**MODULE: 4 (JavaScript)**

1. What is JavaScript?

Ans. JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

2. What is the use of is NaN function?

Ans. The JavaScript is NaN() Function is used to check whether a given value is an illegal number or not. It returns true if the value is a NaN else returns false. It is different from the Number. Is NaN() Method.

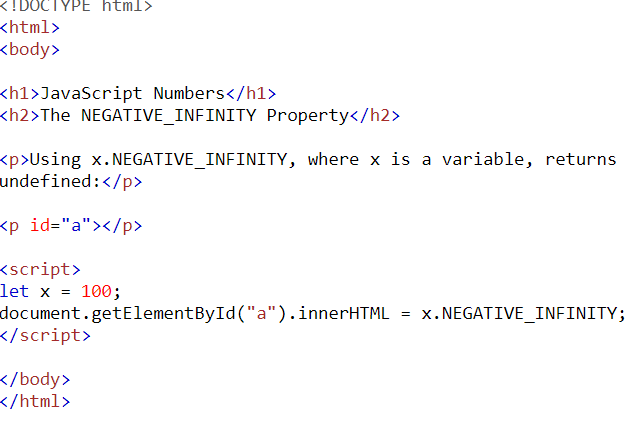
Syntax:

isNaN( value )

 3. What is negative Infinity?

Ans.  Negative infinity is a number in java script, which is derived by 'dividing negative number by zero'.  
- A number object needs not to be created to access this static property.  
- The value of negative infinity is the same as the negative value of the infinity property of the global object.

Example:



|  |
| --- |
|  |

4. Which company developed JavaScript?

Ans. JavaScript was created at Netscape Communications by Brendan Eich in 1995. Netscape and Eich designed JavaScript as a scripting language for use with the company's flagship web browser, Netscape Navigator.

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*.

Undefined Example:

let geek;

Undefined

console.log(geek)

Undeclared Example:

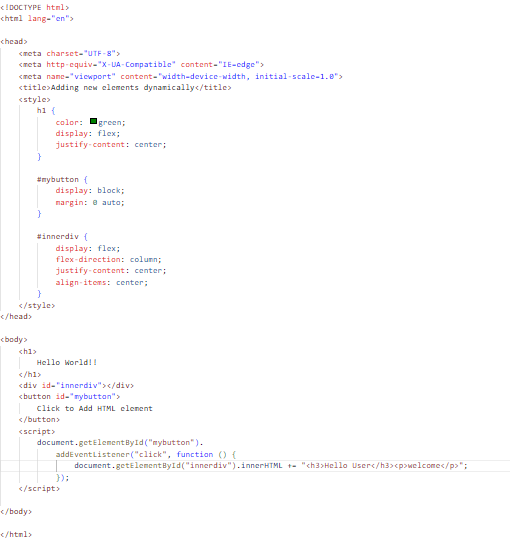
// ReferenceError: myVariable is not defined

console.log(myVariable)

6.Write the code for adding new elements dynamically?

Ans.

Example



Output:



7. What is the difference between ViewState and SessionState?

Ans.

ViewState:

* Maintained at page level only.
* View state can only be visible from a single page and not multiple pages.
* It will retain values in the event of a postback operation occurring.
* Information is stored on the client’s end only.

SessionState:

* Maintained at session level.
* Session state value availability is across all pages available in a user session.
* 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 server.

8. What is === operator?

Ans. The strict equality ( === ) operator checks whether its two operands are equal, returning a Boolean result. Unlike the equality operator, the strict equality operator always considers operands of different types to be different.

9. How can the style/class of an element be changed?

Ans.

1: Changing CSS with the help of the style property:

Syntax:

document.getElementById("id").style.property = new\_style

Example: In this example, we have built a PAN number validator. First, we will take the input value and match it with a regex pattern. If it matches then using JavaScript add an inline style on the <p> tag. Otherwise, add a different style on the <p> tag.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset=”UTF-8”>

<meta Http-equiv=”X-UA-Compatible” content=”IE=edge”>

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

<title>document</title>

</head>

<body>

    <h1 style="color: green;">

        GeeksforGeeks

    </h1>

    <h2>

        How can the style/class of

        an element be changed?

