

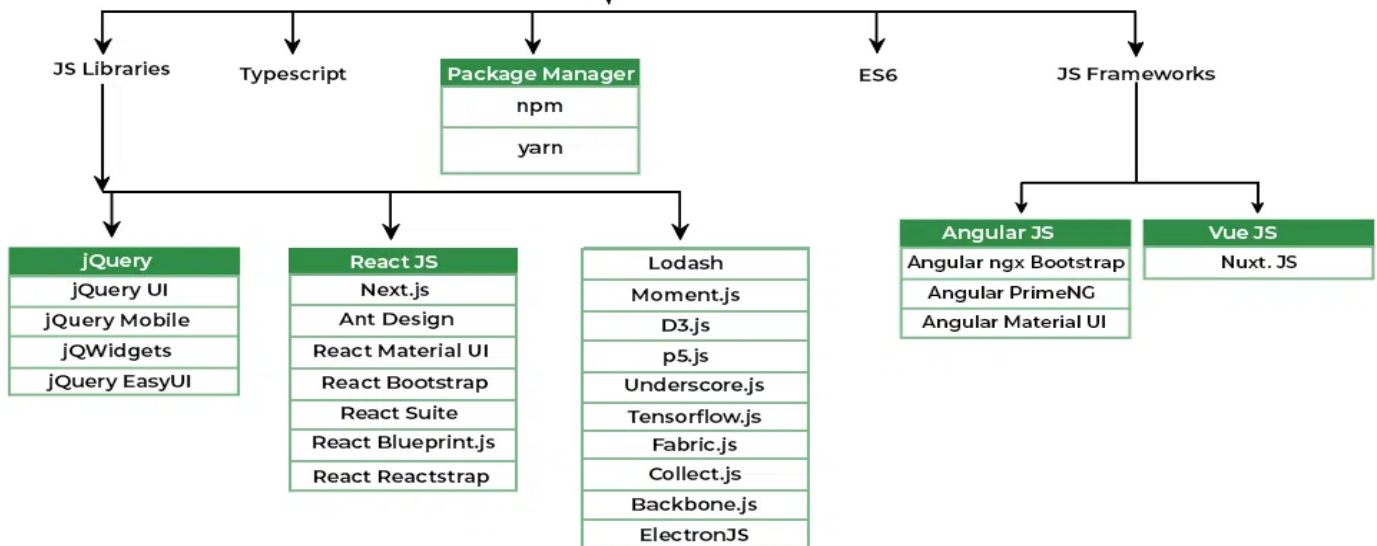
# ASSIGNMENT 6:- JAVASCRIPT

Que.:- 1) What is JavaScript?

Ans.:- JavaScript (JS) is the most popular lightweight, interpreted compiled programming language. It can be used for both Client-side as well as Server-side developments. JavaScript also known as a scripting language for web pages. This JavaScript Tutorial is designed to help both beginners and experienced professionals master the fundamentals of JavaScript and unleash their creativity to build powerful web applications. From basic syntax and data types to advanced topics such as object-oriented programming and DOM manipulation.



## JavaScript



Que.:- 2) What is the use of isNaN function?

Ans.:-

- In JavaScript NaN is short for "Not-a-Number".
- The isNaN() method returns true if a value is NaN.
- The isNaN() method converts the value to a number before testing it.
- isNaN() method returns true if a value is Not-a-Number.
- Number.isNaN() returns true if a number is Not-a-Number.
- isNaN() converts the value to a number before testing it.

```
<!DOCTYPE html>
<html>
<body>
<h2>The isNaN()</h2>
<p>"Good Morning!"</p>
<p id="demo"></p>
<script>
let text = "Hello";
document.getElementById("demo").innerHTML = isNaN(text);
</script>
</body>
</html>
```

