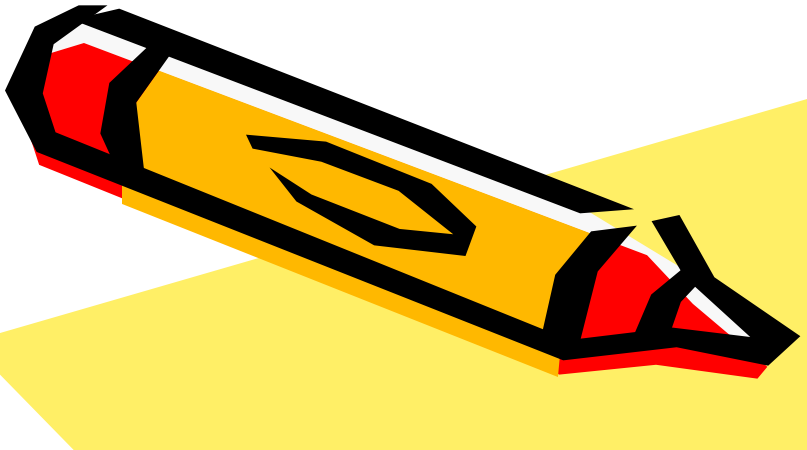
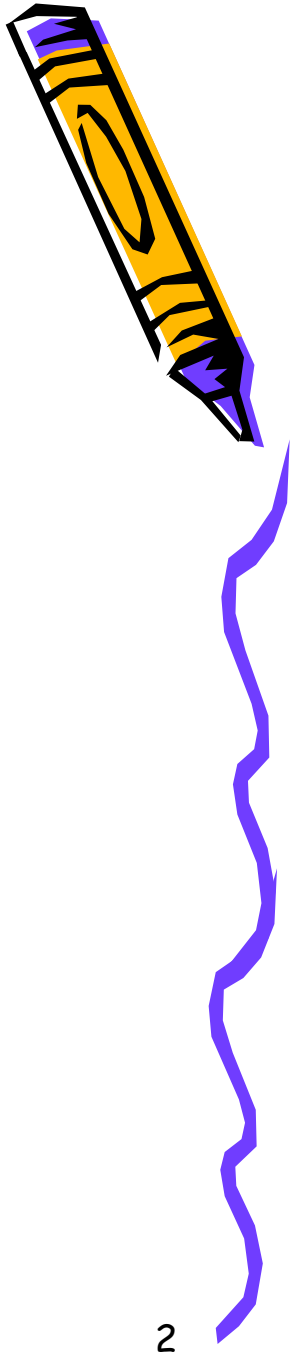


Programing for Digital Media
EE1707

Lecture 4
JavaScript

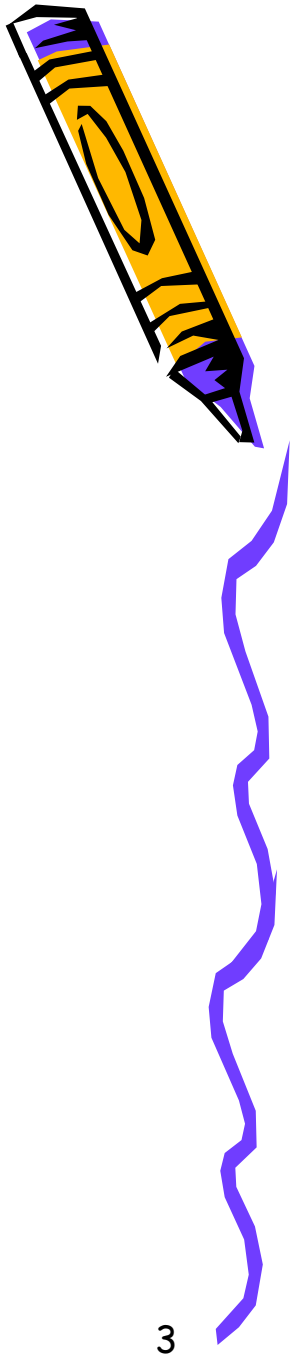
By: A. Mousavi & P. Broomhead
SERG, School of Engineering Design, Brunel University, UK

