JavaScript Tutorial 1: Simple Rollover

Why JavaScript?

- JavaScript is Object Oriented
    - This means it works with objects
    - Javascript considers the Document (the html page) an object.
    - The document object has a bunch of objects inside of it
        - images
        - links
        - body
        - forms
        - etc.

Creating our first JavaScript (jumping in feet first!)

Create a basic HTML page (Use your template). Add an image to the web page (I added an image called kittenasleep.jpg), making sure that the image has a unique id (I gave the id of "kitpic1").

NOTE: JavaScript identifies different elements on your web page by their **id**. It is tempting to use the src or the href, but it is the id that is significant in identifying which element you're modifying.

Now that you have a basic web page, let's add Javascript.

Step 1: The Script Tag

The script tag tells the browser that the code between the two <script> tags is JavaScript (and not XHTML).

In the head section, add a script tag that indicates the area between the two script tags is a javascript. It will look like this:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html lang="en-US" xml:lang="en-US" xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <title>Our first javascript!</title>
        <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />

        <script type="text/javascript">

        </script>

    </head>
    <body>

        <img src="kittenasleep.jpg" name="kitpic1" width="500" height="375"
            id = "kitpic1"/>

    </body>
</html>
```

Step 2: Add a function.

A function is a name we give a certain section of JavaScript. We give small pieces of code names so that we can "call" that code.

In English, let's say you had a bunch of plants to pot. You might have a list of steps for planting the pot.

1. Empty a container
2. Fill the container half-way with soil
3. Put a plant in the pot
4. Fill around the plant in the pot with soil until the pot has soil to ½inch from the top of the pot
5. Add ½ cup of water to the pot.

If you had 80 plants to pot, you wouldn't want to have to tell someone 80 times to do those 5 steps – both you and the person would go nuts. So you might choose to give that list of steps a name, sort of like this:

Pot_Planting_Steps:

1. Empty a container
2. Fill the container half-way with soil
3. Put a plant in the pot
4. Fill around the plant in the pot with soil until the pot has soil to ½inch from the top of the pot
5. Add water to the pot.

Now every time you want someone to pot a plant, you can just say, "Do the Pot_Planting_Steps" and they'll know to go look at the steps associated with the name "Pot_Planting_Steps" . (Granted, this is a pretty lame example. Most people can remember the steps to potting a plant and thus wouldn't need to have the steps written down and named. But you get the idea of writing out a list of steps to execute, and then giving those steps a name, so you just have to refer to the name when you want those steps to be executed).

In JavaScript we specify that a certain section of code has a name by using the word function followed by the name we want to give the code. We specify where the code belonging to the name starts using { and we specify where the code belonging to the name stops using }. So, since we don't know any JavaScript yet, we'll use our English example above and start to JavaScript it. With our English example, we'd get:

```
function Pot_Planting_Steps()
{
        Empty a container
        Fill the container half-way with soil
        Put a plant in the pot
        Fill around the plant in the pot with soil until the pot has soil to ½inch from the top of the pot
        Add water to the pot.
}
```

NOTE: In a function, you do the steps ***in order*** unless otherwise specified. So you'd always do "Empty a Container" before "Fill the container…", etc.

NOTE 2: Your function names MUST NOT have spaces or special characters (all that stuff that isn't a letter or a number) and they MUST start with a letter (not a number). You function name can have _, but that's only character that's not a letter or number.

Note also the two parentheses () after the function. **WE NEED THOSE**. We'll talk about those more later. For now, just put them.

Back to our code. We're going to change the image on our web page. So since I can call my function anything that starts with a letter, I'll call my function "makepicchange()". My script should now look like this:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html lang="en-US" xml:lang="en-US" xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <title>Our first javascript!</title>
        <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />

        <script type="text/javascript">
            function makepicchange()
            {

            }
        </script>

    </head>
    <body>

        <img src="kittenasleep.jpg" name="kitpic1" width="500" height="375"
            id = "kitpic1"/>

    </body>
</html>
```

Step 3: document.images.

Where we're headed:

```
document.images["kitpic1"].src = " kittybelly.jpg";
```

Explained:

JavaScript deals with objects, which is a general term for "things." In our English example, the pot would be an object, the plant would be an object, the soil would be an object, and the water would be an object. In JavaScript, your entire xhtml page is an object (referred to as the "document"). So is every image in your xhtml page.

Note that there's a bit of a hierarchy here. Your document object (the xhtml page) contains the image objects (the images) that are in that page. Other document objects have their own images.

Currently the source (src) of the only image on this page is "kittenasleep.jpg". I want the JavaScript function we're writing to switch the src to "kittybelly.jpg". So we want the function to change an image, but there may

be more than one image on the page (your xhtml page could have a whole slew of images on it, right?). How do we know which one we're changing? We use the image we want to change's **id**. So to refer to the image on the page that we want to change, we'd say:

```
document.images["kitpic1"]
```

We want to change that images src (we could change its width or its height, but we're changing its src). So to change the src, we'd say:

```
document.images["kitpic1"].src
```

And to change it, we'd set it equal to something new (in this case, our new picture "kittybelly.jpg". So to change the images src to "kittybelly.jpg", we'd say:

```
document.images["kitpic1"].src = "kittybelly.jpg";
```

(The semicolon at the end of the line is just that – an indicator that the line ends).

