

Appendix 17: JavaScript (WDD)

JavaScript events (onmouseover, onmouseout and onclick) are used to implement interactive web content.

Events are placed within HTML elements as shown below:

```
<img onmouseover=" " ">
```

The inverted commas contain the action to be executed when the event takes place. Possible actions include:

- ◆ hiding page elements
- ◆ revealing page elements
- ◆ changing the position of an element
- ◆ changing the size of an element
- ◆ changing the colour of an element
- ◆ changing the look of text

Actions can be executed by:

- ◆ referring to the element containing the JavaScript event
- ◆ referring to a different element, identified by an id
- ◆ calling a JavaScript function containing the actions

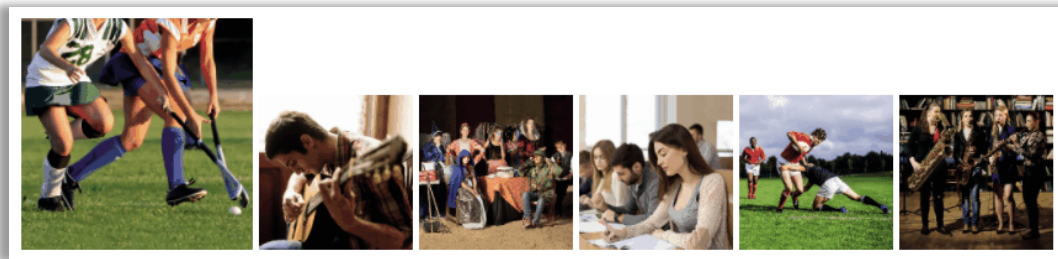
The following examples have been taken from the *Higher Computing Science example website*. The example website can be downloaded as a zip file from the [Higher Computing Science](#) page on SQA's website. To view the full code in context, open and view the source code from the pages noted in each example.

Interactively changing the size of a graphic (home page): example 1

Inline JavaScript can be used to increase and decrease the size of a graphic, as a mouse pointer passes over and out of a graphic.

```

```



This example contains two events, `onmouseover`, `onmouseout` used to execute four actions:

```
width='150px' height='150px' width='100px' height='100px'
```

Each action in a JavaScript statement is separated by a semi-colon.

Each action in the above events are prefixed by `this.style`.

`this` meaning this element. In example 1 this refers to the graphic (`img`) element containing the event.

`style` the style of this element will be altered using a CSS declaration

Interactively changing the size of a graphic (home page): example 2

JavaScript functions may be called to increase and decrease the size of a graphic, as a mouse pointer passes over and out of a graphic.

```

```

The JavaScript functions above are passed the parameter `this`, referring to the element containing the event `displayLarger(this)`

The JavaScript functions are placed within a `<script>` element in the `<head>` section of the HTML document:

```
<script>
function displayLarger(my_image)
{my_image.style.width='150px';
my_image.style.height='150px';}

function displaySmaller(my_image)
{my_image.style.width='100px';
my_image.style.height='100px';}
</script>
```

The parameter is used by each function to implement the same actions as example 1.

Implementing example 2 as a function, allows the same code to be called from multiple events:

```



```

Creating a rollover image (sports page): example 1

Rollover images can be created using two JavaScript events. The first event (`onmouseover`) displays an alternate graphic when the mouse passes over the image element. The second event (`onmouseout`) displays the original graphic when the mouse pointer leaves the image.

The following is the view in browser when the mouse pointer is not over any image:



The following is the view in browser when mouse passes over the right-hand image:



This can be implemented using the inline JavaScript shown below:

```

```

The action `src=' '` points the image element towards a given address.

Creating a rollover image (sports page): example 2

A rollover image can also be achieved using two functions. The first function is called using onmouseover. The second is called using onmouseout. Each function is passed the img element as a parameter using this. As before each function changes the src of the image:

HTML

```

```

JavaScript

```
<script>
function displaySport1A(my_image)
    {my_image.src = '../../../images/netball.jpg';}
function displaySport1B(my_image)
    {my_image.src = '../../../images/football.jpg';}
</script>
```

Highlighting text by dynamically changing its colour (home page):

Text may be highlighted by changing its style when the mouse pointer passes over the element containing the text.

