JavaScript: Lab 2: Video Tutorials2

85 pts, due Apr 26 (84 + 1 pt for including your name somewhere)

Complete problems 1-15, as part of the tutorial, below:

Note: these are tutorials that REALLY CLOSELY parallel week 9's videos/ppts. Please watch week 9's videos for this lab.

Contents

HINTS (Last week's plus more!)

Remember, computers are very unforgiving and details matter A LOT! BE VERY CAREFUL ABOUT DETAILS.

Checklist: If your code doesn't work, check the following:

- A " and a '' are completely different things to the computer. Only the second will work.
- If you opened it, you must close it
    - That goes for ( ), {} , " " and ' '
- Capital and small letters are completely different to the computer and bear no resemblance to each other whatsoever.
    - Alert and alert are completely different words to the computer (only the second one works),
    - as are Function and function (again, the second one works).
- When you name something, do not put any spaces or special characters in the name. Equally, don't start with a number. So:
    - this name is bad
    - This*&^Name is bad
    - 2names is bad
    - thisName1 is good!
- x = prompt("…")
    - Make sure there's a variable on the left of the prompt!
        - Doesn't have to be x – can be any name we want to give it as long as it follows the naming rules (above)
    - Small p for prompt
- r = Math.floor(Math.random() * 8)
    - make sure you've got a variable to the left of Math.floor(Math.random…
        - doesn't have to be r – can be any name we want to give it as long as it follows the naming rules (above)
    - capital M, small f, capital M, small r
    - Watch your parentheses so they match exactly
- document.getElementById()...
    - small d, small g, Capital E, Capital B, capital I, small d
    - between the parentheses MUST go either:
        - an id from your web page (so in quotes) or
        - A parameter or variable holding an id from your web page (not in quotes)
    - document.getElementById('imgid').src only works for images on your web page (it's the src)
    - document.getElementById('p1').style.
        - used to modify styles of a tag on your web page
        - whenever you use a style,
            - don't use –
            - Capitalize the second word
            - E.g., borderStyle, backgroundColor, borderWidth

NOTE: So that your html and javaScript files don't get overwhelmingly confusing, please start a new html and javascript file for these second set of js tutorials.

Part 1(Random Numbers)

So far the programming tools we've learned are:

- Functions,
- Parameters
- Variables
- Alert box
- Prompts
- If conditions
- Else if and else with the if condition

Now we're going to add random numbers!

Random Numbers Explained

Every computer game in the world uses random numbers. Things have to happen randomly or it's not much of a game. Random numbers are used any time you want something to happen unexpectedly – for instance, you may want random pictures to show up in a display, or you may want random words of encouragement to pop up in an app.

These are just a few of many uses for random numbers.

Previously you learned about variables, and about 2 different ways to put values into variables. You can do:

```
x = 3
```

and that puts 3 into the variable x. Once 3 is in x, every time you use x, it is the same as if you were using the number 3.

You can also do:

```
y=prompt("What is your age?")
```

and when the user types in their age, it goes into the variable y. Again, once the age is in y, every time you use y, it is as if you are using the age that was typed in.

You can also put random numbers into variables. Every language lets you do this in a slightly different way. JavaScript lets you do this using:

```
x = Math.floor(Math.random() * 10)
```

This generates a random number between 0 and *10 (not including 10)* and puts that random number into the variable x.

The number 10 in the above example is what specifies that the largest random number to be generated will be 9. You can always change this number. For instance, if you have the following random number generator:

```
x = Math.floor(Math.random() * 3)
```

The possible random numbers that x might hold would be 0, 1, or 2 (not 3 - the smallest number is 0, and the largest is always one less than the number after the * (which is multiplication in javaScript).

Now, what if you want a random number between 5 and 15? You'd do the following:

```
x = Math.floor(Math.random() * 10) + 5
```

The first part generates a random number between 0 and 9, and then, whatever that random number is, 5 is then added to it. So, for instance, if the random number generated is 0, then after 5 is added, x will hold 5. If the

random number generated is 4, then, after 5 is added, x will hold 9, and if the random number generated is 9, then x will hold 14.

