user clicking in it or using the Tab key to move to it
onblur	Occurs when the element loses focus

Responding To An Event

Responding to an event is similar to adding an attribute to an element. For example:

```
<input type="text" onchange="updateText()" />
```

In this example, we are adding a JavaScript event to listen for the ***onchange*** event of the text input (e.g. when the user types some text into it) and running the JavaScript function ***updateText*** (this part of the event declaration is called the ***event handler***). You can enter any valid JavaScript in the event handler.

CREATING A JAVASCRIPT EVENT

You can add JavaScript events to execute code when a button is clicked or when a textbox is typed in, and you can also add JavaScript events to execute code when the state of the document changes. Adding an event to the document loaded event means you can run code to set up the page for the user. In this exercise we'll just show a message to the user.

Try This Yourself:

Open File

Before starting this exercise you MUST open the file Introducing JavaScript_1.html...

1. In your text editor, add the ***onload*** event to the ***body*** element as shown

   "alert" is a built-in JavaScript function used to show a message to the user. JavaScript is case-sensitive so make sure you use upper and lower case exactly as shown…

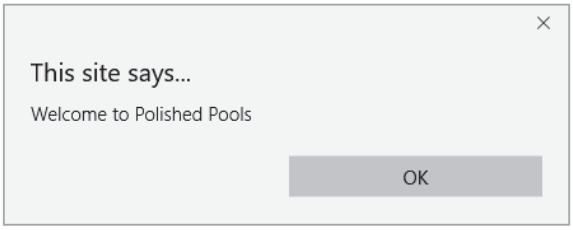
2. Save the changes to the file
3. Open the file in your web browser to display the page

   A message box should pop up. Press [OK] to dismiss it...

4. Click ***Refresh*** to reload the page and see the message again

```
<!-- Created for Polished Pools by PP Web Design -->
<!DOCTYPE html>
<html>
<head>
  <title>Polished Pools Valet Service</title>
  <link rel="stylesheet" type="text/css"
href="IntroducingJavaScript.css">
</head>
<body onload="alert('Welcome to Polished Pools')">
  <div id="container">
    <a name="top">
      <p style="text-align: center">
        <img src="PolishedPoolsLogo.png" alt="Polished Pools
logo" />
      </p>
    </a>
```

1



3

For Your Reference…

To ***create*** a ***JavaScript event***:

1. Add the desired event (e.g. ***onload***) to the element (e.g. ***body***)
2. Add JavaScript code to the event

Handy to Know…

- There are many built-in JavaScript functions that you will learn to use while building websites. Some useful ones are ***alert*** (for showing a message), ***confirm*** (to get a confirmation from the user), ***prompt*** (to ask the user to enter text) and ***setTimeout*** (to take some action after a specified time).

MANIPULATING AN ELEMENT

One of the main uses of JavaScript is to manipulate elements. For example, you can change the element's position, CSS class or text. Before you can manipulate an element you must first get a reference to it. In the exercise below we will add a message below the opening hours table.

Try This Yourself:

Same File

Continue using the previous file with this exercise or open the file Introducing JavaScript_2.html...

1. In your text editor, add a paragraph underneath the opening hours table, as shown

```
      <tr>
        <td>Saturday</td>
        <td>Closed</td>
      </tr>
      <tr>
        <td>Sunday</td>
        <td>Closed</td>
      </tr>
    </table>
    <div id="message"></div>
  </div>
```

2. Change the code in the ***onload*** event as shown

```
<body onload="document.getElementById('message').innerHTML =
'Welcome to Polished Pools'">
  <div id="container">
    <a name="top">
      <p style="text-align: center">
        <img src="PolishedPoolsLogo.png" alt="Polished Pools
logo" />
      </p>
    </a>
```

document.getElementById finds the element with the supplied ID, and innerHTML sets the text of the element…

3. Save the changes to the file

4. Open the file in your web browser to display the page

The paragraph that you added underneath the opening hours table should now have a message in it

For Your Reference…

To ***manipulate an element*** with ***JavaScript***:

1. Get a reference to the element by using ***document.getElementByID***
2. Set the desired element properties

Handy to Know…

- To set an element's text, use ***innerHTML***. To set an element's class use ***className***. To set an element's position use ***style.left***, ***style.top***, ***style.right*** or ***style.bottom***. For any of the position changes to have an effect the element must have a relative, absolute or fixed position.

JAVASCRIPT SYNTAX

JavaScript has a powerful syntax and some familiarity with it is very handy for web development. The below introduction is just a very brief overview, so don't worry if you don't grasp it all straight away. In a nutshell, JavaScript is case-sensitive, has ***variables*** that can be declared and have values assigned, and has ***functions*** for code re-use.

JavaScript Syntax

JavaScript is case-sensitive, which means that you have to be very careful to make sure that you don't mix up lower-case and upper-case keywords. Each line in JavaScript is ended with a semi-colon and comments are created by starting a line with two forward slashes.