Putting it all together, we'd get:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html lang="en-US" xml:lang="en-US" xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <title>Our first javascript!</title>
        <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />

        <script type="text/javascript">
            function makepicchange()
            {
                document.images["kitpic1"].src = " kittybelly.jpg";
            }
        </script>

    </head>
    <body>

        <img src="kittenasleep.jpg" name="kitpic1" width="500" height="375"
            id = "kitpic1" />

    </body>
</html>
```

Step 4: onMouseOver – calling the function.

So far we've got a JavaScript, and that JavaScript contains a function, which is a name for a particular set of instructions, or code. But somewhere and somehow we need to tell that function to actually execute the code inside the function. In our English example, you may have a sheet of paper with the instructions for "Pot_Planting_Steps", but until you say to someone, go follow the "Pot_Planting_Steps", probably no one is

going to actually follow the steps to pot the plant. It is the same with code. We need to say somewhere that when a particular action happens, we want the function to run, or execute.

For this example, we'll have the picture change when our mouse runs over the image on our web page when it's displayed in the browser. The term for this is "onMouseOver". Technically this isn't JavaScript. But this is what makes the JavaScript function happens. We add the onMouseOver command to an image tag (or even a header or paragraph tag) in our xhtml page. Once we've done that, when we run the mouse over that particular element, whatever function is called will execute. We'll add the onMouseOver to the image tag in our xhtml page:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html lang="en-US" xml:lang="en-US" xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <title>Our first javascript!</title>
        <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />

        <script type="text/javascript">
            function makepicchange()
            {
                document.images["kitpic1"].src = " kittybelly.jpg";
            }
        </script>

    </head>
    <body>

        <img src="kittenasleep.jpg" name="kitpic1" width="500" height="375"
            id = "kitpic1"
            onmouseover = "makepicchange()" />

    </body>
</html>
```

That's it! That's your first JavaScript, and your first call to a JavaScript. Try it in your browser. (Note if you want to run it again, you'll have to reload it in the browser. We'll fix that in part 2).

This is your first JavaScript, so we're introducing a lot of concepts here. You may need to go back and read over particular parts of this section later as you become more comfortable with JavaScript and want to understand it better.

Questions:

1. What is a function?
2. What is a document?
3. To change an image on your web page, how do you identify that particular image for changing?
4. How do you make a function's code (instructions inside the function) happen?

Part 2: Changing the image back to the original

Right now the image changes when we roll the mouse over it, but it doesn't change back when we roll the mouse off the image. To have the image change back, we'll first want to write another function that changes the image's src back to the original pic, and then we'll have to call that second function when our mouse rolls off the image.

Step 1: Write a second function

We need to write a new function with a new name. I'll call it changepicback. In that function, I want to change the image's source back to "kittenasleep.jpg". So my function will look like:

```
function changepicback()
{
    document.images["kitpic1"].src = "kittenasleep.jpg"
}
```

The xhtml page head should now look like:

```
<head>
      <title>Our first javascript!</title>
      <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />

      <script type="text/javascript">
      function makepicchange()
      {
          document.images["kitpic1"].src = "kittybelly.jpg"
      }
      function changepicback()
      {
          document.images["kitpic1"].src = "kittenasleep.jpg"
      }
      </script>

</head>
```

Step 2: Add onMouseOut

Now we want to call the function (make the code inside the function run, or execute) when we run our mouse off of the image in the page. To do this we'll use the onMouseOut command. In the body, add to the img code an onmouseout event and a call to changepicback when an onmouseout event happens:

```
<img  src = " kittenasleep.jpg" name = "kitpic1" width = "500" height = "375"
      id = "kitpic1"
      onmouseover = "makepicchange()"
      onmouseout = "changepicback()" />
```

Alltogether, your xhtml page should now look like this:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html lang="en-US" xml:lang="en-US" xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <title>Our first javascript!</title>
        <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />

        <script type="text/javascript">
            function makepicchange()
            {
                document.images["kitpic1"].src = " kittybelly.jpg";
            }
            function changepicback()
            {
                document.images["kitpic1"].src = "kittenasleep.jpg"
            }

        </script>

    </head>
    <body>

        <img src="kittenasleep.jpg" name="kitpic1" width="500" height="375"
            id = "kitpic1"
            onmouseover = "makepicchange()"
            onmouseout = "changepicback()" />

    </body>
</html>
```

Try this. Make sure you understand all the concepts so far. If not, reread the tutorial, google, and ask me a lot of questions.

Questions:

1. In the above code, why did I use "kitpic1" inside both functions?
2. Specifically, what am I changing about the img on the page (e.g., the name, the width, the height, the src, or the id?)
3. onmouseover means what?
4. What function is called when onmouseover is performed?
5. What is the result of that function?
6. onmouseout means what?
7. What function is called when onmouseout is performed?
8. What is the result of that function?