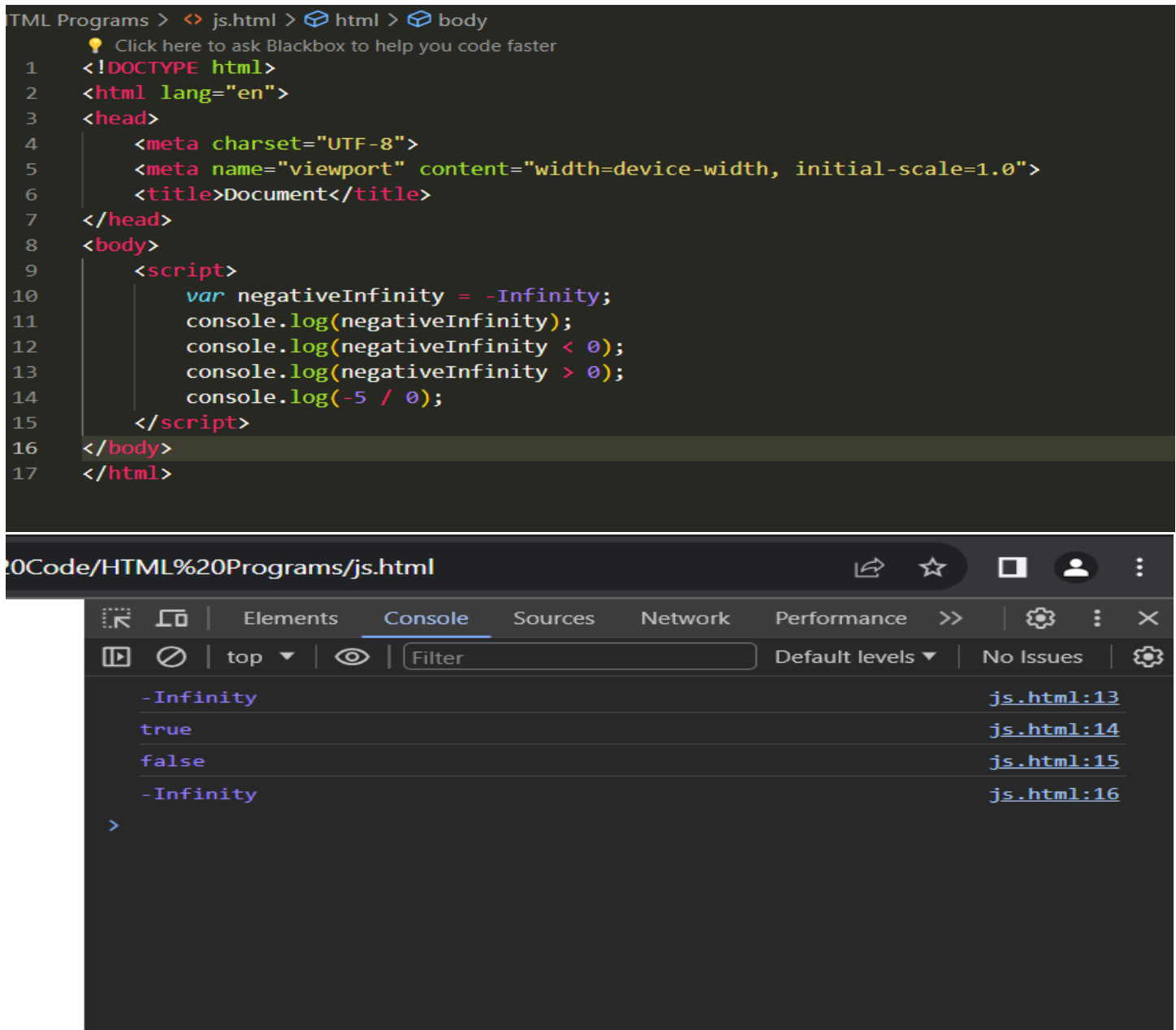
### The isNaN()

Is "Good Morning!"

true

Que.:- 3) What is negative Infinity?

Ans.:- In JavaScript, Infinity is a special value that represents positive infinity, which is a concept in mathematics indicating an unbounded positive quantity. Conversely, JavaScript also has a special value called -Infinity, which represents negative infinity.



