**Ch 3 Written Assignment**

1. In your own words, state what you think the thing-a-ma-jig is supposed to do?
   1. It did teach me how the code should be executed since it is not necessarily that the code should be executed in order.
   2. It did teach me the difference between Global vs Local variables and how the Global variables are being shadowed when used in a function (e.g. clunkCounter variable was constantly changing inside of the display function)
   3. As for the actual code, when passing 1 or 2 to the thingamajig function, it returned clunkCounter as 1, 2 respectively however, when passing value of 3, it returned 6, value of 4 returned 24, value of 5, returned 120, value of 6, returned 720, value of 7 returned 5040.
      1. I tried to see the sequence of the output “1, 2, 6, 24, 120, 720, 5040,…”
      2. With that being said, it looked like the output usually is the factorial of the argument passed to the function thingamajig.
         1. 1! = 1;
         2. 2! = 2\*1 = 2;
         3. 3! = 3\*2\*1 = 6;
         4. 4! = 4\*3\*2\*1 = 24;
         5. 5! = 5\*4\*3\*2\*1 = 120;
         6. 6! = 6\*5\*4\*3\*2\*1 = 720;
         7. 7! = 7\*6\*5\*4\*3\*2\*1 = 5040;
   4. In the text book page 121 it is stated that the Thing-A-Ma-Jig was invented by a curious chap who was fascinated by rearranging words. It is like DOG rearranged is God, OGD, DGO, GDO and ODG. So if a word has three letters, the Thing-A-Ma-Jig says you can make six total combinations from those letters. In the end of the paragraph, it looks like my assumption of rendering mathematical factorial was correct!
   5. So the final answer to this question away from what skills I gained is that it always retrieves the factorial of the argument passed, so if I passed 5 for instance, the thing-a-ma-jig program will render factorial 5 which is 120
2. What is the purpose of the variable, facky? Why do we multiply facky by size?
   1. Assuming that my response and analysis in question 1 is correct, facky variable is a must in this program to calculate the factorial of size.
   2. The value of size is being passed when executing line code thingamajig().
   3. If we decided to pass value of 3 as an argument, then size=3; facky=1; Now we will need to get factorial 3 which should be 3\*2\*1, we cannot do this without a helper local variable which in our case here is variable facky
   4. When executing the while loop in function thingamajig, we will notice that an expression facky = facky\*size, and here, all the magic happens.
   5. For this example, I passed value of 3 so there will be two iterations in the while loop to break out.
   6. The first iteration will be facky = 1\*3 the second iteration will be facky = 3\*2 which will return eventually facky as value of 6 which is 3!
   7. I hope I was able to elaborate and demonstrate properly 😊
3. Compare and contrast global variables with local variables in JavaScript. What is the scope of each? Show me an example in code of both a global variable and a local variable.
   1. As for comparing global vs local variables:
      1. Global variables could be used inside any function within the code and it should be defined/ declared in the body of the script tag and outside of the function.
      2. Local variables can only be used in the same function it was identified in (locally scoped).
      3. When a global variable is being used in a function, it will shadow the original value it was identified with.
   2. As for the scope of each:
      1. The variables defined outside a function are globally scoped
      2. The variables defined inside a function are locally scoped.
      3. We should consider identifying a global variable when we know that the same variable will be used in multiple different functions in a JavaScript code to reduce redundancy and avoid re-identifying new variables in all functions.
      4. We should consider identifying a local variable when we need a helper variable that won’t be used outside of the function. The best example for this is the facky variable in function thingamajig.
   3. As for the example in a code:
      1. A good example should be found in the thing-a-ma-jig code
      2. Please read the comments in green in the code below.

<script>

        function clunk(times) {

          //num is a local variable and couldn't be used anywhere else in the code except w/n the clunk() function

          var num = times;

          while (num > 0) {

              display("clunk");

              num = num - 1;

          }

        }

        function thingamajig(size) {

          // facky is a local variable and couldn't be used anywhere else in the code except w/n the same function {thingamajif()}

          var facky = 1;

          clunkCounter = 0;

          if (size == 0) {

                display("clank");

            } else if (size == 1) {

                display("thunk");

            } else {

                while (size > 1) {

                    facky = facky \* size;

                    size = size - 1;

                }

                clunk(facky);

            }

        }

      function display(output) {

          console.log(output);

          //this global variable will shadow the original value since it was  used in this function {display()}

          clunkCounter = clunkCounter + 1;

        }

        // this is a global variable used in the display function as well as in the thingamajif function

        var clunkCounter = 0;

        thingamajig(7);

        console.log(clunkCounter);

    </script>

1. What happens to the scope of a variable if you leave off “var” when declaring it?
   1. JavaScript will deal with it as a global variable (stated in lecture at “23:50).
   2. Here is an example of a code:

Graphical user interface, text, application, chat or text message

Description automatically generated

* 1. We should pay close attention to variable z, it is being added inside function {trial()} however when I console logged z, you will notice that it returned 9 which is the expected value

1. What gets returned from a function without a return statement?
   1. As stated in the lecture (at 12:40), if I don’t return a value in a function, it will be a procedure which means it performs a code out of order but it doesn’t return anything back in terms of value. it doesn’t give a value back to the function itself.
   2. So in order to pass a value back and put it into memory once more in the main program, then a return statement should be included.
   3. And if I want to see the output without a return statement, the function should include console.log(); or document.window() which means it will output but not return anything.
2. What happens to the global and local variables when you press the reload button in the browser?
   1. Stated in the textbook page 108, Reloading a page is like starting over from scratch as far as the variables are concerned. And if any code was in the middle of executing when you reload the page, any local variables will disappear.
   2. So global variables will be in their initial value.
   3. Local variables will disappear and being re-evaluated based on the code itself.
3. What happens when a local variable shadows a global variable?
   1. Stated in the lecture (24:36), the scope global will become local and the local will gain focus from the global.
   2. And it is important to know that the local and global variables have no effect on each other, if you change one, it has no effect on the other. They are independent variables.
4. Is it dangerous to use all global variables in a program? Explain.
   1. Stated in the textbook page 108 that global variables are often being overused in JavaScript as JavaScript doesn’t enforce a lot of structure or overhead on you. However, the downside is when writing serious code this way which is declaring everything as global variables, and it has to be changed and maintained over the long term. So it is highly recommended to organize the code in a modular way.
   2. So it is not dangerous but not recommended either.
5. What happens to the third argument when you pass 3 arguments into a function with 2 parameters?
   1. As stated in the lecture at (5:45), the 3rd argument will just get dropped, so JavaScript will take the first 2 and leave the 3rd one.
6. When would you use a function that has no parameters?
   1. As stated in the very beginning of the lecture, the purpose of functions or being functional is to avoid redundancy and stop re-writing code over and over again.
   2. Few scenarios you might will need to use/ define a function that has no parameters which for instance if the function returns a prompt with hard coded value such as “Thank you”. Now, we can call this function every time we want to send a Thank you message, instead of coding prompt(“Thank you ……”) we can just call the function.
   3. It is important to know few built-in functions could be used without passing any parameters such as (Math.random() OR Math.PI).