    </h2>

    <b>Validate Pan Number</b>

    <input type="text" id="pan" />

    <p></p>

    <button id="submit">Validate</button>

    <script>

        const btn = document.getElementById("submit");

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

            const pan = document.getElementById("pan").value;

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

            let regex = /([A-Z]){5}([0-9]){4}([A-Z]){1}$/;

            if (regex.test(pan.toUpperCase())) {

                para.innerHTML = "Hurrey It's correct";

                // Inline style

                para.style.color = "green";

            } else {

                para.innerHTML = "OOps It's wrong!";

                // Inline style

                para.style.color = "red";

            }

        });

    </script>

</body>

</html>

2. The className Property: This property is used to set the current class of the element to the specified class.

Syntax:

document.getElementById("id").className = class

Example:

HTML

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">    <head>  <meta charset=”UTF-8”>  <meta Http-equiv=”X-UA-Compatible” content=”IE=edge”>  <meta name=”viewport” content=”width-device-width, initial-scale=1.0”>  <title>document</title>      <style>          .colorBlue {              color: blue;          }            .colorRed {              color: red;          }      </style>  </head>    <body>      <h1 style="color: green;">          GeeksforGeeks      </h1>        <h2>          How can the style/class of          an element be changed?      </h2>        <h3>className Example</h3>        <p class="colorBlue">          GeeksforGeeks is a computer science portal          for geeks.This platform has been designed          for every geek wishing to expand their          knowledge, share their knowledge and is          ready to grab their dream job. GFG have          millions of articles, live as well          as online courses, thousands of tutorials          and much more just for the geek inside you.      </p>        <button id="submit">Change Color</button>        <script>          const btn = document.getElementById("submit");          const para = document.querySelector("p");            btn.addEventListener("click", function () {              para.className = "colorRed";          });      </script>  </body>    </html> |

10.How to read and write a file using JavaScript?

Ans. Files can be read and written by using java script functions – fopen(),fread() and fwrite().  
  
The function fopen() takes two parameters – 1. Path and 2. Mode (0 for reading and 3 for writing). The fopen() function returns -1, if the file is successfully opened.  
  
Example:

file=fopen(getScriptPath(),0);

The function fread() is used for reading the file content.  
  
Example:

str = fread(file,flength(file) ;

The function fwrite() is used to write the contents to the file.  
  
Example:

file = fopen("c:\MyFile.txt", 3);// opens the file for writing  
fwrite(file, str);// str is the content that is to be written into the file.

11.What are all the looping structures in JavaScript?

Ans. JavaScript supports different kinds of loops:

* for - loops through a block of code a number of times.
* for/in - loops through the properties of an object.
* for/of - loops through the values of an iterable object.
* while - loops through a block of code while a specified condition is true.

12.How can you convert the string of any base to an integer in JavaScript?

Ans. Given a string containing an integer value and along with that user passes a base value. We need to convert that string of any base value to an integer in JavaScript.

String          Integer

"1002"            1002

For performing the above-illustrated task, we would be using a method (or a function) provided by JavaScript called as parseInt().

This is a special method, provided by JavaScript, that takes an integer value (of any base which is either specified or not) and further converts the string into an integer value.

Syntax:

* Following is the syntax that a user may use to convert a string into an integer value (of any base)-

parseInt(string\_value, base)

* Alternatively, if we don’t want to specify the base value and just want to convert our string value into an integer value itself, then we may use the following syntax also-

parseInt(string\_value)

Default value returned by base or radix of parseInt() method is 10. In other words, if we don’t specify any base or radix value then it by default converts the string value to an integer value by taking into regard the base or radix value as 10.

Let us visualize all of the above-illustrated facts with some of the following examples-

Example:  In this example, we would be passing the string value in a method (which is explicitly declared for ease purpose) and further that string value is passed inside the parseInt() method which then further converts that string value in the corresponding integer value.

JavaScript

|  |
| --- |
| <script>      let stringConversion = (string\_value) => {        console.log("Initial Type: " + typeof string\_value);        let integer\_value = parseInt(string\_value);        console.log("Final Type: " + typeof integer\_value);        console.log(integer\_value);      };        stringConversion("512000");      stringConversion("126410");      stringConversion("0x8975");  </script> |

Output:

Initial Type: string

Final Type: number

512000

Initial Type: string

Final Type: number

126410

Initial Type: string

Final Type: number

35189

13.What is the function of the delete operator?

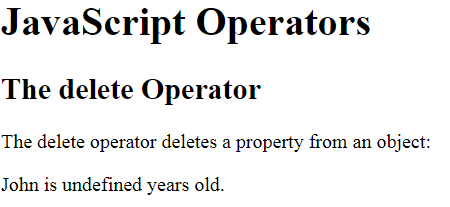
Ans The **delete** operator deletes a property from an object.

The delete operator is designed to be used on object properties. It has no effect on variables or functions.

The delete operator deletes both the value of the property and the property itself.



Output:



14.What are all the types of Pop up boxes available in JavaScript?

Ans. JavaScript has three kind of popup boxes: Alert box, Confirm box, and Prompt box.

## Alert Box

An alert box is often used if you want to make sure information comes through to the user.

When an alert box pops up, the user will have to click "OK" to proceed.