The screenshot shows a VS Code editor window with a file named 'js.html' open. The code is an HTML document with a script block. The script defines a variable 'negativeInfinity' as -Infinity and logs several values to the console. The console output shows the results of these log statements.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7 </head>
8 <body>
9   <script>
10     var negativeInfinity = -Infinity;
11     console.log(negativeInfinity);
12     console.log(negativeInfinity < 0);
13     console.log(negativeInfinity > 0);
14     console.log(-5 / 0);
15   </script>
16 </body>
17 </html>
```

The console output shows the following results:

- Infinity (js.html:13)
- true (js.html:14)
- false (js.html:15)
- Infinity (js.html:16)

Que.:- 4) Which company developed JavaScript?

Ans.:- **Brendan Eich first developed JavaScript, a computer language, in about ten days in May 1995.** The language, formerly known as Mocha, later modified to LiveScript, and is now known simply as JavaScript, was created to be used on the client-side of websites, enabling the addition of dynamic and interactive components to static HTML texts.

**JavaScript was initially implemented in Netscape Navigator, which was the most popular browser at the time.** The language was quickly adopted by Microsoft for use in Internet Explorer. Due to its simplicity of usage and the fact that it was the only client-side scripting language available at the time, JavaScript quickly gained popularity among web developers.

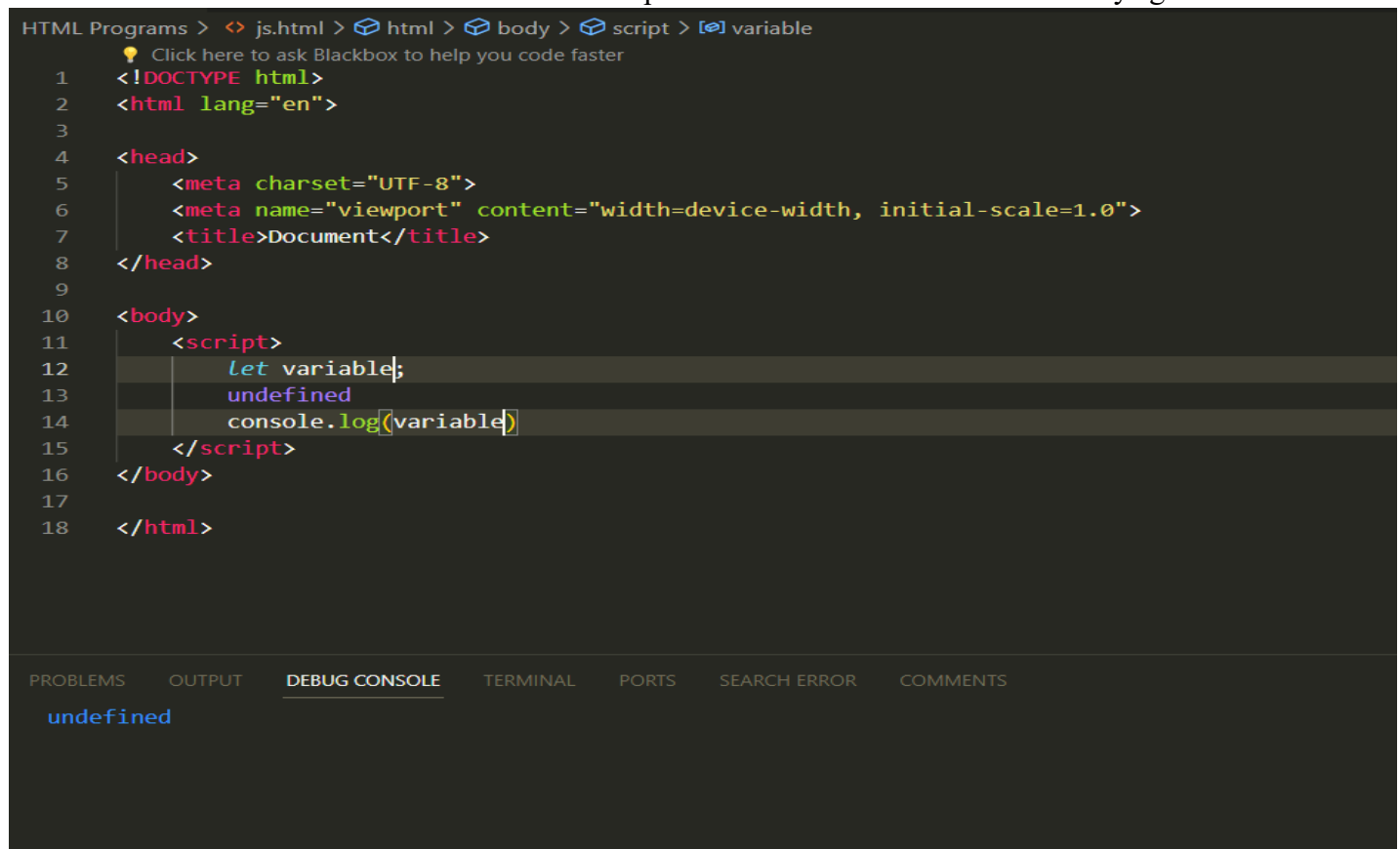
JavaScript gained popularity during the ensuing years and was used to develop a wide range of web applications, such as online games, dynamic menus, and form validation. **ECMAScript 4 a new version of the language, was planned in 2002.** However, it was ultimately abandoned because of conflicts among the various browser vendors.

Que.:- 5) What are undeclared and undefined variables?

Ans.:-

**Undefined:-** It occurs when a variable has been declared but has not been assigned any value. Undefined is not a keyword.

**Undeclared:-** It occurs when we try to access any variable that is not initialized or declared earlier using the var or const keyword. If we use 'typeof' operator to get the value of an undeclared variable, we will face the runtime error with the return value as "undefined". The scope of the undeclared variables is always global.