today



- Event Handling in JavaScript
- Client-Side JavaScript
- Examples

Events



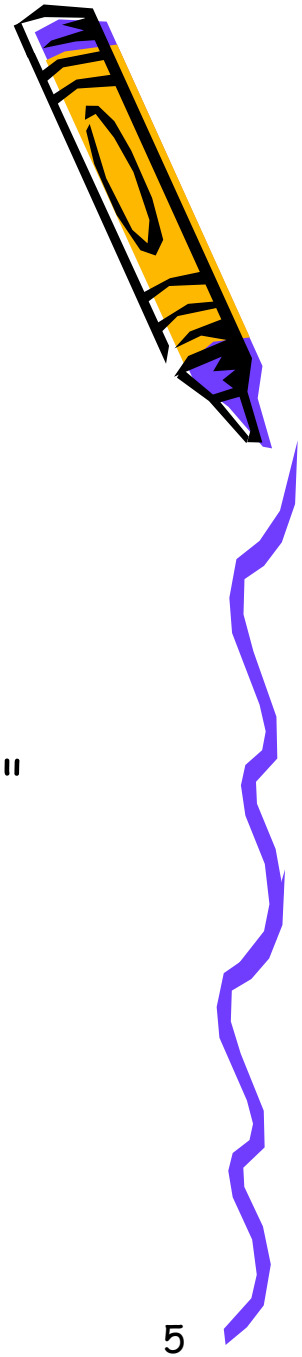
- **Events are user-driven implementation of functions or operations in an application. For example:**
  - Click on a button or mouse
  - Mouse going over an area of the document
  - Pressing on a button on the keyboard
  - …
- ***Events* trigger a predefined operation or function**

Events cont.

- **Browsers cater for the events that are handled by JavaScript**

```
<img src="MyImage.jpg" name="MouseOverTest"
      onMouseOver="document.images['MouseOverTest'].src='MyImag
      e.jpg'";
```

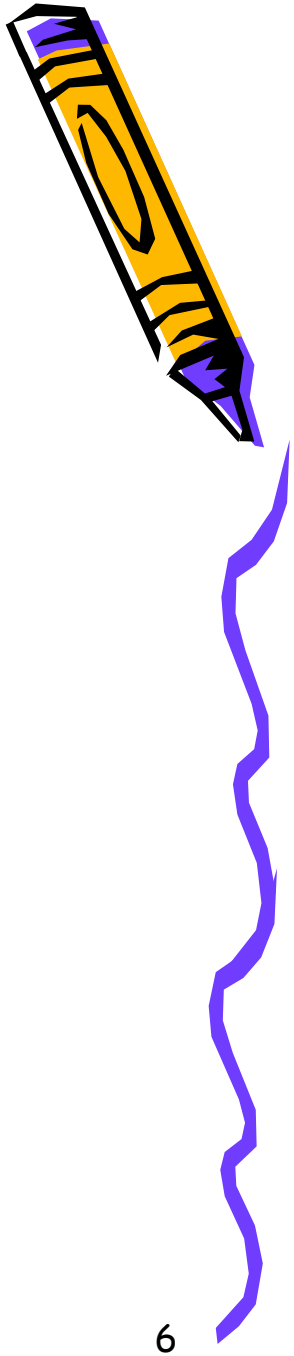
Example 4-1



OnMouseOver event:

```
<html>
<body>

<img src="Bluehills.jpg" name="Blue"
  onMouseOver="document.images['Blue'].src='Winter.jpg'"
  onMouseOut="document.images['Blue'].src='Blue_hills.jpg'"
width="300" height="100" />

</body>
</html>
```

Handling Images in a document



- **Images are one of the most important components of any document**
  - Rollover images for enhanced user experience
  - Animation using off-screen images