Thus the smallest number the random number can generate, which is 0, plus 5, will make the smallest random number 5, and the largest number the random number can generate, which is 9, plus 5,will make 14, so the range is from 5 to 14, which is what we wanted.

You can just memorize how to use this, or you can read the following explanation:

x = Math.floor(Math.random() * *9*)

1. Math.random() generates a random number between 0 and 1 (e.g., 0.4).
2. This number is multiplied by the range (in this case 9):
    a. Math.random() * 9 gives us 3.6
3. Math.floor(3.6) rounds down to the nearest integer, or 3
    a. It chops off everything after the decimal point
4. Finally, you need a variable to put all this in.
    a. x = 3

Let's try random numbers*:*

Step 1: In your html file create a button. Make the button call a function when it is clicked on. Give the button a value. You can make up the name of the function and the value displayed in the function.

Step 2: In javascript file, create a function with the exact same name as the function that is called when you click on the button from step 1.

Step 3:. In the function, generate a random number between 0 and 6 and place it in a variable

Step 4: Use an alert box to show the random number generated.

Now save both files and test.

If it isn't working:

Make sure you've:

- Linked the javascript file to the html file using <script src = …></script> (like in the first tutorial)
- Have the function name in the html file and in the js file EXACTLY the same
- Have both an opening and closing { } at the beginning and ending of the function
- Make sure you have a small f in floor and a small r in random, but a large M for Math.
- Make sure that you have the parentheses exactly right – two opening and two closing parentheses in the Math.floor(Math.random()*7)

Once you have this working, let's expand a bit:

Weather Prediction

First, let's use the random number to bring up a random alert, in this case for the weather.

(Let's be honest – at some point it's crossed everyone's mind that the weather report very well might just be randomly generated)

The above function currently has a variable that will hold a random number, and that random number can be either 0, 1, 2, 3, 4, or 5 (not 6).

Step 5: Now in your .js: after the random number has been generated and placed into a variable, add an if condition.

Step 6: In the if condition, check to see if the random number variable is equal to 0, and if so, use an alert to show a type of weather ("sunny", "thunderstorms","tornados","windy", etc.). (see video)

Step 7: add an else if for if the random number is 1, and then alert with a different weather condition

Step 8: add an else if for if the random number is 2, and then alert with yet another weather condition

Step 9: continue to add else if conditions for random numbers 3 through 5, each with an alert of a different weather condition

Step 10: save and make sure it's working.

If it isn't working 2:

- Double-check all of If it isn't working 1
- Make sure that after each if and else if you've got an opening and closing ( )
- Make sure that when checking to see if your variable is a number, you use === (not =)
- Maker sure that each if and else if has an opening and closing { }
- Make sure the variable that you're using in the if, i.e., if (x === 0) {…. } is exactly the same as the variable that holds the random number, i.e., x = Math.floor(Math.random() *3)

Your Turn:

Problem 1: (5 pts) Weather Prediction Get the weather prediction function working

Problem 2: (7 pts)Magic 8 ball

You've all played with the magic 8 ball, in which you ask the ball a question and then shake it and an answer to your question pops up? You might ask, "Will I win the lottery?" and then shake the ball and the answer appears that might say something like, "It is highly unlikely!"? You're going to write this. Note that the typical magic 8 ball has 8 possible answers, and they are:

- As I see it, yes.
- Ask again later.
- Better not tell you now.
- Cannot predict now.
- Concentrate and ask again.
- Don't count on it.
- It is certain.
- It is decidedly so.

But feel free to be creative and come up with 8 of your own potential answers.

Create a magicEight function in your javascript file.

Inside the function, use a prompt box to ask the user to enter a question. Place a variable on the right to hold the answer. NOTE: you will not use the variable that holds the answer again. THERE IS NO CORRELATION BETWEEN THE QUESTION AND THE ANSWER THAT POPS UP!