The example below highlights a paragraph element dark red, by using an onmouseover event:

```
<p onmouseover="this.style.color='darkRed'">

Wednesday 12th June.<br> The Computing ... Room B9.
</p>
```

Highlighting text by dynamically changing its colour (study page):

To implement an action on a different page element, an id is required to identify the element the action will be performed on.

An element is identified using document.getElementById(' ')

The example below highlights the paragraph text when the mouse passes over the image contained in the paragraph:

```
<p id="highlight"><b>Thursday 13th June.</b><br> The guidance ...
suits them. </p>
```

Revealing an element using onclick (study quiz page):

When a web page loads, elements may be initially hidden by applying the CSS declaration `display:none`. This can be implemented using an inline or internal style, or by implementing an external style (using a class, as shown below):

CSS

```
.hidden{display:none}
```

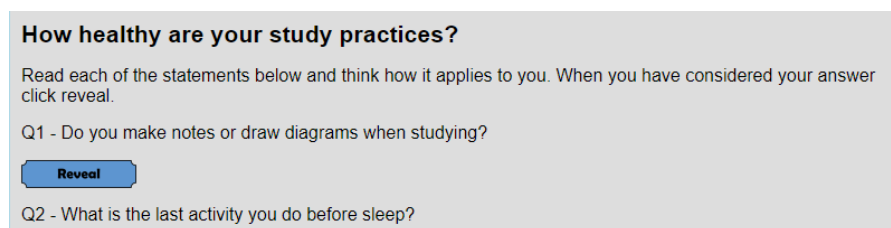
The graphic element in the example (reveal.png) contains an onclick event that uses the paragraph element's id to execute the action 'display=block' on the paragraph. The result of this action is that the paragraph becomes visible on the page.

HTML

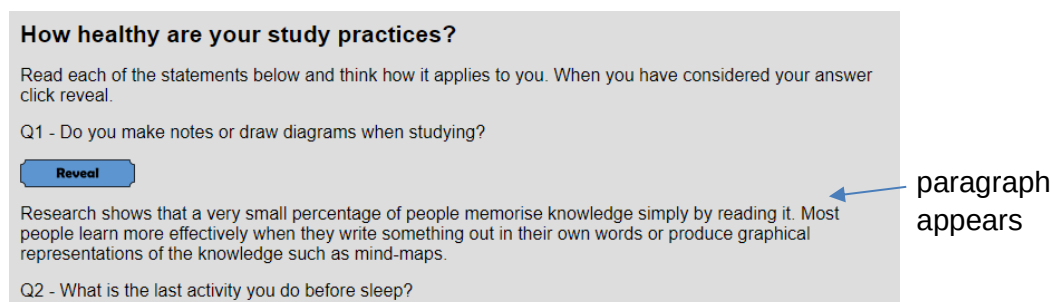
```
<p> Q1 - Do you make notes or draw diagrams when studying? </p>

<p id="reveall" class="hidden"> Research shows that a very small
percentage of people memorise knowledge simply by reading it. Most
people learn more effectively when they write something out in their
own words or produce graphical representations of the knowledge such
as mind-maps.</p>
```

The following is the view in browser when page loads:



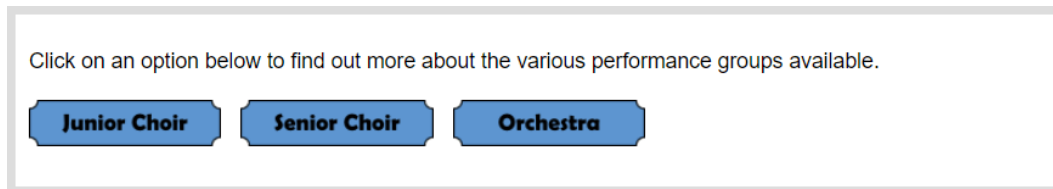
The following is the view in browser after 'Reveal' has been clicked:



