All Together2:

Global Variables, multiple functions sharing data, changing style,

Note: The following exercises will (if you so choose) be part of your final project

Contents

lobal Variables, multiple functions sharing data, changing style,		. 1
Movii	ng position:	. 1
Your Turn:		. 2
1.	(5 pts) Get the above working	. 2
2.	(7 pts) Move Up and Down	. 2

Moving position:

.In the last tutorial, we repositioned elements using random numbers to set the number of pixels down from the top and over from the left.

For this exercise, you will be systematically moving an image by adding or subtracting 10 to its current location.

Step 1: in your html page, add an image of something that moves that can be the hero character in your final project (that will go around the page and "catch" things (mine was a zombie, but you can pick any themed hero you want). Since images belong inside of another tag, place it inside of a div tag and give the div an id ('z1' in my case) and give the div an in-line style. The style should include positioning the image absolutely. Start by positioning it 0 px down from the top and 0 px in from the left.

Step 2: add a button to the html page, outside of the div, and, again, position it absolutely (I did mine approximately 400 px from the top and 150 px in from the left, but you can always adjust this if it isn't where you like it or ends up under something) and a z-index of 11. (you can always play with all the numbers if you're not happy with any of them) Make this function call the function MoveRight('z1') when clicked on .

Step 3: Save your html and load it to see what it looks like so far. Make sure both the image and the button shows up.

Step 4: Still in the html, create yet another button. This button calls the function MoveLeft('z1') when clicked on, so make the button's value be 'Left'.

Step 5: Now in your .js file, you're going to create a global variable that will hold the position of the hero. You want the position variable to be global because you want both a MoveLeft and a MoveRight function to use the same position variable.

So at the top of your .js file, create variable hleft and set it to the same number you positioned the hero div in from the left in step 1. (i.e., if I positioned my zombie 20 pixels in from the left absolutely in step 1, I want to set hleft=20)

Now, because the variable hleft is outside and above all the functions, it can be used by all the functions you write below it in the .js file.

Step 6: Now create a function MoveRight(par).

Step 7: Add 10 to the hleft variable

hleft = hleft + 10

Step 8: Now, to be safe, just like we did with the scared goodie in the last tutorial, use document.getElementById(par) to position the zombie absolutely.

Step 9: Next use document.getElementById(par).style.left =hleft to reset the image's position over from the left. Altogether my function looks like this:

function MoveRight(par) {

hleft = hleft + 10 document.getElementById(par).position "absolute" document.getElementById(par).style.left = hleft + "px"

Step 10: Save this and then test it by clicking on the right button in your web page. Your hero image should move consistently to the right.

Step 11: Let's repeat the process for a MoveLeft function. Note that for this you do not have to add a separate global variable, because for moving right and left, you want to use the same position variable. So in your .js file create a function MoveLeft(par).

Step 12: The function MoveLeft should be EXACTLY the same as the MoveRight function, with one difference: you're just going to subtract 10 from hleft to move to the left.

So moving to the right means adding 10 to hleft, and moving left means subtracting 10 from the hleft pos. The line hleft = hleft + 10 should be changed to hleft = hleft - 10. This way you're moving over 10 pixels to the left each time.

Step 13: Save this and test all your functions. When you click, your hero should move right or left depending on which button you clicked on.

Your Turn:

}

- 1. (5 pts) Get the above working
- 2. (7 pts) Move Up and Down: Repeat steps 1-13 for MoveUp and MoveDown (you'll need another global variable for htop)