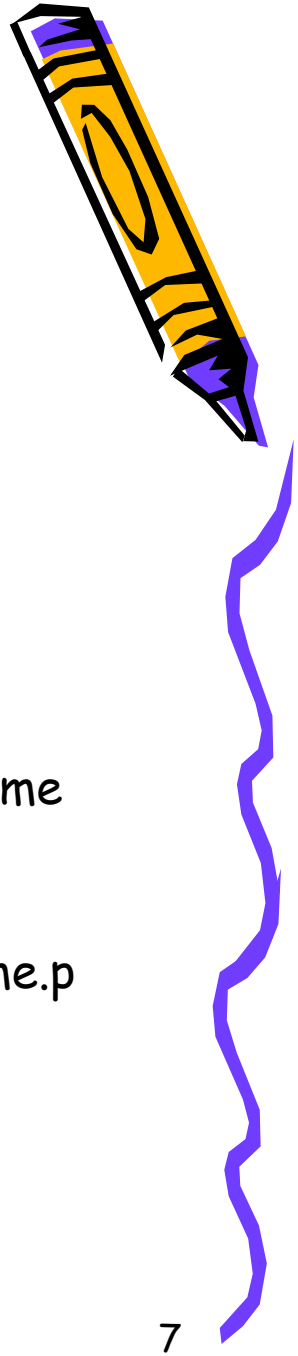
- **Good practices:**
  - Name images in the html document

    (**
  - Names can be assigned to a variable

    var imgName = "myPhoto";

    document.images[imgName];

Another Exercise



A rollover image with a link

```
<html>
<body>

<a href="mailto:info@holidays.com"

<img src="contactme.png" name="ContactButton"

    onMouseOver="document.images['ContactButton'].src='contactme
    2.png'"

    onMouseOut="document.images['ContactButton'].src='contactme.p
    ng'" width="100" height="50" />

</body>
</html>
```

Three Methods to access Images and Image Objects

1. ***images[ ]* array using a string index**

```
document.images["ImageName"].src = "ImageName.png";
<img src= "...                    />
```

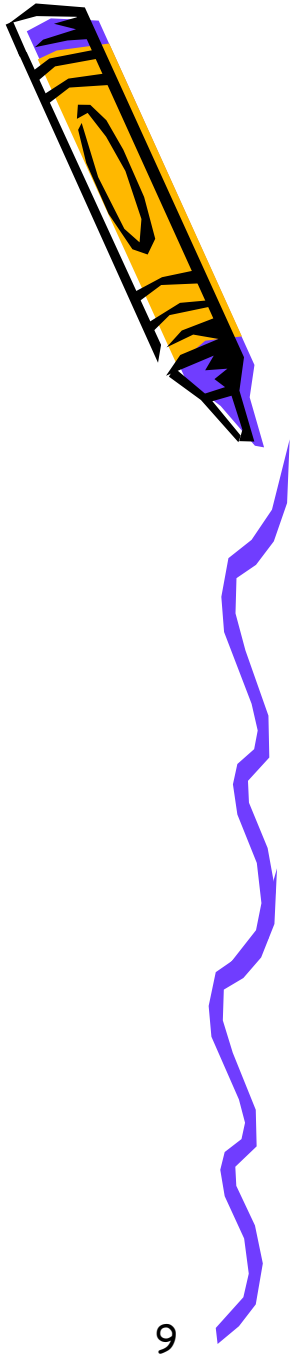
2. **Using the numeric index in the *images[ ]* array**

```
document.images[0].src= "ImageName1";
document.images[1].src = "ImageName2";
<img  src=...                  />
```

3. **Using the <img> name on the images[ ] array object**

```
document.images.ImageName.src = "ImageName.png";
```

Timer Event



Timer events are important features in JavaScript. You can design applications such as: animation, displayable clocks, rotating advertisements, …

