

# JavaScript Workshop

---

Image Object

# Image object

- Each image on an .html page (`<img src = "image.gif">`), is accessible via objects in JavaScript.
- Each of these image are stored in the `images[]` array as an `img` object. This array is a property of the another object called `document`. A `document` object (discussed in a later session) basically refers to everything on the web page itself.
- The first image on the web page is referred to as `document.images[0]` or `document.images["name"]`. Where "name" is the name attribute you assign it in HTML.

# Creating image rollovers

- A "rollover" effect can be created using images and events. when the user moves their mouse over an image the image changes.
- This is accomplished by using the onMouseOver and onMouseOut event handlers (covered last session) and the image object src property.

```
<a href = "link.html" onMouseOver = "function()" onMouseOut =  
"function()" > <img src = "image.gif" border = "0" > </a >
```

- The onMouseOver and onMouseOut event handlers are used with the <a> as opposed to the <image>, to show the cursor change of a hyperlink.

# Variations of the image rollover

- The dilemma for this technique is that each image that needs a pair of functions for each "roll-over" effect.
- Techniques for referencing the image are:

*document.images[0] ---> index number is the placement order on page*

*document.images["image name"] ---> image name is the HTML name*

# Step 1: Creating image references

- Using the image object, you can create object references to actual image files (.jpg, .gif). With this object variable referencing an image file, it will also automatically preload the image into the web browser cache (where all web page images go). This is very important because the transfer from one image to the other (onMouseOver to onMouseOut) must seem instantaneous.

*Create an object variable to reference an image file:*

```
var mouse1_off = new Image();
```

```
mouse1_off.src = "mouse1_out.gif";
```

## Step 2: Using the eval() function

- After all the images are stored in object references (see Step 1). The next step is use the eval() function to dynamically reference the particular image that needs to be changed.
- The eval() function takes a string value and executes it as if it were an object reference.

```
function MouseOn(imgName) {
```

```
    document.images[imgName].src = eval(imageName + "_on.src");
```

```
}
```

## Step 3: Putting it all together

```
var mouse1_off = new Image(); mouse1_off.src = "mouse1_out.gif";  
var mouse1_on = new Image(); mouse1_on.src = "mouse1_over.gif";
```

```
function MouseOn(imageName) {  
    document.images[imageName].src = eval(imageName + "_on.src");  
}
```

```
function MouseOff(imageName) {  
    document.images[imageName].src = eval(imageName + "_off.src");  
}
```

```
<a href="javascript:void(0)" onMouseOver="MouseOn('mouse1');  
"onMouseOut="MouseOff('mouse1'); ">  </a>
```

- JavaScript Workshop website:  
<http://www.javascriptworkshop.com>
- Any questions?  
[frank@javascriptworkshop.com](mailto:frank@javascriptworkshop.com) - Frank Stepanski
- Recommended reading:  
[JavaScript: A Beginner's Guide, Second Edition](#)