Javascript

Client-side scripting

The **client** is the computer system (including tablets and mobile devices) which is running the web browser.

Client-side scripting can be used to **make web pages change** after they arrive at the browser.

Client side scripts can also be used to **perform validation of data entered into forms**, either using **JavaScript** or some new features of **HTML5**.

Client-side scripts are:

- **interpreted** by the **browser**
- **executed** on the **client system**

- It is possible to generate a query client-side.
- It is **not** possible to execute the query on a client.
- Some tasks will always require server-side scripting.

Client-side scripting using JavaScript

JavaScript is the main client-side scripting language that can be used to create, delete and manipulate HTML elements.

Extensive JavaScript code is often contained within its own file and linked to a webpage in the same way that a CSS file can be linked.

Some common tasks that can be performed using JavaScript relate to how the webpage responds to action taken by the user using their mouse or touchscreen.

These are called **mouse events**.

JavaScript mouse events

Using JavaScript, functions are created that will **change what appears on screen** depending upon **what the user does with the mouse**.

There are a number of mouse events that are used in HTML code to trigger the execution of JavaScript functions.

- onmouseout - occurs when the **cursor is moved away from an element** such as a button or a heading.
- onmouseover - occurs when the **cursor is hovered over an element** such as a button or a heading.
- onmousedown - occurs when the user **presses a mouse button over an element**.

- onmouseup - occurs when the user **releases the mouse button over an element**.
- onclick - occurs when the user **clicks on an element** such as a button or hyperlink.
- onmousemove - occurs when the **cursor is moving over an element** such as an image or menu.

JavaScript mouse events example

This script has been written to create two functions, mouseOver() and mouseOut().

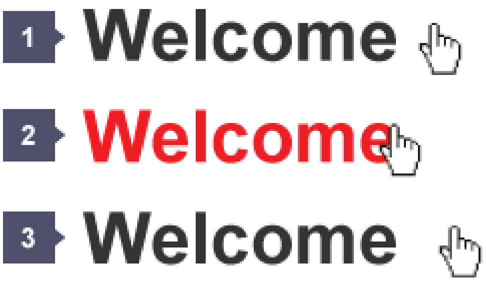
```
Line 1: <script>Line 2: function mouseOver() {Line 3:
document.getElementById("demo").style.color = "red";Line 4: }Line 5: function
mouseOut() {Line 6: document.getElementById("demo").style.color = "black";Line 7:
}Line 8: </script>
```

The <script> and </script> tags on lines 1 and 8 are used to let the browser know that anything contained between these tags is written in JavaScript.
Within the <script> and </script> tags it is possible to write JavaScript code or link to an external JavaScript file.
Lines 2-4 show the function called mouseOver(). This function will change the style of an element to red if the ID of that element is called 'demo'.

Lines 5-7 show the function called mouseOut(). This function will change the style of an element to black if the ID of that element is called 'demo'. For this code to run, it is necessary for there to be an HTML element with the ID 'demo'.

In the HTML shown below a heading is being used. This heading is identified as 'demo'.

```
<h1 id="demo" onmouseover="mouseOver()"
onmouseout="mouseOut()">Welcome</h1>
```



1. This heading will display the word ‘Welcome’ on screen.
2. The mouseOver() function is activated if an onmouseover event occurs. This means it will run if the mouse cursor is placed over the word welcome. The mouseOver() function is set to change the style to red.
3. If the user moves the mouse away from the word ‘Welcome’ then the text colour would change to black again because the mouseOut() function would be triggered by the onmouseout event. The onmouseout event activates the function for mouseOut() when the cursor is moved away from the word welcome.