Declaring And Assigning Variables

Variables are defined in JavaScript by use of the ***var*** keyword. The following example declares a number with an initial value of 1:

```
var number = 1;
```

And the following example declares a message with the value of "Welcome!":

```
var message = "Welcome!";
```

After variables are declared you can assign different values to them:

```
number = 2;
number = 3;
message = "We hope you are enjoying your stay!";
```

Declaring Functions

JavaScript code can be bundled up into functions, which can be re-used from different locations. Functions are declared with the ***function*** keyword, followed by parentheses. The body of the function is enclosed in braces:

```
function showWarning() {
  // This is the body of the function
}
```

CREATING AN INTERNAL SCRIPT

Internal scripts are scripts that are defined at the beginning of a document within ***<script>*** tags and are useful if you are unlikely to need to use the scripts for another document. In the exercise below we will create a script within our html document.

Try This Yourself:

Same File

Continue using the previous file with this exercise or open the file Introducing JavaScript_3.html in the text editor and browser...

1. In your text editor, add the code as shown between the ***<head>*** tags to create a function called *addMessage*

   We have changed the message below the opening hours to show the current time…

2. In your text editor, change the body tag to call the function we just created
3. Save the changes to the file
4. Switch to your Web browser, then click on ***Refresh*** to redisplay the page

   If you don't get quite the result you expect, check the casing and semi-colons. It is very easy to accidentally change case or leave out a semi-colon

1
```
<!-- Created for Polished Pools by PP Web Design -->
<!DOCTYPE html>
<html>
<head>
  <title>Polished Pools Valet Service</title>
  <link rel="stylesheet" type="text/css"
href="IntroducingJavaScript.css">
  <script>
    function addMessage() {
      var currentTime = new Date();
      var hours = currentTime.getHours();
      var minutes = currentTime.getMinutes();
      document.getElementById('message').innerHTML = 'It is
currently ' + hours + ':' + minutes;
    }
  </script>
</head>
```

2
```
<body onload="addMessage()">
  <div id="container">
    <a name="top">
      <p style="text-align: center">
        <img src="PolishedPoolsLogo.png" alt="Polished Pools
logo" />
      </p>
    </a>
```

3

Our Opening Hours

Day	Open
Monday	9:00 AM - 5:00 PM
Tuesday	9:00 AM - 5:00 PM
Wednesday	9:00 AM - 5:00 PM
Thursday	9:00 AM - 8:00 PM
Friday	9:00 AM - 9:30 PM
Saturday	Closed
Sunday	Closed

It is currently 14:55

Polished Pools offers pool products, equipment installation, maintenance and repairs, chemicals delivery and expert servicing designed to suit your individual needs and budget.

Valet Service

Our valet service means that we provide personal, individual attention to you and your pool or spa. Our ***Polished Pools*** on site, discuss your requirements, and find the best solution for you minimum cost to you, drawing on our ***10 Points To Perfection Plan***. to establish a regular pool maintenance schedule allowing us to take

For Your Reference…

To ***create*** an ***internal script***:

- Add the opening ***<script>*** tag between the <head> tags
- Add the JavaScript code that you require
- Add the closing ***</script>*** tag

Handy to Know…

- You can use JavaScript to display the time in a different format (e.g. using AM and PM) or include the day of the week or many other things. That's a topic for another day, though.

CREATING AN EXTERNAL SCRIPT FILE

If you have a web site with a series of pages, each of which has shared functionality, it makes sense to place the scripts in a single external file and link that to the individual pages. Any changes that are required can be made once in the external file. An external script file can be created directly from an internal script.

Try This Yourself:

Same File

Continue using the previous file with this exercise or open the file Introducing JavaScript_4.html in the text editor and browser...

1. In your text editor, select the text between the ***<script>*** tags (but not the tags themselves)
2. Copy the code to the clipboard
3. Create a new text document and paste the code into it
4. Save the document as **MyScriptFile.js** in the ***Course Files for Web Development*** folder and close the document
6. In your text editor, modify the code as shown, replacing the <script> tags and code with a link to the new file
7. Save the changes to the file
8. Switch to your Web browser, then click on ***Refresh*** to redisplay the page

   It should look unchanged, even though the code is now in a separate file

```
<!-- Created for Polished Pools by PP Web Design -->
<!DOCTYPE html>
<html>
<head>
  <title>Polished Pools Valet Service</title>
  <link rel="stylesheet" type="text/css"
href="IntroducingJavaScript.css">
  <script>
    function addMessage() {
      var currentTime = new Date();
      var hours = currentTime.getHours();
      var minutes = currentTime.getMinutes();
      document.getElementById('message').innerHTML = 'It is
currently ' + hours + ':' + minutes;
    }
  </script>
</head>
```

1

```
<!-- Created for Polished Pools by PP Web Design -->
<!DOCTYPE html><html>
<head>
  <title>Polished Pools Valet Service</title>
  <script src="MyScriptFile.js"></script>
</head>
<body onload="addMessage()">
```

6

For Your Reference…

To ***create*** and ***apply*** an ***external script file***:

1. Save your code in a file with the extension .js
2. Add a ***<script>*** tag in your HTML file
3. Add the ***src*** attribute with the name of the file

Handy to Know…

- The script tag needs a closing tag, even when it's empty and pointing to an external script file.