

HTML/JAVASCRIPT TUTORIAL FOR DESIGNERS GALLERY SLIDE SHOW PART 2

MUD(TM) - TAKASHI OKAMOTO
MUD@MUDPUB.COM

1. INTRODUCTION

In part 1 of this example, we worked on an html section that enabled us to change the embedded image without having to load a new html file by using javascript. We used a *dummy index*¹ which was named, `imgNumber`. We run into a problem when a user clicks on the next button² more than five times. We run out of images to show and the last image stays on while the counter keeps on increasing. The same can be said about the previous button. We can fix all this by imposing conditions to our javascript. Here we will employ the *if-else statements*.

2. THE CODE

2.1. **HTML.** We will use the html file you created in Part 1 of this example.

2.2. **JAVASCRIPT.** Right now you should have one `<script>` tag inside the `<head>` tag of your html document. The `<script>` should look something like:

```
<script language="javascript" type="text/javascript">
  var imgTotal = 5; // total number of images
  var imgGallery = new Array();
  for (i = 0; i < imgTotal; i++) {
    imgGallery[i] = new Image();
    imgGallery[i].src = 'images/page_' + ( i + 1 ) + '.gif';
  }
  var imgNumber = 0;
  function mudNext() {
    imgNumber = imgNumber + 1
    document.images.imagename.src = imgGallery[imgNumber].src
  }
  function mudPrev() {
    imgNumber = imgNumber - 1
    document.images.imagename.src = imgGallery[imgNumber].src
  }
</script>
```

Date: October 30, 2003.

¹Or a counter variable, or just plain counter or index...you can call it whatever you want, just remember it's just an arbitrary number.

²Which increases the `imgNumber`.

So the problem right now is that the `imgNumber` variable does not stop counting after the last image while clicking on next, and similarly for clicking on previous. We need to add certain conditions to tell the function `mudNext()` and `mudPrev()` to stop increasing or decreasing `imgNumber` after it reaches a certain point. To stop the `mudPrev()` from decreasing past 0 is easier, so we will start there.

The conditional statements *if-else* in javascript, as with other programming languages work like this:

```
if (condition) {
    statement1;
}
else {
    statement2;
}
```

(read: if condition is met, do statement1. Otherwise do statement2.) You can expand on this to include more than one condition. I'll show it here just as reference, but we will not use it.

```
if (condition1) {
    statement1;
}
if (condition2) {
    statement2;
}
else if (condition3) {
    statement3;
}
else {
    statement4;
}
```

So we can also introduce an `else if` and add another condition before reaching `else`. Anyway, going back to fixing the `mudPrev()` function; we can add some *if-else* condition that goes something like:

“If `imgNumber` is greater than 0, then decrease `imgNumber` by 1, otherwise (ie, if `imgNumber` is less than or equal to 0) keep `imgNumber` at 0. Now change the current image with a new image corresponding to the new `imgNumber`.”

How do we translate this into javascript. It's quite easy, and looks like:

```
if (imgNumber > 0) { // check whether imgNumber>0
    imgNumber = imgNumber -1; // if imgNumber>0, imgNumber decreases by 1
}
else { // if the above condition is not met
    imgNumber = 0; // imgNumber should be set to 0
```

```

}
document.images.imagename.src = imgGallery[imgNumber].src // change the image

```

exercise: Modify `mudPrev()` and check that it works!

Now for `mudNext()`. What is the maximum `imgNumber` that we can assign? It should be the last image in the *array*, `imgGallery[]` that we created. We already have a variable called `imgTotal`. Is that the maximum `imgNumber`? Close, but no. Remember that elements in the array start with 0, while `imgTotal` starts from 1. So in our case, if `imgTotal = 5`, then the maximum number in the array is 4. So we should just start by defining a new variable, call it `imgLimit`. It should have the following value:

```
var imgLimit = imgTotal - 1;
```

Like `mudPrev()`, the `if-else` condition should say something like:

“If `imgNumber` is less than `imgLimit`, then increase `imgNumber` by 1, otherwise keep `imgNumber` at `imgLimit`. Now change the current image with a new image corresponding to the new `imgNumber`.”

In javascript, it looks something like:

```

if (imgNumber < imgLimit) {
    imgNumber = imgNumber + 1;
}
else {
    imgNumber = imgLimit;
}
document.images.imagename.src = imgGallery[imgNumber].src

```

exercise: Modify `mudNext()` and check that it works!

You should have the following javascript code:

```

<script language="javascript" type="text/javascript">
    var imgTotal = 5; // total number of images
    var imgGallery = new Array();
    for (i = 0; i < imgTotal; i++) {
        imgGallery[i] = new Image();
        imgGallery[i].src = 'images/page_' + ( i + 1 ) + '.gif';
    }
    var imgNumber = 0;
    var imgLimit = imgTotal - 1;
    function mudNext() {
        if (imgNumber < imgLimit) {
            imgNumber = imgNumber + 1;
        }
        else {
            imgNumber = imgLimit;
        }
    }

```

```
    }  
    document.images.imagename.src = imgGallery[imgNumber].src;  
  }  
  function mudPrev() {  
    if (imgNumber > 0) {  
      imgNumber = imgNumber - 1;  
    }  
    else {  
      imgNumber = 0;  
    }  
    document.images.imagename.src = imgGallery[imgNumber].src;  
  }  
</script>
```

Now you have a properly working gallery! For those who may want a challenge, try the following. Instead of having generic text as your *close/previous/next* buttons, create a graphic button. Make another version of *previous/next* buttons that are “dimmed”³. Now modify the `if-else` conditions such that when you reach the last page the *next* button is dimmed, similarly if you reach the first page, the *previous* button is dimmed. All this can be done just by using **one** html file! Remember, there’s no one way of doing anything. **HINT:** remember to give your *previous/next* images a name. Need more hints? Just email me.

REFERENCES

1. D. Goodman, *Dynamic HTML. 2nd Ed.* O’Reilly & Associates, Cambridge, 2002.
2. D. Flanagan, *Javascript: The Definitive Guide. 4th Ed.* O’Reilly & Associates, Cambridge, 2002.

Typeset using L^AT_EX 2_ε for Darwin, AMS package under Mac OS X and TeXShop.

³eg. make a version that has 50% opacity.