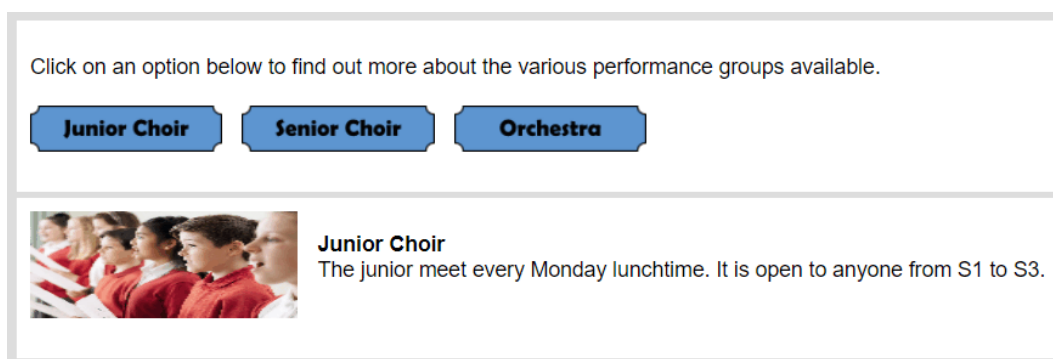
Revealing one of three hidden elements using onclick (music page):

Image elements, with associated onclick events, can be used to offer users a choice of what they view on a web page. Each onclick event is used to reveal a hidden <section> element containing a graphic and text relating to each option.

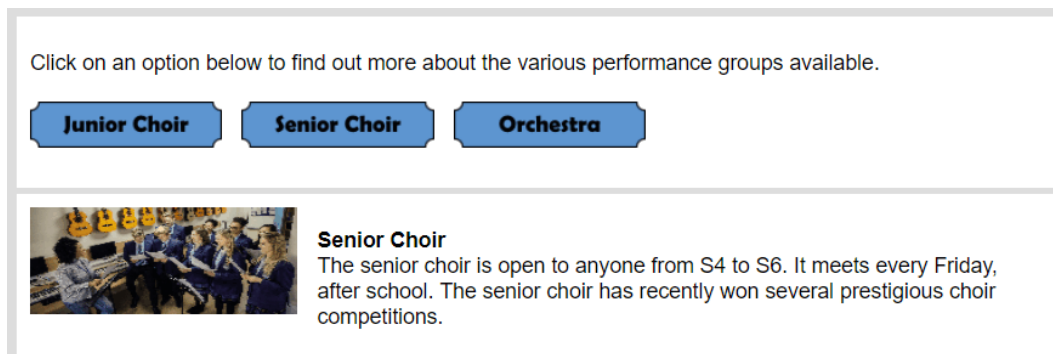
The following is the view in browser when page loads:



The following is the view in browser when the 'Junior Choir' graphic is then clicked.



The following is the view in browser when the 'Senior Choir' button is then clicked.



Each onclick event calls a function:

```
<p> Click on an option below to find out more about the
various performance groups available. <br><br>



</p>
```

The three section elements, containing the graphic and text to be revealed, are each identified by an `id` and initially hidden using `display:none`.

```
<section id="junior" style="display:none">

<p><b>Junior Choir</b><br>The junior meet every Monday
lunchtime. It is open to anyone from S1 to S3.</p>
</section>
<section id="senior" style="display:none">

<p><b>Senior Choir</b><br>The senior choir is open to anyone
from S4 to S6. It meets every Friday, after school. The
senior choir has recently won several prestigious choir
competitions.</p>
</section>
<section id="orchestra" style="display:none">

<p><b>Orchestra</b><br>The orchestra is not age limited,
instead being open to anyone who plays their instrument at
grade 3 level or higher.</p>
</section>
```

Each of the three JavaScript functions called by the onclick event reveals one of the three `<section>` elements and hides the other two using three `style.display` actions:

```
<script>
function displayJC() {
    document.getElementById("junior").style.display="block";
    document.getElementById("senior").style.display="none";
    document.getElementById("orchestra").style.display="none";
}

function displaySC() {
    document.getElementById("junior").style.display="none";
    document.getElementById("senior").style.display="block";
    document.getElementById("orchestra").style.display="none";
}

function displayO() {
    document.getElementById("junior").style.display="none";
    document.getElementById("senior").style.display="none";
    document.getElementById("orchestra").style.display="block";
}
</script>
```