For loops and Window Objects:

So far we've been dealing with the document object. The document object has many properties – all elements within the html document are part of the document object. Thus we can refer to the document's images or the document's elements (identified with their id) using document.getElementById. But what if we want to change the properties of a particular tag? What if we want to change all the elements tagged with the <p> tag? Or all the elements tagged with the <h1> tag?

We can do this using getElementsByTagName(). The function getElementByTagName() creates an array of all the elements in a page with a particular tag. So, for instance, if you said

arrayofparagraphs = document.getElementsByTagName('p');

arrayofparagraphs would then be an array of every paragraph in the document (the html page). Then to access a particular paragraph (in the following example, the 4th paragraph), you'd use

arrayofparagraphs[3].innerHTML = "this is paragraph 4".

Try this to see how it works:

```
<script type = "text/JavaScript">
function function2() {
    arrayofparagraphs = document.getElementsByTagName('p');
    arrayofparagraphs[3].innerHTML = "This is paragraph 4";
}
</script>
</head>
<body>
      <p>  This is a paragraph </p>
      <p> This is another paragraph </p>
      <p> Yet another paragraph </p>
      <p> One more paragraph </p>
      <p> Almost the final paragraph </p>
      <p> And the final paragraph </p>
      <input type="button"  value="Which is paragraph 4?"
      onClick="function2()">
</body>
```

Try it

What if we wanted to change the font color of every paragraph in a document? Remember, you change font color using .style.color. So we have an array of paragraphs. This is an array just like the arrays you created using arrayHolder = new Array(). So we can access the elements in the same way. For instance, you could change each paragraph sequentially using a counter that increments every time you click on the button. For instance:

```
<script type = "text/JavaScript">
function function2() {
    arrayofparagraphs = document.getElementsByTagName('p');
    arrayofparagraphs[counter].style.color = "#FC32C1";
    counter = counter + 1;
```

```
}
</script>
</head>
<body>

<p>  This is a paragraph </p>
<p> This is another paragraph </p>
<p> Yet another paragraph </p>
<p> One more paragraph </p>
<p> Almost the final paragraph </p>
<p> And the final paragraph </p>
<input type="button" value="Change paragraph color." onClick="function2()">
</body>
```

Try it

(You can change other style elements as well. There's always background color: .style.backgroundColor).

Now, what if you want to change the color of the text of every paragraph with just one click? For this you'll need to use a loop. The loop will change the text color of the first paragraph, then the text color of the second paragraph, etc. It will stop right after the text color of the last paragraph has been changed.

You've seen while loops before. We could use a while loop to change the color of every paragraph . First we need to know the length of the array: we don't necessarily know the number of paragraphs in the html document, and each paragraph becomes a separate entry into the array created by getElementsbyTagName(). So if each paragraph is a separate array element in the array, to make sure we loop through every paragraph, we need to make sure we loop through every element in the array. Thus we need to know the length (or the number of elements) in the array. We use .length for this. (Remember in our picture show we said, "if (counter == arrayHolder.length)…" ? What we were saying was, "if the value in the counter variable is equal to the length (the number of elements) in the array called arrayHolder…").

So if we want to use a while loop to change the color of every paragraph in th array we've created, we will make the while loop continue until we hit the length of the array:

```
counter = 0;
while (counter < arrayofparagraphs.length)
```

Now we'll change the color of the current paragraph in the array:

```
counter = 0;
while (counter < arrayofparagraphs.length)
{
        arrayofparagraphs[counter].style.color = "#324FF2";
```

And finally, we MUST increase the value inside the counter variable. (If we don't increase the value inside the counter variable, it will ALWAYS be 0, so the value inside counter will ALWAYS be less than arrayofparagraphs.length(). That means that the while loop will continue forever, and never stop. This is known as an Infinite Loop and it is BAD (as in "oops, browser went boom" bad. You really don't want to do it. It just never ever ever ever ever ever ever ever ever ever ever ever ever ever ever stops. Does this make sense?

