**Test Paper JavaScript, ES 6 and Node.js**

1. Which method of the string object is used to divide the string into separate strings?

Ans – split() is used to divide string into an array of substrings based on delimiter

substr() is used to create a substring without modifying original string.

1. Which methods of the array is used to iterate over the array (mentioned all those are used in iteration)?

Ans – 1)We can iterate by looping via for() or while()

2)forEach() is also used to iterate, but it can’t use ‘break’ statement

3)map() may also be used.

1. Which method of array is used to remove the first element from array?

Ans – shift() is used to pop from front.

1. Which method is used to add a new element at the first position in array?

Ans – unshift() is push from front.

1. What is the difference between forEach() and map() method of array?

Ans – forEach() doesn’t return a value, it is just used for processing.

map() – for every element it creates a new array with the result returned

By the calling function.

1. What is the new concep?t provided in ES 6 for concatenation of string

Ans – besides using the + operator to concat, template string are also provided.

If one of the operands of the + operator is a string then the result will also be

In string type.

1. When should we use Arrow (=>) operator?

Ans - We use arrow operator mainly when a function is supposed to return a value

Ie callback function

1. How to read multiple HTML UI elements using the ‘class’ attribute?

Ans – document.getElementsByClass(ClassName) will return an array of elements

Having same class = ClassName.

1. If the HTML page is having the ‘window.onload’ event and the JavaScript file reference that is using Immediately Invoked Function Expression (IIFE) then whart will be executed first onload event or IIFE?

Ans – IIFE will be executed first.

1. How to define public method in function reference object in JavaScript?

Ans - var func = function(){//body}

1. When should we use rest parameters?

Ans – we use rest parameters when the number of parameters to be passed are not determined at compile time and are varying.

1. What will be the output of the following JavaScript Code

Ans - 0 1 2 3 4 5 6 7 8 9 read the sequence 10

|  |
| --- |
| function\* myGenerator(start=0,end=100, step=1){  let count= 0;  for(let i=0; i<end;i++){  count++;  yield i; // read the object and advance to next  }  return count;  }  const generator = myGenerator(1,10,1);  let res = generator.next();  while(!res.done){  console.log(res.value);  res = generator.next();  }  console.log(`Read the Sequence ${res.value}`); |

1. What is the difference between generators and iterators?

Ans – Iterators are used to traverse while generators are a mechanism to simplify iterators

1. Which collection will be used to store data in Key/ Value pair?

Ans – maps

1. What will be the output of the following code – Ans = Size of setObjects = 3

|  |
| --- |
| let setObjects = new Set();  setObjects.add({id:101, name:'A'});  setObjects.add({id:102, name:'B'});  setObjects.add({id:103, name:'C'});  setObjects.add({id:101, name:'A'});  console.log(`Size of setObjects = ${setObjects.size}`);  console.log(setObjects); |

1. What will be the output of the following code? – Ans = true,false,true

|  |
| --- |
| var x = 123; // number  var y = "123"; // string  console.log(x == y); // returns 'true' value equality uses parser to read values instead of type of value  console.log(x === y); // return 'fasle', === is type and then value base equaity aka 'deep equality'  console.log(x !== y); // return true, !== not equal to |

1. What will be the result of the following code?

Ans – After updating obj1.x to 900, the value of obj1.x=900 and obj.x=900

|  |
| --- |
| var obj = {x:10};  console.log('obj.x = ' + obj.x);  var obj1 = obj; // Same reference pointers  console.log('After equating obj.x to obj1, the obj.x = ' + obj.x + ' and obj1.x = '+ obj1.x);  obj1.x = 900;  console.log('After updating obj1.x to 900 , the value ob obj1.x = ' + obj1.x + ' and value of obj.x = ' + obj.x); |

1. What will be the result of the following code?

Ans – inside loop I = 0123456789

Outside loop I = 10

Outside loop after increment = 11

|  |
| --- |
| for(var i=0;i<10;i++) {  console.log('Inside loop i = ' + i);  }  console.log('Ouside loop i = ' + i);  i++;  console.log('Ouside loop after increament i = ' + i); |

1. What will be result of the following code?

Ans - values array will be sorted lengthwise ascending and displayed then in

Descending order lengthwise and displayed.

|  |
| --- |
| let values = ["Mahesh", "Suprotim", "Vikram", "Subodh", "Sumit", "Pankaj", "Saket", "Manish", "Abhijit", "Kumarmangalam"];  let sortByLengt = values.sort((v1,v2)=>{  return v1.length - v2.length;  });  console.log(JSON.stringify(sortByLengt));  sortByLengt = values.sort((v1,v2)=>{  return v2.length - v1.length;  });  console.log(JSON.stringify(sortByLengt)); |

1. What is the new iteration loop introduced by ES 6 to iterate over the array?

Ans – the for-of loop is used to iteratre.

1. What will be result of the following code?

Ans – [10,20,30]

[10,20,30,40,50,60,70,80]

|  |
| --- |
| let arr = [10,20,30];  console.log(arr);  arr = [...arr,40,50,60,70,80];  console.log(arr); |

1. In node.js what is the use of @types/node package?

Ans – Intellisense and type definitions for node

1. What is the use of **require()** object in Node.js?

Ans – used to include dependencies from various modules

1. What is the role of **module.exports** in Node.js?

Ans – used to export the dependency ie the module to be imported from some other code.

1. What is the use of **request.url**  property of the Request object of the **createServer()** method of the http module?

Ans – used to get url from request to extract paramters from it

1. Which of the Request object of the **createServer()** method of the http module is used to check if the request get or post?

Ans – req.method ===”GET”, req.method ===”POST”

Where req is request object

1. When should we use **async** methods in Node.js or in ES 6?

Ans – When the method is asynchronous or returns promises ie Acknowledgement from server then we use async methods

1. Can we execute awaitable state in method that is node marked as async?

Ans – Yes, as await only works inside async methods

1. What is the advantage of the **q** module on Node.js?

Ans – use for asynchronous promise resolution

1. What is the use of http.request() method in Node.js?

Ans – request method is used to receive request body,headers etc from the api calls

1. What method of the Sequelize’s Model object is used to delete the record from table?

Ans – destroy()

1. What method the Sequelize’s Model object is used to add a new record in table?

Ans – create()

1. What method the Sequelize’s Model object is used to search record from table based on condition?

Ans- findOne({where:{}})

1. What is the difference between AUTO\_INCREMENT and Primary key in MySQL?

Ans – when AI field is not specified then it is incremented based on previous value of record. Primary key is unique and not null

1. Can we use AUTO\_INCREMENT key as Primary key?

Ans – We can auto increment on primary key.

1. How to modify the column in MySQL table write a small script?

Ans – alter table tablename set columnName = newValue where…….

1. What command is used to generate the JavaScript Models from MySQL database using Sequlize?

Ans - sequelize-auto -h localhost -d dbName -u root -x P@ssw0rd\_ --dialect mysql -t tableName

1. What is the mandatory package must be installed along with Sequlize and sequlize-auto to generate JavaScript models from MySQL database?

Ans – body-parser,express,mysql2

1. What is the use of DataType class in Sequlize?

Ans – used to set column data type (ex int, varchar etc)

1. What is the use of Sequlize.query() method?

Ans – used to execute SQL queries.