(Q) What is java script?

Java script is a scripting language mainly used to make any website dynamic and interactive. It is the most used programming language in the world. Used as a client-side programming language.

(Q) What is the use of isNAN function?

The isNAN() function accepts a value and determines whether the given value is a number or not. If so, this method returns true else it returns false. You can also call this method using number object.

(Q) What is negative infinity?

The Negative infinity is a constant value that is used to represent a value lowest of all. This means there is no other value lesser than this value. Negative infinity is a special numeric value that is returned when arithmetic operators or mathematical function generates a negative value greater than the largest representable number in java script.

(Q) Which company developed java script?

Java script was invented by Bredan Eich in 1995. It was developed for natscape2. And became the EACMA 262 standard in 1997. After Netscape handed java script for the firebox browser.

(Q) What are the undeclared and undefined variable?

Undeclared variables are those variables which are not written with let, var or const, if we access them in the code execution phase them will throw reference error.

Example:

Undeclared:console.log(y)// where y is not declared above undefined variable are those which are declared with let, var or const but ain’t assign with value. If we try to access them in the code execution phase them will throw undefined as value.

Example-

Input-

<button onclick="a()">click</button>

    <p id="one"></p>

    <script>

        function a() {

            if (typeof variable === "undifined") {

                string = "Variable is undefined";

            }

            else{

                string = "variable is defined"

            }

            document.getElementById("one").innerHTML=string;

        }

    </script>

Output-



(Q) Write the code for adding new eliments dynamically?

Input-

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <p id="firstp">first p</p>

    <script type="text/javascript">

          function addNode() {

            var newP = document.createElement("p");

            var textNode = document.createTextNode("This is a new text node");

            newP.appendChild(textNode); document.getElementById("firtp").appendChild (newp);

          }

    </script>

</body>

</html>

(Q) What is the difference between ViewState and SessionState?

|  |  |
| --- | --- |
| ViewState | SessionState |
| Maintained at page level only | Maintained at session level. |
| ViewState can only be visiable from a single page and not multiple pages. | Session stat value availability in user session |
| Information is stored on the client’s end only | Information is stored on the server. |
| ViewState value are cleared when new page is loded | Session state can be cleared by programmer or user on is case of timeouts. |

(Q) What is === operators?

In javascript === is strict equality operator. Used to compare two variable and check both value and datatype. If both datatypes and value matches of two variables it will return Boolean result (true or False).

Input-

<script>

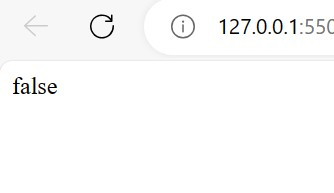
        var a = 10

        var b = "10"

        document.write(a === b)

    </script>

Output-



(Q) How can the style/class of an element be changed?

In this example, using the class name property to change the color of the element.

Input-

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <style>

        .one{

            color: blue;

        }

        .clolrblack{

            color: black;

        }

    </style>

</head>

<body>

    <p class="one">Lorem ipsum dolor sit amet consectetur adipisicing elit. Omnis asperiores nemo sint maxime expedita repellendus, aspernatur quam nobis illo! Cum rerum qui adipisci aliquid vero cupiditate ad? Alias, amet sequi?</p>

    <button id="click">click</button>

    <script>

        const button = document.getElementById("click");

        const para = document.querySelector("p");

        button.addEventListener("click",function() {

            para.className= "color black"

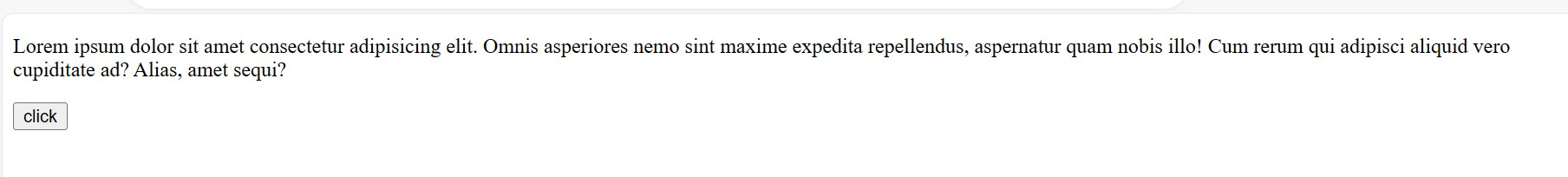
        })

    </script>

</body>

</html>

Output-



(Q) How to read and write a file using javaScript?

There are two ways to do it.

