Luis Mejia
CIS-14A-48590
Written Assignment 10

Answer the following questions. Please submit the answer with the question in the document. Submit as a word or a PDF file.

Questions 7 and 8 are worth twice the normal points (5 points each).

1. Show me an example of code using a function expression. Explain how this is different than the standard function declaration.

```
// Function expression example

var square = function(num) {

 return num * num;

}

// Function declaration example

function square(num) {

 return num * num;

}
```

Function expressions as seen in the first example in the code above, are different from the standard function declarations in that the browser will look for function declarations first before anything else and then execute code in the order it is presented. This means that functions we declare can be called *before* OR *after* they are defined in our program since the browser creates them along with a variable to hold their reference *before* executing any code. With function expressions on the other hand, we must define and create the function *before* we call/ invoke them because they are evaluated in the order they are presented in our program. Another key difference is that with function expressions, we assign the value of the function(its address/ reference) to a variable with the assignment operator (=) ourselves whereas function declarations take care of assigning the reference to a variable themselves.

2. When a browser parses a page, before it evaluates any code, it looks for _____ .

When a browser parses a page, before it evaluates any code at all it looks for *function declarations*. After the browser scans/parses the page and has found function declarations (if any), it will create a function and assign the resulting reference to a variable with the name you used to declare the function. Afterwards, the browser will start back at the top of the page and will begin to execute the code as it normally would from top to bottom.

Luis Mejia
CIS-14A-48590
Written Assignment 10

3. Function declarations are evaluated _____ the rest of the code is evaluated.

Function declarations are evaluated by the browser with the rest of the code BUT are created before evaluating/executing any code within the program. The creation of functions that are declared before executing any code is called hoisting in Javascript which is its default behavior of moving all declarations to the top of the current scope.

4. Can you hold function references in variables?

Yes, you can hold function references in variables. In fact, with both function declarations and expressions, you get a reference to a function. Variables can hold only a single value or a single reference to a function or an object such as an array. They are too small to hold a whole function or a whole array, therefore, it is ideal for them to hold their address in memory and point to those entities instead.

5. Why is a function declaration not an expression?

A function declaration is not an expression because we are not assigning a function as a value to a variable. In other words, because we use the assignment operator when we write function expressions, we are technically saying that the function expression will evaluate to something. In this case, the value will not be the function as a whole but be a reference to the actual function.

6. You need to start thinking of a function as a _____, just like other objects and primitive types.

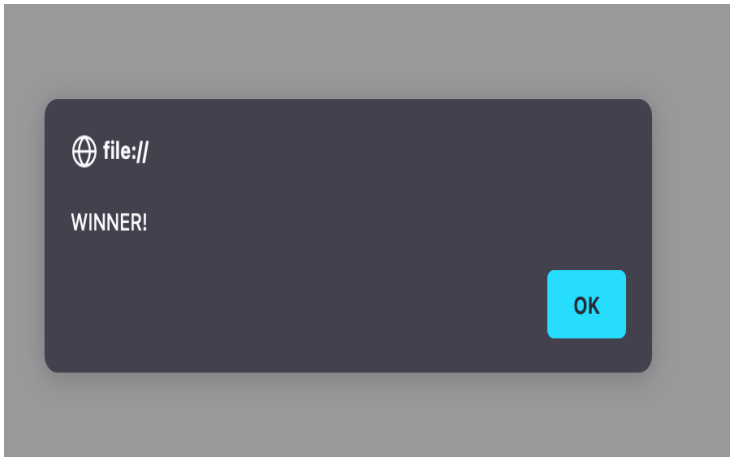
Just like other objects and primitive types, you need to start thinking of functions as *values* as well. The only difference is that this results in a value we can invoke. Whether you define a function with an expression OR declaration, you get a reference (value we can invoke) pointing to the function.

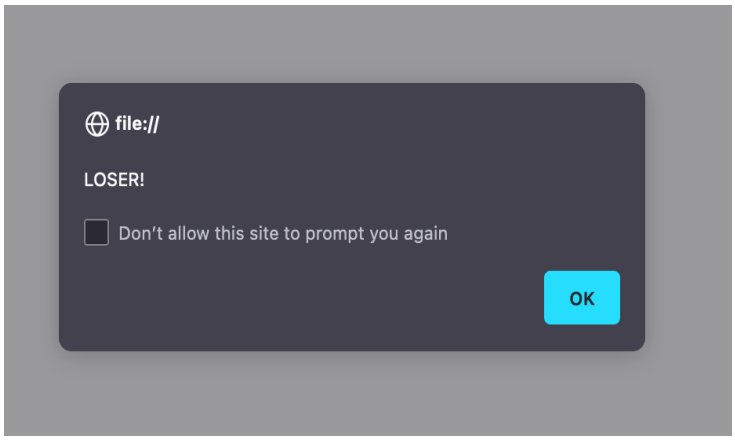
Luis Mejia
CIS-14A-48590
Written Assignment 10

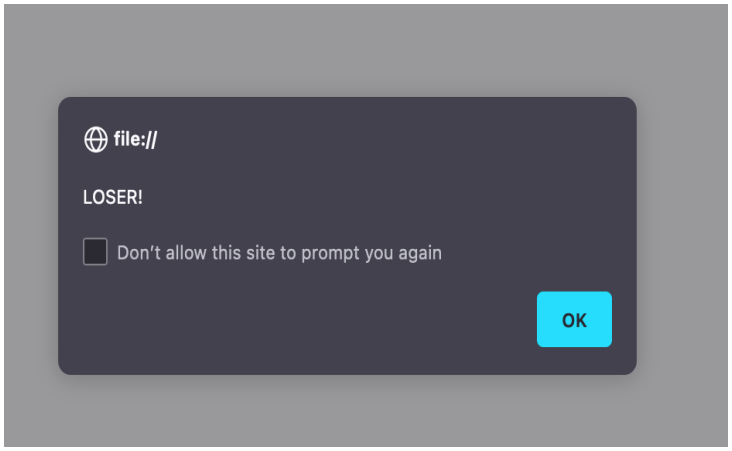
7. Implement the Shell Game on page 440 (Sharpen Your Pencil). Show screenshots and code.

```
// Function declarations

var winner = function() { alert("WINNER!") };

var loser = function() { alert("LOSER!") };

// Assign references to other variables

var a = winner;

var b = loser;

var c = loser;

// Lets try our luck

c = a;

a = b;

b = c;

c = a;

a = c;

a = b;

b = c;

// Invoke

a();

b();

c();
```

Luis Mejia
CIS-14A-48590
Written Assignment 10







Luis Mejia
CIS-14A-48590
Written Assignment 10

8. Implement the array sort method program on pages 459 and 460. Change the numbers that are sorted; add a bigger range of numbers than what's in the book. I want the number array that is sorted to be at least 20 numbers. I need to see screenshots and code.

```
// List of numbers to be sorted
var numbersArray = [456, 92503, 711, 10002, 178000, 1994, 2000, 2021, 911, 2012,
                    1, 27, 35, 9000, 42, 951, 123456789, 99, 100, 145];
// This function compares two values at a time
function compareNumbers(num1, num2) {
 if(num1 > num2) {
   return 1;
 } else if(num1 === num2) {
   return 0;
 } else {
   return -1;
 }
}
// Sort numbers array
numbersArray.sort(compareNumbers);
// Print sorted array to console
console.log(numbersArray)
```