```
HTML Programs > <> js.html > html > body > script > variable
  Click here to ask Blackbox to help you code faster
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5    <meta charset="UTF-8">
6    <meta name="viewport" content="width=device-width, initial-scale=1.0">
7    <title>Document</title>
8  </head>
9
10 <body>
11   <script>
12     let variable;
13     undefined
14     console.log(variable)
15   </script>
16 </body>
17
18 </html>
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR COMMENTS

undefined

```
js.html
HTML Programs > <> js.html > html > head
Click here to ask Blackbox to help you code faster
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8 </head>
9
10 <body>
11   <script>
12     // Undeclared variable
13     x = 5;
14     console.log(x);
15   </script>
16 </body>
17
18 </html>
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR COMMENTS

5

Que.- 6) Write the code for adding new elements dynamically?

Ans.-

```
HTML Programs > <> js.html > html > body
Click here to ask Blackbox to help you code faster
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8 </head>
9 <body>
10   <script>
11     let array = [1, 2, 3, 4];
12     // Adding elements dynamically
13     array.push(5);
14     array.unshift(8);
15     console.log(array);
16   </script>
17 </body>
18 </html>
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR COMMENTS Filter (e.g. test)

> (6) [8, 1, 2, 3, 4, 5]

Que.:- 7) What is the difference between ViewState and SessionState?

Ans.:-

ViewState	SessionState
Maintained at page level only.	Maintained at session level.
View state can only be visible from a single page and not multiple pages.	Session state value availability is across all pages available in a user session.
It will retain values in the event of a postback operation occurring.	In session state, user data remains in the server. Data is available to user until the browser is closed or there is session expiration.
Information is stored on the client's end only.	Information is stored on the server.
Used to allow the persistence of page-instance-specific data.	Used for the persistence of user-specific data on the server's end.
ViewState values are lost/cleared when new page is loaded.	SessionState can be cleared by programmer or user or in case of timeouts.

Que.:- 8) What is === operator?

Ans.:- The === operator is a strict equality operator in JavaScript. It is used to compare two values for equality without performing any type conversion. This means that not only do the values have to be equal, but they must also be of the same data type.

- If the operands are of the same type and have the same value, === returns true.
- If the operands are of the same type but have different values, === returns false.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!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>
  <script>
    let x = 5;
    let y = "5";
    console.log(x == y);
    console.log(x === y);
  </script>
</body>
</html>
```

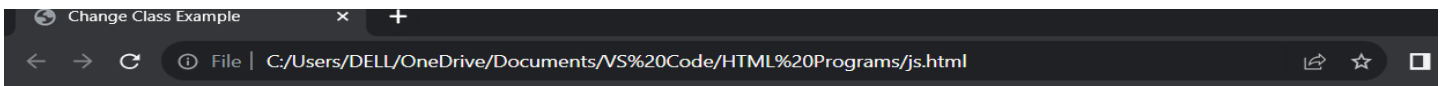
DEBUG CONSOLE

```
true
false
```

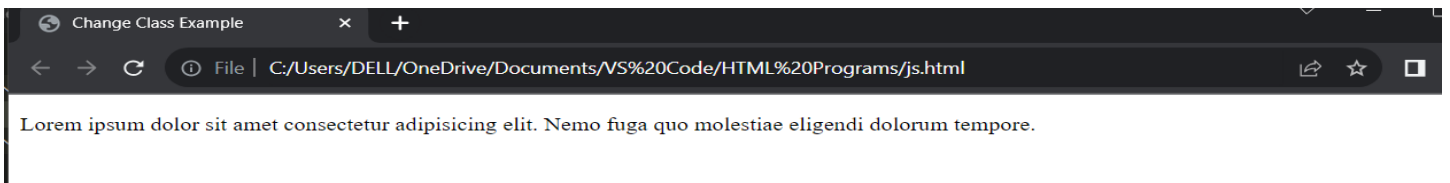
Que.:- 9) How can the style/class of an element be changed?

Ans.:-