1. Using javascript extension (runs from javascript editor)
2. Using a webpage and Active objects (internet Explorer only)

(Q) What are all the looping structure in javascript?

1 for – loops through a block of code a number of times

2 for/in -loops through the properties of an object

3 for/of- loops through the value of an iterable object

4 while – loops through a block of code while a specified condition is true.

(Q) How can you convert the string of any base to an integer in javascript?

To convert a string to integer parseInt(). And unary operators (+) function is used in javascript, parseInt() function returns Nan (not a number) when the string doesn’t contain number. If a string with a number is sent, then only that number will be returned as the output.

(Q) What is the function of the delete operator?

The delete function is used on object properties.

(Q) Write a program to show an alert.

document.write (alert(“This is Tops tech”));

(Q) What will be the result for these expression?

1. 5>4

True

2.”Apple” > “pineapple”

False

3.”2” > “12”

True

4. undefined == null;

True

5. undefined === null;

False

6. null == “\n0\n”

False

7. null === “\n0\n”

False

(Q) Will alert be show? If (“0”) {alert(‘Hello’};

Yes, if we put “0” in condition it is work as a string and code under this condition would execute. But if we put 0 in condition it is work as a number and code under this condition would never execute.

(Q) What is the code below going to output? Alert (null || 2 || undefined);

Result will be 2 . Because null and undefined refers to empty string.

(Q) The following function returns true if the parameter age is greater than 18. Otherwise it asks for a confirmation and returns its result.

Function checkage (age) {

If (age >18 )

{return true;}

Else

{ return confirm (‘did parents allow you ?);}

}

(Q) Replace Function Expressions with arrow function in the code below:

Function ask (question , yes , no)

{  
 if (confirm(question))yes();

Else no ();

}

ask (“Do you agree?”,

function () {alert(“You agreed .”);},

function () {alert(“you canceled the execution.”);}}

Function ask (question , yes , no)

{

If (confirm (question)) yes ();

Else no ();

?

Ask(“do you agree?”,

() =>{alert(“you agreed.”);},

() => {alert(“You canceled the execution.”);})

Module 2 : (Data Type and Objects)

(Q) write the code, one line for each action:

a) create an empty object user.

Let person = new Object();

b) Add the property name with the value john.

Person.name = “john”;

c) Add the property surname with the value smith.

Person.surname = “smith”;

d) Change the value of the name to pete.

Person. Name = “pete”;

e) Remove the property name from the object.

Delete person.name;

(Q) Is array copied? let fruits = ["Apples", "Pear", "Orange"]; // push a new value into the "copy" let shoppingCart = fruits; shoppingCart.push("Banana"); // what's in fruits? alert( fruits.length ); // ?

= Yes array is copied

Fruits = [“Apple”, “pear”, “Orange”, “Banana”]

The fruits length will be 4.

(Q) Map to names let john = { name: "John", age: 25 }; let pete = { name: "Pete", age: 30 }; let mary = { name: "Mary", age: 28 }; let users = [ john, pete, mary ]; let names = /\* ... your code \*/ alert( names ); // John, Pete, Mary

= let john = { name: ”john”, age:25}:

let pete = {name: “pete” , age:30};

let mary = {name: “Mary”, age:28};

let users = [john,pete,marv];

let names = users.map (( item) => {item.name});

alert(names);

(Q) • Map to objects let john = { name: "John", surname: "Smith", id: 1 }; let pete = { name: "Pete", surname: "Hunt", id: 2 }; let mary = { name: "Mary", surname: "Key", id: 3 }; let users = [ john, pete, mary ]; let usersMapped = /\* ... your code ... \*/ /\* usersMapped = [ { fullName: "John Smith", id: 1 }, { fullName: "Pete Hunt", id: 2 }, { fullName: "Mary Key", id: 3 } ] \*/ alert( usersMapped[0].id ) // 1 alert( usersMapped[0].fullName ) // John Smith

= let usersMapped = users.map((user) =>({

fullname : ${user.name} ${user. Surname}’,

id : user.id

}));

(Q) Sum the properties There is a salaries object with arbitrary number of salaries. Write the function sumSalaries(salaries) that returns the sum of all salaries using Object.values and the for..of loop.If salaries is empty, then the result must be 0. let salaries = { "John": 100, "Pete": 300, "Mary": 250 }; alert( sumSalaries(salaries) ); // 650

= let salaries = {

“john”: 100,

“pete”:300,

“mary”:250,

};

Function sumSalaries(salaries){

Var sum = 0 ;

For (const salary of Object .values(salaries)){

Sum += salary;

}

return sum;

}

alert(sumSalaries(Salaries));