SetTimeout2:

With positioning and random numbers

(Due Thurs)

Contents

Note: The following exercises will (if you so choose) be part of your final project. As such, you will probably want to create a new html page and a new .js page for these exercises Your final project will be a game in which your goal is to catch as many objects as possible with your hero. For mine I had my goodies be scared people/animals and my hero being a zombie out to catch the goodies to eat their brains. I've got a warped sense of "hero". You can pick any theme you want. It should have something that will catch something else, and something that will be caught. The thing that will be caught is your goodie.

Positioning Dynamically:

So before we use setTimeout, let's use document.getElementById and style to change the position of an element dynamically

Step 1: Add an image of a goodie that you'd want to catch in your final project – it should thematically go with whatever hero you picked above, so my zombie wanted to catch scared things and eat them. So scared things were my goodies. I gave my scared things the id 'c1' and I gave it an in-line style that included positioning the image absolutely. You can position it anywhere to start (I think I positioned mine off the screen). But be careful not to position it on top of the button you'll create in step 2.

Step 2: add another button, and, again, position it absolutely (maybe 400 px from the top and 150 px in from the left, with a z-index of 11). Make this function call the function ChangePos('c1') when clicked on (this button will eventually go away in the final project – we're using it to test the ChangePos function)

Step 3: Save your html and load it to see what it looks like so far. Make sure the images are showing up.

Step 4: In your .js file, add 2 global variables at the top of your js file for the goodie's top and left position (mine were ctop and cleft). Set them to the same exact position you positioned your goodie to in step 1.

Step 5: Create a function ChangePos(par). In this function, set ctop to be 400, and set cleft to be 400.

Step 6: Now use document.getElementById(par).position = "absolute" (You already positioned the image absolutely in your inline style in your html code – this is just to make absolutely (he he he) sure the image is positioned absolutely.

Step 7. Set the images left position using:

```
document.getElementById(par).style.left = cleft + "px"
```

Step 8: Set the image's top position (remember, you're positioning down from the top) using:

```
document.getElementById(par).style.top = ctop + "px"
```

Step 9: Save. Load into a browser and test by clicking on the button.

Your goodie should move 400 pixels down from the top and 400 pixels in from the left.

Randomly Position

So now you know how to reposition an element on your web page. You could reposition a side bar, a table, a header, a div tag – yo you can reposition any element with an id using the position style.

But right now every time you click on the button, it moves to exactly the same place. That's boring. Let's generate a random value between 0 and 800 and set cleft to be that random number. We can then use the random number in cleft to position the goodie over to the left that random number. We can generate another random number for the ctop variable and position the goodie down from the top that random number.

Step 10: To start, generate a random number between 0 and 800 and place it in the cleft variable, as follows:

```
cleft = Math.floor(Math.random()*800
```

(you will be replacing the line, cleft = 400 with the above line)

Step 11: Set the left position to be the random a value as follows:

```
document.getElementById(par).style.left = cleft + "px"
```

Step 12: Underneath (but still within the ChangePos function) Repeat steps 10 and 11, only for the ctop variable. So generate a random number between 0 and 800 and place it in ctop, then use document.getElementById to set the style.top to the ctop variable.

Step 13: Save. Load into a browser and test by clicking on the second button. Every time you click, your goodie should move randomly all over the page.

setTimeout – Automatically Moving:

Right now, you've got to click to reposition the goodie. But what you might want is for the goodie to move around to random locations on its own – maybe jump around every 5 seconds. To do this, we'll use (yep, you guessed it!) setTimeout.

In this case, what we're trying to replicate is your clicking on the button in your html code. Every time you click, you call the ChangePos function. So in this case, we're going to have setTimeout call the ChangePos function – even though it's calling it from within the ChangePos function!

Step 14: Add as a final line to the ChangePos function a setTimeout that calls changePos

Your Turn:

ChangePos (8 pts)

Get the above working!