Maxwell Bryant
3/9/16

Web Design: Quiz

1. We can call a JavaScript file in the body section of an HTML page. It is recommended that you call it at the end of the body because if you want to call ID tags, you'll have to put them before calling the JavaScript. For example, <script src = "Java_Stuff.js" type = "text/javascript"></script>
2. Like I explained above, You must put JavaScript at the bottom of your page to call everything correctly. To iterate more, we can use the term "Document Object Model" or DOM, which states that a web page is to go in order of its tags and bodies. You start with the HTML tag as you should start with your appetizer at dinner. You don't start with dessert first, that would be plain awkward, as would putting a footer tag at the start of your page. Here is an example, <button id = "stuff" onclick = "stuff(0)"></button>... and then… <script src = "Java_Stuff.js" type = "text/javascript"></script>
3. We can make random numbers by creating the "Math" instance which can do many things. After declaring "Math", we declare the "random" modifier which will randomize the integers in your list. Finally, we have the list of integers or just a random (pun intended) list which will be shown in the example. Example, Math.floor((Math.random() * 4) + 0);
4. In JavaScript, it is extremely easy to create an Array. All you must do is make a variable, name the variable, and finally create the items you wish it to contain. Example, var cars = ["Camaro", "SRT Viper", "Suburban", "F-150"];
5. We can pass value in JavaScript by giving items in the HTML id tags. It's like when you tag cows, their records can be maintained and transferred as they mosey along in life because they are given a tag. Example in this would be… In HTML: <p id = "example">Text goes here.</p>... and in the JavaScript: document.getElementById("example").innerHTML = "Changed";

6. Like I explained above, the getElementById command in JavaScript will receive an id tag from the HTML as labeled. I could even use the example above: <p id = “example”>Text goes here.</p> then, document.getElementById(“example”).innerHTML = “Changed”;
7. I was able to change the colors in the CSS in JavaScript primarily through tags and ids. For example, I made the body under the id “body” (pretty original too) and in the JavaScript, I wrote this: var change = document.getElementById("body"); Finally, I changed the color and other things after calling the id: change.style.backgroundColor = '#212171';
8. We check values if they are greater, less than, or equal to because of logic. Hypothetical situation: Billy is special. Billy doesn’t know anything unless you tell him to know anything. You tell Billy, “Which is more? Five dollars or two dollars?” Billy then cries because you didn’t tell Billy that 5 was greater than 2. This is how greater than, less than, and equal to works in JavaScript. You need to know so that the computer can know.
9. We can compare variables in JavaScript by doing “==” this will use two variables and check to see if they equal each other. If they do not equal each other, then the code will ignore it. Now, if you use “!=” then it is the opposite. This will check to see if they are not equal, if they are, the code will run. If not, the code will ignore it once again.
10. The data types in JavaScript are: numbers, strings, arrays, and objects. Before creating all these thing however, you must have “var” in front of it. This will ensure to the program that it is a data type of some sort. Example: var names = [“Billy”, “Joel”, “Reid”, “Fontana”]; this is an array data type because it lists strings under the array name “names”.

Web Design: Essay

JavaScript; a tedious, evil character indeed. However, if you learn how to harness its power, you can do many great things. This is how I created a rock, paper, scissors game. The original outline in mind was to do rock, paper, scissors, lizard, Spock because we needed five buttons. Although, we had already used a working CSS switch color button, so instead used that. Then, with the footer and epilepsy in mind, I created a footer in which you could hover over it and it would change color. I thought of this as a button because I had put a bit of work into it. Then, I started to create the layout of the page, I love minimalistic styles, so my webpages represent that. After this, I started to create my rock, paper, and scissors. I felt great about everything until I tested it. It had failed again and again; more than any man would want in their life. The last straw was asking Mr. Davis what to do. After this, it had finally worked! However, it was the last day, so I could not create an AI for it.

So, how everything in my website primarily works is by giving the JavaScript the ability to call tags from the HTML. For example, in the html, I could create a button like this: <button id = “rock” onclick = “choice(0)”></button>. In the JavaScript, I could call it like this: document.getElementById(“rock”).innerHTML = “I changed stuff”;. The innerHTML portion of the example actually allows you to change what everything does. In my case, I replaced it with an image like this: .innerHTML = “img src = ‘rock.png’>”;. This changed a p tag with nothing, into an image retrieved from a directory.

In conclusion, rock, paper, and scissors can be more than meets the eye when it comes to computers. The outline and how everything works in the JavaScript is like a brain; you just have to create it. So keep playing my friend, keep playing.