setTimeout( ) function

- ***setTimeout ( )* is a method of the browser window object**

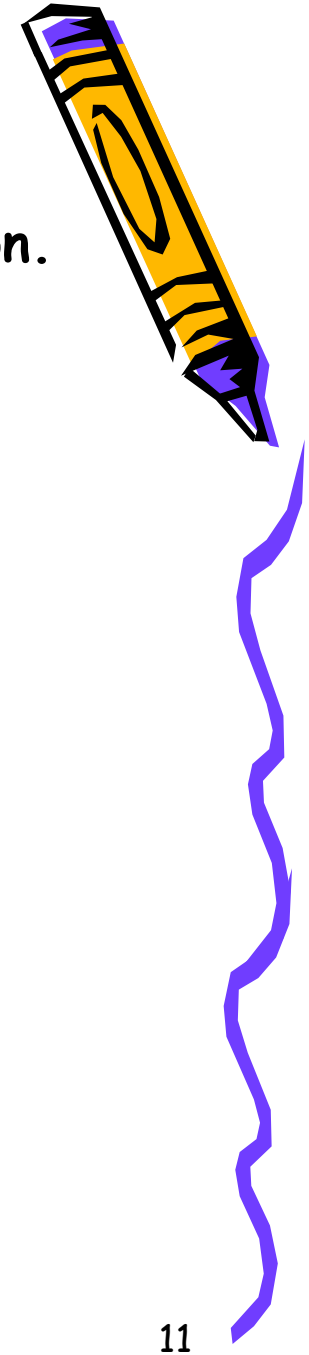
- **Two arguments are allowed:**
  - A string semicolon separated statements
  - The delay in milliseconds

  Example:

  // run a function called imageRotate with 0.1 second delay

  timer = setTimeout("imageRotate( );" , 100);

Examples

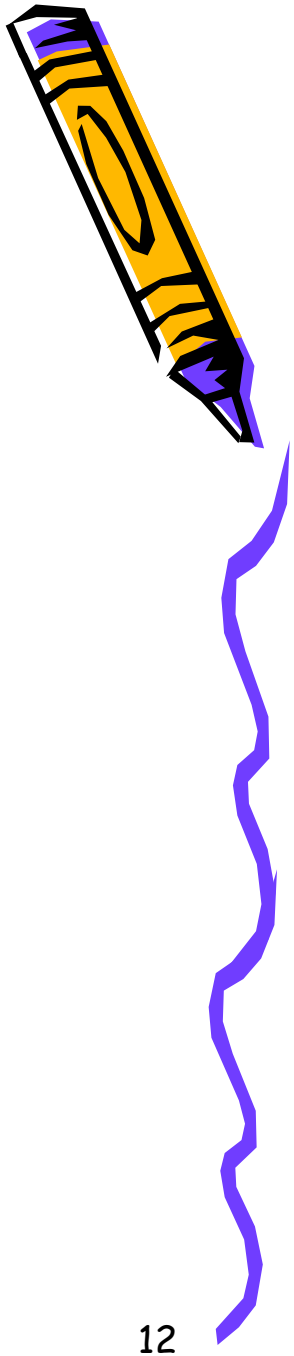


E4-3 (start and stop a timer) and E4-4 a small animation.

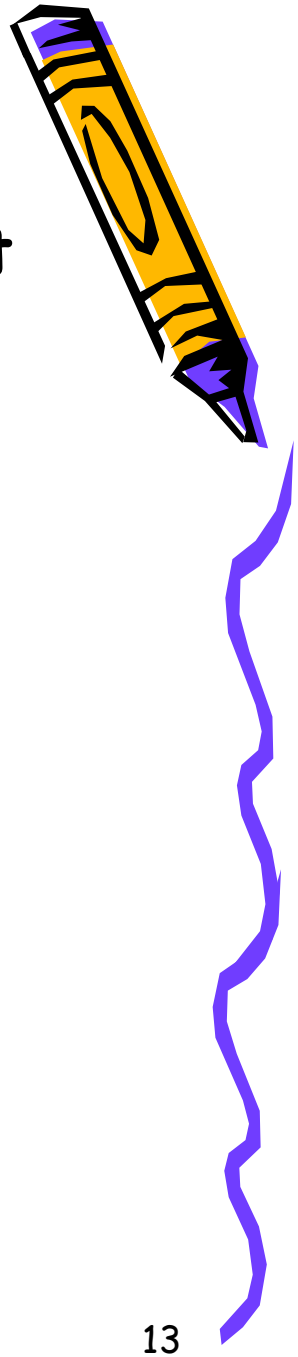
```
<html>
<head>
<script type="text/javascript">
function shakeleft()
{
document.getElementById('image').style.position="relative";
document.getElementById('image').style.left="3";
timer=setTimeout("shakeright()",100);
}

//continued on next slide
```

