

CSCU9B2 Practical 6: Multimedia and JavaScript

Aims:

- To do further JavaScript programming.
- To use JavaScript to control the display of media on a web page.

Please register your practical attendance: Go to the GROUPS\CSCU9B2 folder in your Computer folder and double-click on the Register icon. Ask a demonstrator if you need help or something goes wrong.

This sheet contains one checkpoint (see end of sheet).

This week's practical is about using the HTML5 multimedia tags, and using JavaScript to change the content of tags. Ideally you will need headphones to listen to sound clips, but you can complete the practical without them.

HTML5 <audio> and <video> tags

These tags are used to include audio and video multimedia entities on a web page. They are described at a very basic level in the HTML5 document on the web site (see the link on the <http://www.cs.stir.ac.uk/courses/CSCU9B2/practicals/> page).

A full list of the properties and methods of <audio> and <video> objects can be found at: http://www.w3schools.com/tags/ref_av_dom.asp

There are some simple examples of using multimedia audio and video tags at: <http://www.cs.stir.ac.uk/courses/CSCU9B2/resources/MediaExamples/>.

We will now modify these examples:

- Copy folder **MediaExamples** from **Wide\Groups\CSCU9B2** to your Web folder: this contains examples of using the <audio> and <video> tags, which you will modify.

Audio

- Go into your MediaExamples folder and open **testaudio.html** in IE and TextPad.
- In IE, play the sound track and click the button to find out information about it.
 - You will need your own headphones to listen to this in the lab. If not, then you can at least see its progress on the duration bar!
- Now examine the code in TextPad.

You should see the <audio> tag: it has an id, so that we can refer to it in JavaScript, and the `controls` attribute, so that the audio controls are visible. It contains multiple possible sources for the audio file to be played. The browser will load the first one it finds.

Make the following modifications (you will need to consult the W3 Schools URL given above to find appropriate methods and properties for the audio tag):

- Alter the **showinfo()** function so that instead of displaying all possible sources, it only displays the URL of the actual source being used (there is an audio property for this).
- Create your own buttons for play, pause and rewind (this last one is tricky!). You can achieve the correct effect either using a function or directly (see the code in the video example).
- Add code so that the paused time is displayed on the screen when the pause button is pressed (it might be convenient for pause to be implemented in an explicit function to be able to add this code).

Video

- Open your copy of **testvideo.html** in IE and TextPad.
- In IE, play the video and click the button to find out information about it.
 - You will need your own headphones to listen to this in the lab. If not, then you can still see the video!
- Now examine the code in TextPad.

You can see that controlling audio and video is very similar. We will make one change to this file:

- Add code so that an alert box pops up with a suitable message when the video is loaded and ready to play (this is a little tricky, but actually only requires a very small bit of code involving an event attribute for the video tag: see W3 Schools, as above).

Changing content using JavaScript

Replacing images

Remember the JavaScript examples from the previous practical? If you look at the JavaScript examples **6** and **6a** at <http://www.cs.stir.ac.uk/courses/CSCU9B2/resources/JSexamples/> you will see how an image can be replaced by another image when the mouse cursor is placed over the image, or removed from the image.

Of course, the actions taken for the `onmouseover` or `onmouseout` events can be completely different. So try this:

- Using the arrows in your copy of the **JSexamples** folder (from practical 5), put together a web page with a few arrows across the top (perhaps as a table).
- Make sure that each image has an `id`, so that you can refer to it in your JavaScript.
- Now make the `onmouseover` and `onmouseout` events for each arrow cause a change to a *different arrow* (e.g. make an arrow that was pointing upwards point downwards. Examples **6** and **6a** will give you clues as to the sort of code to use.)
- If you feel up to it (programming wise), change the arrows in a way that depends on where the arrows are currently pointing (see **Example 6a** above for some suggestions about how to do this). You can use a variable for each arrow to hold its current state.

Changing HTML element types

In addition to altering the image in a tag, JavaScript can also be used to alter the HTML itself. Indeed, there seems to be little limitation on how this can be applied. For example, using the `setAttribute` method, one can turn a text field into a button!

Suppose we define the input tag:

```
<input id="numberinput" value = "xyz">
```

A line like:

```
document.getElementById("numberinput").setAttribute("type", "button");
```

will turn this input field into a button with the same value, and

```
document.getElementById("numberinput").setAttribute("type", "field");
```

will turn it back to a field.

The `setAttribute` method takes two parameters: the name of the attribute, and the value it is to be set to. (This is a very powerful technique, but note that as well as allowing you to change existing attributes, it also allows you to create new ones. For example, if you mis-spelt `type` as, say `tyzpe`, the method would create a new attribute called `tyzpe`, and give it a value!)

- **Try it out:** create a small web page with an input field (with an id and containing some initial text), and two buttons, one to change the input field to a button, and one to change it back.

Changing element values

A rather more common requirement is to change the HTML inside some page element, for example, one might have:

```
<p id="message1"> Error message goes here </p>
```

One way to do this is to use the `innerHTML` property: this is, apparently, not a standard, but it appears to be supported on all the browsers I've tried. A method call like:

```
document.getElementById("message1").innerHTML = "Error in reading file" ;
```

can change the text appropriately.

- **Try it out:** add a similar example to this to the web page you created in the previous step.

CHECKPOINT [MEDIA]

Put together the code for the audio, video, the arrow changing, the tag type changing and the tag content changing into a single web page, and demonstrate this code to the tutor.