```
HTML Programs > js.html > html > head > style > .highlighted
Click here to ask Blackbox to help you code faster
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Change Class Example</title>
7   <style>
8     .highlighted {
9       color: rgb(65, 9, 250);
10      font-weight: bold;
11      font-size: larger;
12    }
13  </style>
14 </head>
15 <body>
16   <p id="myParagraph">Lorem ipsum dolor sit amet consectetur adipisicing elit. Nemo fuga quo molestiae eligendi
17     dolorum tempore.</p>
18   <script>
19     let paragraph = document.getElementById('myParagraph');
20     paragraph.classList.add('highlighted');
21     setTimeout(() => {
22       paragraph.classList.remove('highlighted');
23     }, 2000);
24   </script>
25 </body>
26 </html>
```



Lorem ipsum dolor sit amet consectetur adipisicing elit. Nemo fuga quo molestiae eligendi dolorum tempore.



Que.:- 10) How to read and write a file using JavaScript?  
Ans.:-

```

HTML Programs > js.html > html > body > script
Click here to ask Blackbox to help you code faster
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7  </head>
8  <body>
9      <input type="file" id="fileInput" />
10     <button onclick="readFile()">Read File</button>
11     <script>
12         function readFile() {
13             let fileInput = document.getElementById('fileInput');
14             if (fileInput.files.length > 0) {
15                 let file = fileInput.files[0];
16                 let reader = new FileReader();
17                 reader.onload = function (e) {
18                     console.log("File content:", e.target.result);
19                 };
20                 reader.readAsText(file);
21             } else {
22                 console.log("No file selected.");
23             }
24         }
25     </script>
26 </body>
27 </html>
28

```

127.0.0.1:5500/HTML%20Programs/js.html

Choose File No file chosen

Read File

Que.:- 11) What are all the looping structures in JavaScript?

Ans.:-

- **for loop:-**

The for loop is a common loop that repeats a block of code a specified number of times.

**Ex.:-** for (let i = 0; i < 5; i++) {  
     console.log(i);  
 }

- **while loop:-**

The while loop repeats a block of code if a specified condition is true.

**Ex.:-** let i = 0;  
     while (i < 5) {  
         console.log(i);  
         i++; }  
     }

- **do-while loop:-**

Similar to the while loop, the do-while loop repeats a block of code as long as a specified condition is true. The key difference is that the do-while loop guarantees that the code block is executed at least once, as the condition is checked after the block is executed.

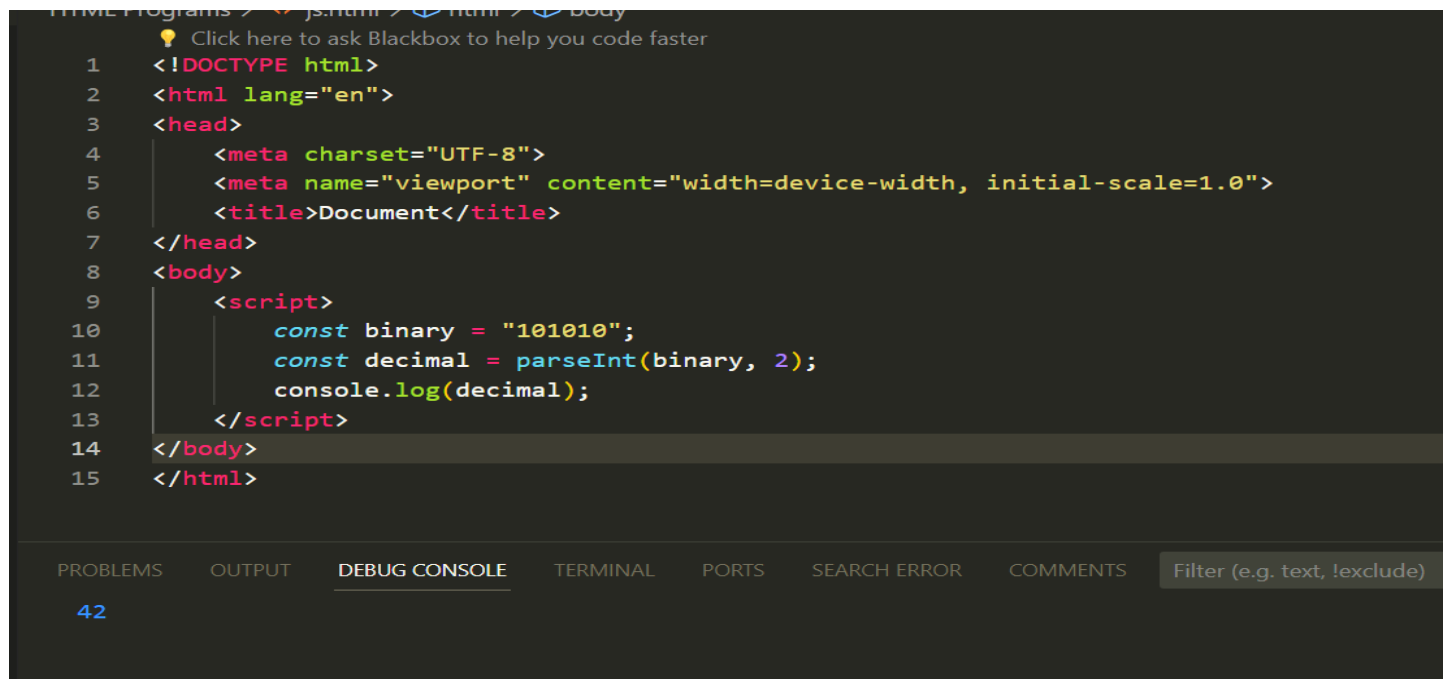
**Ex.:-**