Now below the prompt box, generate a random number between 0 and 8 (again, note that the possible random numbers generated will be 0,1,2,3,4,5,6, and 7, but not 8) and make sure you place the random number in a variable, to the left of the random number generator.

Now, like you did with the weather predictor, use the random number generated to bring up one of the 8 possible answers. So you might have something like,

```
if (x === 0) {
        alert("As I see it, yes.")
}
else if (x === 1) {…
```

Add a button to your html code that calls this magicEight function when you click on it

Test and make sure the function is working.

Problem 3: (8 pts) Lucky Guess

This problem is similar to the problem from the last lab, where you had one button that was the right button, and all the other buttons were wrong. But now the "right" button is going to be determined randomly.

In your html code, add a table with 6 buttons (more if you like). Give each button a value of 0,1,2,3,4, and then 5, respectively. Make each button call the same function when clicked on. However, when the button is clicked on and the function is called, make sure each button calls the function with its number as input to the parameter. So you might have something like this:

<input type = "button" value = "0" onClick = "Guessfunc(0)">

Please don't copy, or if you do, make sure you retype the quotes. Remember, quotes in Microsoft word don't work in javaScript because they're funky.

Do this for each button.

Now in your javascript file, write a function with the same name as the one being called by the above buttons. Make sure it has a parameter to hold the number being passed in.

Inside the function, generated a random number between 0 and 6, and make sure you have a variable to hold that random number on the left side of the random number generator.

Now use an alert box to show the random number

Now use an if condition to check to see if the random number is the same as the parameter. If it is, use an alert box to say, "You guessed correctly!!"

Otherwise, have an alert box that says, "Sorry, wrong button".

Save your files. Test and make sure it works.

Part 2: document.getElementById()

So far we don't have a way of dynamically changing existing elements on our web page. For instance, what if I wanted to click on a word and have the web page's picture change to one representing that word (i.e., I clicked on an elephant and a pic of an elephant appears on the web page, along with a paragraph about elephants)? What if I wanted to be able to make an image smaller or larger by clicking on a button? What if I was low vision and wanted to make the font size larger on my web page?

All these can be done in a number of ways. We're going to start by learning about how to use:

```
document.getElementById()
```

getElementById only lets you change elements in your html code that have an id. Luckily you can go in and give any element an id. Since every id in your web page must be unique, this only allows us to change one thing at a time.

Let's start by using it to change an image on your web page.

.src

Example: if you have an image on your web page, it might look like this in the html code:

```
<img src = "Images/penguin.jpg" style = "width: 220px; height: 220px;"  alt = "a cute baby penguin" >
```

You can add an id to the image (e.g.,)

<img src = "Images/penguin.jpg" style = “width: 220px; height: 220px;” alt = "a cute baby penguin" **id = "pic1">**

Then in the html page, have a button that calls a function that changes the image, or something like this:

<input type = "button" value = "Click to change" **onClick = "changeSrc()">**

Now, let’s create the function that changes the pic. In your javascript, create a function-in this case, name it changeSrc()

Inside the function, add a document.getElementById. To identify the image uniquely, I’d say the following:

document.getElementById(“pic1”)

and to change the pic that shows up, I want to change the src

Note that in your html code, the img src=”penguin.jpg” is what sets the image you see on the web page to be the penguin.jpg pic. So to change the pic, I must change what the src is set to.

So if I wanted to change the pic to a wombat, I’d have inside my function,

document.getElementById("pic1").src = "Images/wombat2.png"

(I have my wombat2.png file inside an Images folder)

So my entire function looks like:

```
function changeSrc() {
        alert("Changing pic")
        document.getElementById("pic1").src = "Images/wombat2.png"
}
```

You can try the above:

In your html function:

Step 1: Add an image to your html page. Save the web page and load it. Make sure the image shows up.

Step 2: Give the image a unique id. It can be anything, as long as nothing else on your web page has that id (and it can only be 1 word – no spaces or special characters)

Step 3: Underneath, add a button that calls a function that changes the image when it is clicked on.

