1. What is hoisting in JavaScript? Can you think of any good practice to stay away from hoisting?
2. What is JavaScript window and document objects? How are they related?
3. What is difference between following two statements?

```
setTimeout(booyah, 2000);

setTimeout(booyah(), 2000);
```

4. What problems associated with `var` are solved by `let` in ES6.
5. How `const` like let and how is it different?
6. Which of the following statements are true?
    a. Variable declared with `let` should be assigned immediately.
    b. Object reference assigned to `const` can be changed.
    c. Properties of an object assigned to `const` can be changed.
7. How can you make JavaScript interpreter run in strict mode? Why is function scope strict mode better option?
8. What will the alert after execution of following code?

```
var foo = 1;
function bar() {
      if (!foo) {
            foo = 10;
      }
      alert (foo);
      var foo = 5;
}
bar();
```

Note: Debugging JS?

See next page.

9. Write expected output of following code on paper

```
var x = 1;

function foo() {
    console.log(x);
    var y = 2;

    function foo1() {
        var y = 3;
        if (!x) {
            z = 4;
        }
        var x = 5;
        console.log(x + y + z);
    }

    if(y==2){
        var x = 3;
    }
    //console.log(z);
    foo1(); /* Comment this call and see the changes */
    console.log(x);
    console.log(y);
    console.log(z);
}

console.log(x);
//console.log(y);
//console.log(z);
foo(); /* Comment this call and see the changes */
console.log(x);
//console.log(y);
//console.log(z);
```

a. Run and compare the results with your expected output, make sure you understand the outputs.
b. Now uncomment commented parts **one at a time** and you will notice you won't see all the outputs, it will fail at the line you just uncommented, see the console for reason of failure and make sure you understand why.

c. Next, comment the call to inner function **foo1()** and examine the result.

See next page

10. Figure out the output of following lines of code:

```
var x = 1;
var y = 2;
var z = 3;

function foo() {
    function foo1() {
        console.log(`value of x: ${x}`);
        console.log(`value of y: ${y}`);
        console.log(`value of z: ${z}`);
    }

    var z = 4;
    foo1();
    var x = 3;

    console.log(`value of x: ${x}`);
    console.log(`value of y: ${y}`);
}
console.log(`value of y: ${y}`);
foo();
console.log(`value of x: ${x}`);
console.log(`value of z: ${z}`);
```