```
let i = 0;
do {
  console.log(i);
  i++;
}
while (i < 5);
```

**Que.:-** 12) How can you convert the string of any base to an integer in JavaScript?

**Ans.:-** You can use the `parseInt` function to convert a string representation of a number in a given base to an integer. The `parseInt` function takes two arguments: the string to be converted and the base of the numeral system.

**Syntax:-** `parseInt(string, radix);`

A screenshot of a code editor interface. The top part shows a file explorer with 'js.html' selected. The main editor area displays an HTML document with a JavaScript snippet. The snippet defines a constant 'binary' with the value '101010', converts it to a decimal integer using 'parseInt(binary, 2)', and logs the result. The bottom of the editor shows tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', 'PORTS', 'SEARCH ERROR', and 'COMMENTS'. The 'DEBUG CONSOLE' tab is active, showing a single log entry '42'.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7 </head>
8 <body>
9   <script>
10    const binary = "101010";
11    const decimal = parseInt(binary, 2);
12    console.log(decimal);
13  </script>
14 </body>
15 </html>
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR COMMENTS Filter (e.g. text, !exclude)

42

**Que.:-** 13) What is the function of the delete operator?

**Ans.:-** The delete operator in JavaScript is used to delete an object's property or an element at a specified index in an array.

**Syntax:-** `delete array[index];`



```
HTML Programs > js.html > html > body > script
Click here to ask Blackbox to help you code faster
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7 </head>
8 <body>
9   <script>
10    const array = [1, 2, 3, 4, 5];
11    delete array[2];
12    console.log(array);
13  </script>
14 </body>
15 </html>
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR COMMENTS

> (5) [1, 2, ..., 4, 5]

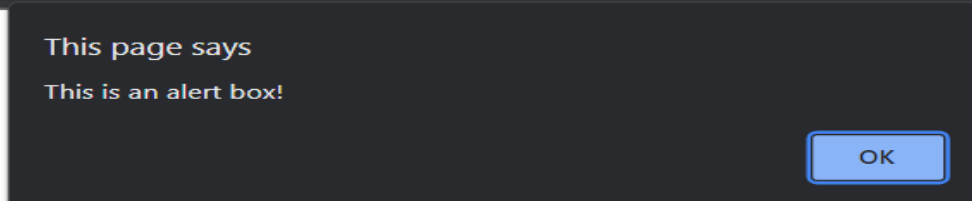
Que.:- 14) What are all the types of Pop up boxes available in JavaScript?