Now in your javaScript file:

Step 4: Create a function. Make sure the function has the exact same name as the html button calls.

Step 5: Inside the function, include an alert box that alerts the user that the pic is being changed.

Step 6: Now add the document.getElementById. Make sure you have the id that you gave to the image in your html page between the parentheses, and make sure you’re changing the .src to another image

Step 7: Save this file. Test Everything and make sure everything works.

If it’s not working:

- Make sure that the id you gave to the image is the id between the parentheses for document.getElementById
- Make sure the id is in quotes in between the parentheses in document.getElementById(“pic1”)
- Make sure you did not capitalize the d in Id, just the I (which is an I as in Iguana, not an l as in llama)
- Make sure you did not copy from this document and thus have funky quotes that MS Word creates
- Make sure your html and your js files are linked

(I am trying very hard to go back and change the I in Id to be Times New Roman so you can see the difference between a capital I and a small case letter l, but I may miss one here and there. They're EYES, not ELLS!)

Your Turn:

Problem 4: (5)ChangeSrc Get the above function working

Problem 5: (6) Image as Button

Instead of a separate button, you can add the onClick command directly inside the img html tag, e.g.,

```
<img src="bla.jpg" style ="width: 300px; height: 200px;" alt = "blab la bla" id = "mybla" onClick = "thefunc()">
```

Try It

1. Add a new image to your web page. Give it a different unique id.
2. Download an alternate picture. I went with a snail pic and an explosion pic, but you can go with whatever you like.
3. Now add another function to your .js code. The function should change the .src to the explosion pic.
4. Now modify the image in your html code so that when you click on the image, the function you just wrote is called.
5. Save all and test by loading into the browser. When you click on the image, it should change to your second pic.

Problem 6: (8)Random Pic

1. For this in your html add another image. Give it a unique id. Make the image call a new function when the image is clicked on.
2. Download 3 more images
3. In your javaScript, create another function with the same name as the function you're calling with the image you just added.
4. Inside the function, generate a random number between 0 and 3 (not including 3). Make sure you've got a variable to the left of the random number generator.
5. Below that, use an if condition to change the src to the first image you downloaded if the random number variable holds 0, the second if the random number variable holds 1, and the third if the random number holds 2.

Passing the ID in as a parameter

When you call a function, you've passed in words (strings) and numbers so far. You can also pass an id into a function's parameter when you call the function. So, for instance if you have the following image code:

```
<img src="bla.jpg" style = "width: 300px; height: 200px;" alt = "blab la bla" id = "mybla" onClick = "thefunc()">
```

Right now when you call the function thefunc(), you are passing nothing in, so the function in your javascript file should not have a parameter.
However, you can add a parameter. You can place the id of the image between the parentheses in the onClick as follows:

```
<img src="bla.jpg" style = "width: 300px; height: 200px;" alt = "blab la bla" id = "mybla" onClick = "thefunc('mybla')">
```

And then in your javaScript, make sure the function has a parameter as follows:

```
function thefunc(idpar) {
        document.getElementById(idpar).src = " bang.png"
}
```

So now when you click on the image, you're passing in the image's id into the parameter, and then the parameter, which holds the id is what goes between the parentheses for the document.getElementById().

NOTE THAT VARIABLES and PARAMETERS DO NOT HAVE QUOTES AROUND THEM!

And now, when you click on the image, that image's id goes into the parameter, and inside the function it is used in document.getElementById, and the document.getElementById is changing the src of the image with the id passed in to be "bang.png"

Your Turn:

Problem 7: (5pt)Image id as a parameter:

1. In your html, add an image, and give the image an id. Make it so that,when the image is clicked on, a function is called. Make it be called with the id of the image (make sure that you put single quotes around the id when it is in between the parentheses when you call the function)
2. Now in your javascript create a function. Make sure the function's name is the same as the function called by the image above. The function should have a parameter.
3. Inside the function, use document.getElementById and the input parameter holding the id of the image above to change the src of the image to a different image
4. Save both files, test and make sure it works.

