Week 1 Graded Questions

1. Which of the following keywords can be used to declare a variable with global scope? [MCQ : 1 point]

A. let
B. var
C. const
D. All of the above

Answer: D

Solution: All of these can be used to create variables with global scope, if a variable is declared with them outside a function, which is not limited to any code block.

2. Which of the following statements is true regarding JavaScript language? [MSQ : 1 point]

A. JavaScript is a statically typed language.
B. JavaScript is a dynamically typed language.
C. JavaScript is a weakly typed language.
D. JavaScript is a strongly typed language.

Answer: B and C

Solution: JavaScript language is a dynamically typed language because it doesn't require the data type of the variable is to be specified, when it is declared. It is also a weakly typed language, as it allows various implicit conversions, and provides a lot of flexibility.

3. Which of the following statement(s) is/are correct regarding JavaScript? [MSQ : 1 point]

A. It is a forgiving language like HTML.
B. It inserts a semicolon after each statement automatically.
C. Blocks and functions both use { } for body contents.
D. Blocks use { } but functions use indentation for body contents.

Answer: A, B and C

Solution: JavaScript is considered a forgiving programming language, as it gives a lot of flexibility to the user like coercion etc. JavaScript doesn't require a semicolon to be put at the end of each statement explicitly, as it does it on its own. Both the blocks and functions make use of curly braces ({ }) to define their body or scope, unlike Python, which uses indentation.

4. Consider the following program, and predict the last value shown on the console? [MCQ : 3 points]

```
const timer = (time) => {
  let x = 1
  let handlar = setInterval(() => {
    console.log(2 * x)
    x++
  }, 1000)

  setTimeout(() => {
    clearInterval(handlar)
  }, time)
}

timer(5000)
```

A. 4
B. 8
C. 5
D. 10

Answer: B

Solution: In the program given above, setInterval() function is triggered after every 1 second (1000 milliseconds), and will terminate after waiting for 5 seconds. The first value that will be shown on the console is 2 (after waiting for 1 second), then 4 (after waiting for 1 more second), then 6 (after waiting for 1 more second), then 8 (after waiting for 1 more second). The setTimeout() function will be executed after waiting for 5 seconds. Thus, the last value shown on the console is 8.

5. Consider the following code snippet,

```
const x = { name: 'rohit' }
x.name = 'Mohit'
```

What will be the output of console.log(x.name)? [MCQ : 2 points]

A. It will throw a TypeError.
B. Rohit
C. Mohit
D. undefined

Answer: C

Solution: Using a const keyword while creating or declaring an object will make the reference constant. However, the properties of the object (declared with const) can still be changed.

6. What will be the output of the following program? [MCQ : 2 points]

```
const obj = {
  name: 'Rohit',
  changeName: (name) => {
    this.name = name
  },
}

obj.changeName('Mohit')
console.log(obj.name)
```

A. undefined
B. Rohit
C. Mohit
D. null

Answer: B

Solution: Arrow function does not have “this” of its own. Arrow functions establish "this" based on the scope the Arrow function is defined within. So, the “name” property of the “obj” object will not change. It will remain “Rohit”.

7. What will be the output of the following program? [MCQ : 3 points]

```
const obj = {
  name: 'Rohit',
  arrowFunction: null,
  normalfunction: function () {
    this.arrowFunction = () => {
      console.log(this.name)
    }
  },
}

obj.normalfunction()
obj.arrowFunction()
```

A. Rohit
B. Mohit
C. undefined
D. Syntax Error

Answer: A

Solution: Arrow function does not have “this” of its own. Arrow functions establish "this" based on the scope the Arrow function is defined within. So, the value of “this” will be the “obj” object. Because arrowFunction is defined inside the normalFunction so it will inherit the value of “this” from normalFunction.

8. Which of the following is correct output for the following code snippet? [MCQ : 2 points]

```
setTimeout(() => console.log('hello from setTimeOut one'), 0)
console.log('hello for main one')
setTimeout(() => console.log('hello from setTimeOut two'), 0)
console.log('hello from main two')
```

A. 'hello from setTimeOut one'
'hello for main one'
'hello from setTimeOut two'
'hello from main two'

B. 'hello for main one'
'hello from setTimeOut one'

'hello from main two'
'hello from setTimeOut two'

C. 'hello for main one'
'hello from main two'
'hello from setTimeOut two'
'hello from setTimeOut one'

D. 'hello for main one'
'hello from main two'
'hello from setTimeOut one'
'hello from setTimeOut two'

Answer: D

Solution: setTimeout() is an asynchronous function. So, the callback function will be called once the call stack is empty, i.e. first the synchronous codes will be executed and then asynchronous codes.

9. Consider the following HTML document,

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <script>
      setTimeout(() => {
        document.getElementById('div1').style.backgroundColor = 'yellow'
      }, 0)
    </script>
    <title>Document</title>
  </head>
  <body>
    <div
      id="div1"
      style="height: 100px; width: 100px; background-color: red"
    ></div>
  </body>
</html>
```

What will be the background color of an element having ID ‘div1’ in the rendered webpage corresponding to this document? [MCQ : 2 points]

A. red
B. black
C. yellow
D. None of the above

Answer: C

Solution: setTimeout() is an asynchronous function. So, the callback function will be called once the call stack is empty, i.e. first the synchronous codes will be executed and then asynchronous codes. Till the time, the statement document.getElementById('div1').style.backgroundColor = 'yellow' will be executed the DOM is already built.

10. Consider the following JavaScript program,

```
let startNamePrinter = (name) => {
  let x = name.split('').reverse()
  let handler = setInterval(() => {
    let y = x.pop()
    console.log(y)
  }, 1000)

  setTimeout(() => {
    clearInterval(handler)
  }, (name.length + 1) * 1000)
}

startNamePrinter('orange')
```

What will be the last value shown on the console after 4 seconds? [MCQ : 3 points]

A. r
B. a
C. n
D. o

Answer: C

Solution: Callback function defined as the parameter of setIntervel will be executed after each 1 second. After every 1 second, it will pop one character from the string and logs it on the console. So, after 4 seconds, it will log character ‘n’.