Ans.:-

- **Alert Box:-**

The alert function displays a simple dialog box with a message and an "OK" button. It is often used for informational purposes.

```
HTML Programs > js.html > html > body > script
Click here to ask Blackbox to help you code faster
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7 </head>
8 <body>
9   <script>
10    alert("This is an alert box!");
11  </script>
12 </body>
13 </html>
```



- **Confirm Box:-**

The confirm function displays a dialog box with a message, an "OK" button, and a "Cancel" button. It is used when a user needs to confirm or cancel an action.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7 </head>
8 <body>
9   <script>
10    const result = confirm("Are you OK!");
11    if (result === true) {
12      console.log("confirmed");
13    } else {
14      console.log("canceled");
15    }
16  </script>
17 </body>
18 </html>
```

This page says  
Are you OK!

OK Cancel

- **Prompt Box:-**

The prompt function displays a dialog box with a message, an input field for the user to enter data, and "OK" and "Cancel" buttons. It is used when the user is required to input some value.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7 </head>
8 <body>
9   <script>
10    const Input = prompt("Please enter your name:-", "");
11    if (Input !== null) {
12      console.log("data entered: " + Input);
13    } else {
14      console.log("User canceled the prompt");
15    }
16  </script>
17 </body>
18 </html>
```

This page says

Please enter your name:-

OK

Cancel

Que.:- 14) What is the use of Void (0)?

Ans.:- The use of void(0) in JavaScript is typically associated with the href attribute in HTML anchor (<a>) tags. When you use void(0) or javascript:void(0) as the value for the href attribute, it is often used as a placeholder for a JavaScript action that doesn't navigate to a new page.

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7      <style>
8          a{
9              font-size: 50px;
10         }
11     </style>
12 </head>
13 <body>
14     <a href="#" id="demo">Click Here</a>
15     <script>
16         document.getElementById('demo').addEventListener('click', function (event) {
17             event.preventDefault();
18             doSomething();
19         });
20     </script>
21 </body>
22 </html>
```

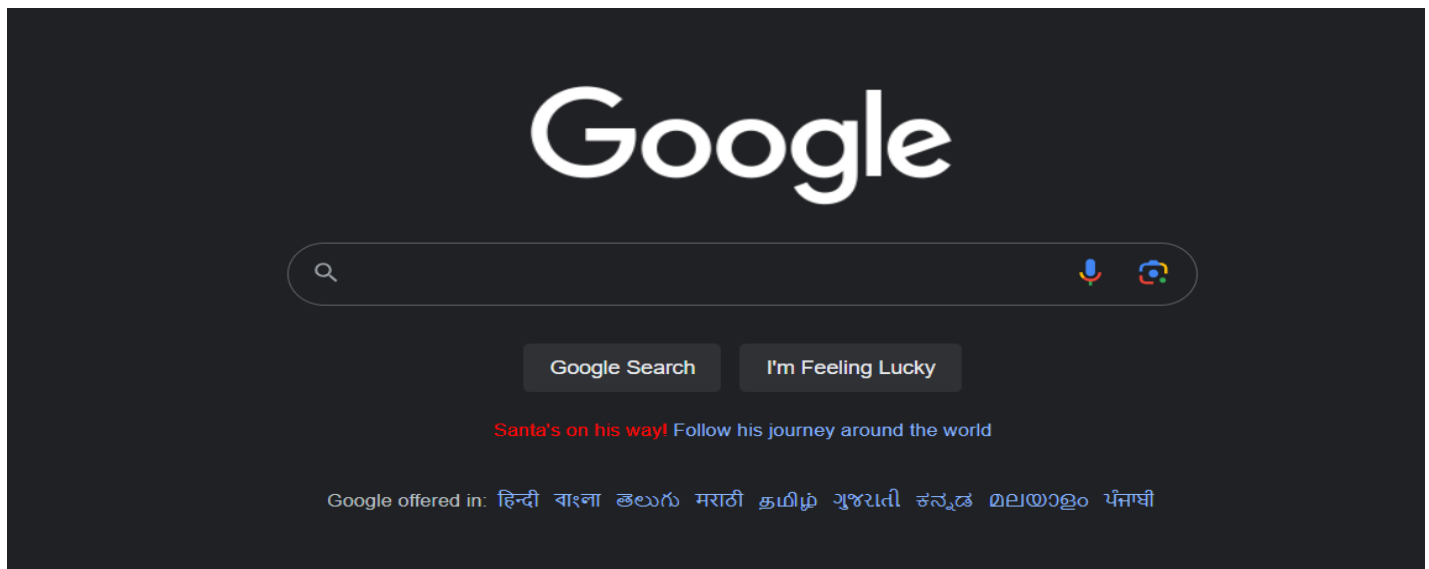
[Click Here](#)

Que.:- 15) How can a page be forced to load another page in JavaScript?

Ans.:- You can use the window.location object to change the current page's URL, effectively forcing the browser to load a new page.

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <title>Document</title>
7  </head>
8  <body>
9    <button onclick="NewPage()">New Page</button>
10   <script>
11     function NewPage() {
12       window.location.href = "https://www.google.com";
13     }
14   </script>
15 </body>
16 </html>
17
```