If it doesn't:

- Make sure the id of the image and the id you pass into the function are exactly the same.
- Make sure there are single quotes around the id when you call the function with onClick
- Make sure that you do not put quotes around the parameter when you're using it inside the paretheses in document.getElementById

Problem 8 (6pt): Multiple Images

1. To your html file, add 3 more images, each with its own unique id. Make each of the images call the function you wrote for Part 1, right above, when clicked on, and make sure each image passes in its unique id when it calls the function.
2. Now save and test your code. When you click on any of the above 4 images (1 from part 1 and 3 from part 2), the image's src should change to the image you set it to in the function

Part 3: getElementById to Dynamically Change Style

Using document.getElementById() with the id of an image on your web page, you can modify:

- the src (we've seen this)
- the alt (if you changed the src to, say, "tornado.gif", you would want to change the alt to "pic of a tornado" as well)

We can also use document.getElementById to dynamically change the style of a web page tag (images and other tags)!

Using document.getElementById to change style:

To use document.getElementById to change the style of an html element on your web page, the element you're changing must have a unique id.

Step 1 Add a paragraph to your html code. Give the paragraph an id (I gave it an id of 'p1'). With the id, my paragraph looks like this:

```
<p id = 'p1'> Computers are good at following instructions, but not at reading your mind.
</p>
```

Step 2: Below this paragraph, create a button. The button should have a value of "green", and should call a greenfunction when it is clicked on. The id of the above paragraph should be passed into the parameter when the button is clicked on. So you might have something like this:

```
<input type = "button" value = "Green" onClick = "greenfunc('p1')">
```

Step 3: now in your .js file, create a new function. This function should take an input parameter. This function has to have the same name as the function called with the input button, and it must have a parameter.

Step 4: inside the function, use document.getElementById to set style of the element whose id was passed into and is now held by the parameter. To change the style, you should use .style and then specify the style you want to change. So my greenfunc looked like this:

```
function greenfunc(x) {
        document.getElementById(x).style.borderColor = "DarkGreen"
        document.getElementById(x).style.backgroundColor = "LightGreen"
        document.getElementById(x).style.color = "DarkGreen"
}
```

Step 5: Save your files. Test to make sure they work

If it doesn't:

- Make sure you are using borderColor and not border-color
- Make sure you use = and not :

Your Turn:

Problem 9: (3 pts) ChangeStyle1: Get the above code working

Problem 10: (6 pts)MoreColors:

1. In your html, add two more buttons: one that calls a redfunction, and one that calls a bluefunction. Make sure the buttons call those functions with the id of the paragraph
2. Now in your javaScript file, create two new functions: a redfunction, and a bluefunction. Both functions should have a parameter that will hold the id of the element being passed in.
3. For each function, modify the style being created so that it is appropriate to the function (e.g., for the red function the border becomes a reddish color, the background color becomes a different reddish color, etc., and for the blue function the border becomes a bluish color, etc.
4. Test your 3 buttons – each should change the paragraph's color style to be that of the button they clicked on.

Problem 11: (5 pts): For Other Elements

1. In your html code, add a header element. It can be an <h1> or a <h2> or any of the header elements. Give the header an id.
2. Below the header, copy the three buttons from above and modify them slightly so that they now call the same functions, but with the id of the header you just created.
3. Save and test your code. When you click on any of these 3 buttons, the header's style should change to the color specified.