So our code now looks like this:

```
<script type = "text/JavaScript">
counter = 0;
function function2() {
    arrayofparagraphs = document.getElementsByTagName('p');
    while (counter < arrayofparagraphs.length)
    {
         arrayofparagraphs[counter].style.color = "#3278D2";
         counter = counter + 1;
    }
}
</script>
</head>
<body>

<p>  This is a paragraph </p>
<p> This is another paragraph </p>
<p> Yet another paragraph </p>
<p> One more paragraph </p>
<p> Almost the final paragraph </p>
<p> And the final paragraph </p>
<input type="button" value="Change Text Color" onClick="function2()">
</body>
```

Try it

Instead of using this while loop, we can use a sort of short-hand for this particular kind of loop. The shorthand is known as a "for" loop. Here is what a for loop that does exactly the same thing as the code above does looks like:

```
<script type = "text/JavaScript">
function function2() {
    arrayofparagraphs = document.getElementsByTagName('p');
    for (counter= 0; counter < arrayofparagraphs.length; counter = counter+1)
    {
         arrayofparagraphs[counter].style.color = "#3278D2";
    }
}
</script>
</head>
<body>

<p>  This is a paragraph </p>
<p> This is another paragraph </p>
<p> Yet another paragraph </p>
<p> One more paragraph </p>
<p> Almost the final paragraph </p>
<p> And the final paragraph </p>
<input type="button" value="Changing Text Color" onClick="function2()">
</body>
```

Try it

Let's look at the for loop more closely:

```
    for (counter= 0; counter < arrayofparagraphs.length; counter = counter+1)
    {
```

```
        arrayofparagraphs[counter].style.color = "#3278D2";
    }
```

What this really says is:

for (*variable=starting value*; *variable* < *endvalue*; *variable=variable+1*)
{
code to be executed
}

In our while loop example, we had a separate line in which we put 0 into the counter variable before the loop started. Now that is incorporated into the loop. In the while loop example, we also had to have a separate line inside the while loop that increased the value in the counter variable (so we would eventually reach the end of the array). We also incorporate that line into the for loop at the end. Thus this for loop does the exact same thing as the while loop above.

A simpler example of a for loop is:

```
<script type="text/javascript">
function func2()
{
        for (countvar=0; countvar < 5; countvar = countvar + 1)
        {
                document.write("countvar now holds " + countvar);
                document.write("<br />");
        }
}
</script>
</head>
<body>
<input type="button" value="Click here" onClick="func2()">

</body>
```

Try it

The equivalent code using a while loop would be:

```
<script type="text/javascript">
function func2()
{
        countvar = 0;
        while (countvar < 5)
        {
                document.write("countvar now holds " + countvar);
                document.write("<br />");
                countvar = countvar + 1
        }
}
</script>
</head>
<body>
```

```
<input type="button" value="Click here" onClick="func2()">
```

</body>

Your Turn:

1. Using a while loop, write a function that changes the color of every other paragraph on a page. Now write the same thing using a for loop.
2. Write a function that changes every other paragraph color to one color, and every other paragraph to another color (think stripes). There are a number of ways to do this. I suggest using a loop. You may (but don't have to) want to use the modulus function (which is %). The modulus function gives you the remainder of the division of 2 numbers. For instance, 17/5 is 17 divided by 5, and gives you 3 with a remainder of 2. 17%5 gives you 3. I can use this to tell when a number is evenly divisible by another number. E.g.,

```
firstnum = 15;
secondnum = 3;
if ((firstnum % secondnum) == 0)
{
        document.write("<p> Yes! firstnum is evenly divisible by secondnum! </p>");
        document.write("<br />");
}
else
{
        document.write("<p> Nope. firstnum is not evenly divisible by secondnum. </p>");
        document.write("<br />");
}
```

3. Write a html page with a bunch of paragraphs and headers and more than one button. Change the color scheme of the page using while loops and for loops. Depending on which button you push, the color scheme should be different.

4. Write a javascript function that asks the user to input two numbers, then computes the sum of the numbers in between. If the sum is evenly divisible by 2, write "You're an even-keeled person". If the sum is evenly divisible by 3, write "You're an eccentric, interesting person". If the sum is evenly divisible by 5, write "You're a bit erratic." Otherwise, write "You're a rare bird." (Note that you must determine which number the user entered is the smaller, and which is the larger.)

5. (Challenging). Write a function. This function asks the user to input a number. The function then takes the number and creates a pyramid based on the size of the number. For instance, if the user inputs the number 5, your resulting pyramid can look like this:

```
00000
1111
222
33
4
```

If you prefer, it could look like this:

```
1
22
333
4444
55555
```