Example continued

```
function shakeright()
{
document.getElementById('image').style.left="0";
timer=setTimeout("shakeleft()",100);
}
function stoptimer()
{
clearTimeout(timer);
}
</script>
</head>
<body>
<b>Mouse over the heart to beat</b><br />
<img id="image" src="heart.png"
onmouseover="shakeleft()"
onmouseout="stoptimer()" width="50" height="50" />
</body>
</html>
```



Managing your JavaScript Applications



1. **Make sure that your web pages work without JavaScript (graceful degradation)**

2. **Separating structure from behaviour**

3. **Making sure that older versions of browsers handle your pages gracefully (backward compatibility)**

Also see DOM Scripting by J. Keith

Graceful Degradation

- **Modern Browser applications have the capability to block JavaScript (i.e. pop-up pages)**

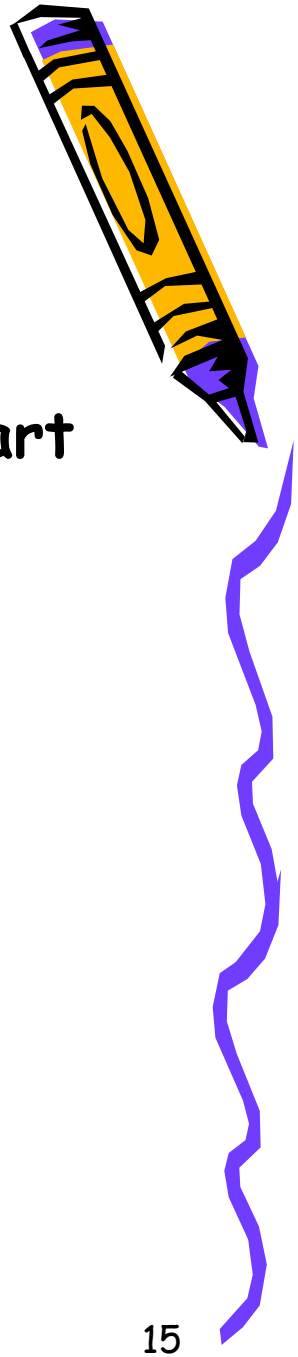
window.open(url, name, features)

Example

```
function popUp(winURL) {
        window.open(winURL, "popup", "width = 400, height=200");
```

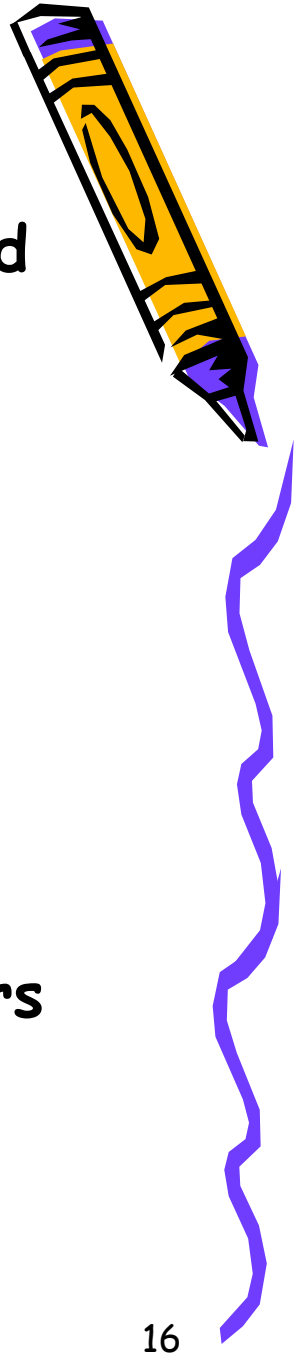
- **You have to make sure that if the JavaScript is blocked your web pages and web site could be navigable – Graceful Degradation**

Separating structure from behaviour



- Nowhere more applicable than CSS
- Keep styles in an external file rather than part of the document
- This separation also allows for graceful degradation

Backward compatibility



- **Quiz the browser if it support JavaScript and if so to what level**

```
// use a condition statement
window.onload = function ( ) {
    if ( !document.getElementByID)
        return false;
        }
```

- **Browser sniffing – trying to read the browsers properties. Not very reliable.**