Some things you can change with the style:

- backgroundColor
- backgroundImage
- backgroundPosition
- backgroundRepeat
- borderColor
- borderStyle
- borderWidth
- margin
- padding
- width
- position
- color
- fontFamily
- fontSize
- fontWeight
- fontStyle
- lineHeight
- textAlign

And more…

http://www.w3schools.com/jsref/dom_obj_style.asp

So, for instance, you could have a function as follows:

```
function changeStyle(par) {
        document.getElementById(par).style.backgroundColor= "DarkRed"
        document.getElementById(par).style.color= "LightYellow"
        document.getElementById(par).style.border= "6px solid orange"
        document.getElementById(par).style.padding= "15px"
        document.getElementById(par).style.fontFamily= "arial"
}
```

That changes the border, the padding, and the font's family as well as the background color and the font's color

Your Turn:

Problem 12: EasierToReadStyle (6 pts):

1. Create a function in your javaScript that has a parameter that will hold the id of an element on your web page. Inside the function modify the element's style so that the element would be easier to read: increase the size of the font, set the background color to black or very dark and the font color to white or very light, change the font to a sans-serif font like arial, and change the line height to be nicely spaced.
2. Now back in your html page, create a few paragraphs and headers (h2, h3, something like that) At least 2 paragraphs and at least 1 header. Give each of these its own unique id. Make each of these elements call the easiertoread function when you click on it, and make sure it passes in its id.
3. Save and test – now when you click on those paragraphs and header, their style should change so that the element clicked on is easier to read.

Changing width and height

As you can see in the above style elements, you can change the width and height of an element.

Step 6: In your web page, add an image. Give the image an id. Style the image so that it's relatively small on the web page.

Step 7: Create a button below the image. The button should have a value of "make larger" and it should call a function with the id of the image as the value that will go into the function's parameter.

Step 8: Save and make sure that the image shows up.

Step 9: Now in your javaScript file, create a function with the same name as the one the button above calls when clicked on. The function should have a parameter (that will hold the id of the image)

Step 10: Inside the function, change the style of the image's width and height using document.getElementById and the style.width and style.height. My function looked as follows:

```
function grow(x) {
        document.getElementById(x).style.width = "300px"
        document.getElementById(x).style.height = "300px"
}
```

Step 11: Now save your javaScript. Test your code. When you click on the button below the image, it should grow to 300 px in size.

Your Turn:

Problem 13:Grow (4 pts) : Get the above code working

Problem 14:Shrink (4 pts): Create a second button that makes the image smaller. Create a corresponding javaScript function that changes the style to make the width and height smaller

Changing the width and height randomly:

We can change the height and width of an element to be a random number. There are a few small tricks to do this successfully.

First, you don't want to ever generate a width or height of 0. A width or height of 0 would mean you don't see the image (no pixels). So let's make a minimum width/height be 20. And let's say the largest width or height we want is 470 (I just made that number up). We'll want to shift the range of random numbers up by 20 (see the Random Number video/tutorial for more info)

The range is 450 (470 – 20), so I'd want to generate a random number, with a range of 450:

```
w=Math.floor(Math.random() *450)
```

And then shift it up so that the smallest number is 20:

```
w=Math.floor(Math.random() *450) + 20
```

Either way will adjust the random number in w so that it will now be 20 to 470 instead of 0 to 450.

So so far the function would look like this:

```
function randSize(x) {
        w = Math.floor(Math.random() * 450) + 20
}
```

The next part is using the random number to set the size.

Normally you would just do the following:

```
document.getElementById(x).style.width = "300px"
```

The number we want to replace is the 300, so instead of 300, we want to use w.

But, of course, the w should not go inside the quotes (because variables don't go inside quotes).

So somehow you want to join the variable w's content with the "px" because we need to specify that it's pixels.

To do that, you'd do the following:

```
function randSize(x) {
        w = Math.floor(Math.random() * 450) + 20
        document.getElementById(x).style.width = w+"px"
}
```

(Note: if you copy this, make sure you retype the quotes because of the whole ms word funky quote issue)

And finally, we'll repeat the process for the height:

```
function randSize(x) {
        w = Math.floor(Math.random() * 450) + 20
        document.getElementById(x).style.width = w+"px"
        h = Math.floor(Math.random() * 450) + 20
        document.getElementById(x).style.height = h+"px"
}
```

Your Turn:

Problem 15: (6 pts) Random Size

Get the above working so that when you click on the button that calls this function, the image's size changes randomly.

WOO WOO!!! YOU'RE A PROGRAMMER!!! GO GEEKS!!!