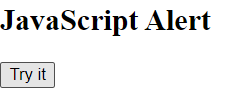
### Syntax

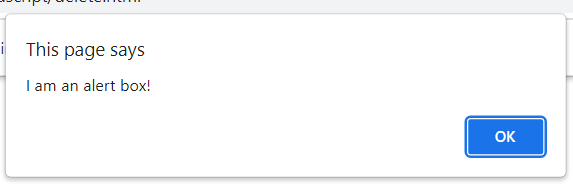
window.alert("*sometext*");

Example:



Output:





## Confirm Box

A confirm box is often used if you want the user to verify or accept something.

When a confirm box pops up, the user will have to click either "OK" or "Cancel" to proceed.

If the user clicks "OK", the box returns **true**. If the user clicks "Cancel", the box returns **false**.

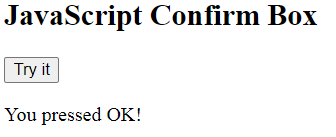
### Syntax

window.confirm("*sometext*");

Example:



Output:



## Prompt Box

A prompt box is often used if you want the user to input a value before entering a page.

When a prompt box pops up, the user will have to click either "OK" or "Cancel" to proceed after entering an input value.

If the user clicks "OK" the box returns the input value. If the user clicks "Cancel" the box returns null.

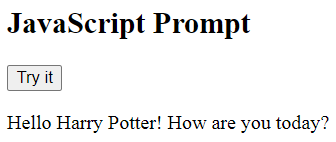
### Syntax

window.prompt("*sometext*","*defaultText*");

Example:



Output:



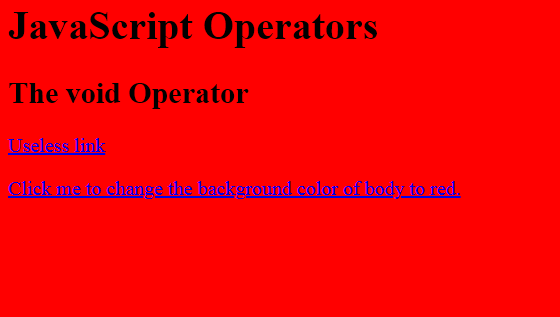
15.What is the use of Void (0)?

Ans. void(0) means return undefined as a primitive value. We use this to prevent any negative effects on a webpage when we insert some expression.

Example:



Output:



16. How can a page be forced to load another page in JavaScript?

Ans.We can use window.location property inside the *script* tag to forcefully load another page in Javascript. It is a reference to a Location object that is it represents the current location of the document. We can change the URL of a window by accessing it.

Syntax:

<script>

window.location = <Path / URL>

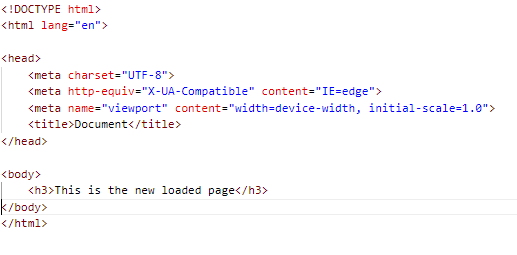
</script>

Example:

**Step 1:** Create a file named ***index.html***. Add a heading and two buttons to it. One button forcefully loads a page with a live URL and the other button loads a local HTML page. In the *<script>* tag we have two functions, one loads gfg home page, and the second loads a local HTML page using ***window.location*** property.



**Step 2:** Create a file named ***newPage.html***. This is the local HTML page that would be loaded by Javascript.



17. What are the disadvantages of using inner HTML in JavaScript?

Ans. Following are the disadvantages of using inner HTML :

### Inner HTML is slow

Inner HTML is slow because when we use the inner HTML property in the code it allows us to change using the JavaScript language. It is very slow because as inner HTML already parses the content even we have to parse the content again so that’s why it takes time.

* Event handlers attached to any DOM element are preserved

When we have used the event handlers then the event handlers are not automatically attached to the new elements created by innerHTML.

* Replacement is done everywhere

When innerHTML property is used to modify, all the DOM nodes will have to be parsed and created again.

* It is not possible to append innerHTML

In JavaScript, ‘+=’ is commonly used for appending. However, when using innerHTML to append to an HTML tag, the entire tag is re-parsed

* Breaks the document

InnerHTML does not provide proper validation, so any valid HTML code can be used. This has the potential to break the JavaScript document. Even broken HTML can be used, which can cause unexpected issues.

* Used for Cross-site Scripting

The text and images or elements in the webpage can used by hackers or malicious users to change the text or data and show some different undesired or threatful content by the other HTML element tag. This leads to change of sensitive and confidential information.