New Page



Que.:- 16) What are the disadvantages of using innerHTML in JavaScript?

Ans.:-

- **The use of innerHTML very slow:-** The process of using innerHTML is much slower as its contents are slowly built, also already parsed contents and elements are also re-parsed which takes time.
- **Preserves event handlers attached to any DOM elements:-** The event handlers do not get attached to the new elements created by setting innerHTML automatically. To do so one has to keep track of the event handlers and attach it to new elements manually. This may cause a memory leak on some browsers.
- **Content is replaced everywhere:-** Either you add, append, delete or modify contents on a webpage using innerHTML, all contents are replaced, also all the DOM nodes inside that element are reparsed and recreated.
- **Appending to innerHTML is not supported:-** Usually, += is used for appending in JavaScript. But on appending to an HTML tag using innerHTML, the whole tag is re-parsed.

- **Old content replaced issue:-** The old content is replaced even if `object.innerHTML = object.innerHTML + 'html'` is used instead of `object.innerHTML += 'html'`. There is no way of appending without reparsing the whole innerHTML. Therefore, working with innerHTML becomes very slow.
- **Can break the document:-** There is no proper validation provided by innerHTML, so any valid HTML code can be used. This may break the document of JavaScript. Even broken HTML can be used, which may lead to unexpected problems.

Que.:- 17) Create password field with show hide functionalities.

Ans.:-

```

HTML Programs > js.html > html
  Click here to ask Blackbox to help you code faster
1  <!DOCTYPE html>
2  <html>
3  <body>
4  Enter Password:- <input type="password" id="myPass"><br><br>
5  <input type="checkbox" onclick="myFunction()">Show Password
6  <script>
7  function myFunction() {
8      var x = document.getElementById("myPass");
9      if (x.type === "password") {
10         x.type = "text";
11     } else {
12         x.type = "password";
13     }
14 }
15 </script>
16 </body>
17 </html>

```

Enter Password:-

☒ Show Password

Que.:- 18) Create basic math operation in JS

Ans.:-

Click here to ask Blackbox to help you code faster

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7 </head>
8 <body>
9   <h1 id="ans"></h1>
10  1st Number:- <input type="number" id="num1">
11  <br>
12  2nd Number:- <input type="number" id="num2">
13  <br>
14  <button id="add">+</button>
15  <button id="sub">-</button>
16  <button id="mul">*</button>
17  <button id="div">/</button>
18  <button id="mod">%</button><br>
19  Answer is:- <input type="number" id="answer">
20  <script>
21    var n1 = document.getElementById("num1");
22    var n2 = document.getElementById("num2");
23    var result = document.getElementById("answer");
24    document.getElementById("add").addEventListener("click", function () {
25      result.value = parseInt(n1.value) + parseInt(n2.value);
26    });
27    document.getElementById("sub").addEventListener("click", function () {
28      result.value = parseInt(n1.value) - parseInt(n2.value);
29    });
30    document.getElementById("mul").addEventListener("click", function () {
31      result.value = parseInt(n1.value) * parseInt(n2.value);
32    });
33    document.getElementById("div").addEventListener("click", function () {
34      result.value = parseInt(n1.value) / parseInt(n2.value);
35    });
36    document.getElementById("mod").addEventListener("click", function () {
37      result.value = parseInt(n1.value) % parseInt(n2.value);
38    });
39  </script>
40 </body>
41 </html>
```

1st Number:-

2nd Number:-

Answer is:-