**FRONTEND ASSIGNMENT**

**WEB DESIGNING**

**MODULE: 4 (JavaScript Basic & DOM)**

1. **What is JavaScript?**

**Ans-** Java script is a lightweight, cross-platform, and interpreted scripting language. it is well-known for the development webpages.

Java script is used to make webpages interactive (e.g., having complex animations, clickable buttons, popup menus, etc.)

Java script can be used for client side development as well as server side development

1. **What is the use of is NaN function?**

## Ans – In java Script NaN is Short for “Not -a-Number”. The is NaN() method returns true if a value is NaN. The Nan () method converts the value to a number before testing it.

## 3.What is negative Infinity?

## Ans – (i). The negative infinity in JavaScript is a constant value which is used to represent a value which is the lowest available.

## (ii). This means that no other number is lesser than this value.

## Note:- JavaScript shows the NEGATIVE\_INFINITY value as -Infinity.

## Ex:-

## 

## Which company developed JavaScript?

## Ans- Java Script is a scripting language developed by Netscape.

## It can be used to program web browser.

## What are undeclared and undefined variables?

Ans- **Undefined:-** variable means a variable has been declared but does not have a value.

## Undeclared:- variable means that the variable does not exist in the program at all.

## Ex:- Undefined example

## var a ;

## console.log(a);

## output :

## *Undefined*

## Undeclared example

## Console.log(cat);

## Output:

## *Undeclared*

## 6.Write the code for adding new elements dynamically?

## Ans-

## 

## 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.

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.

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.

**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:-  Reading from the file

After the File System module is imported, the reading of the file in JavaScript can be done by using the readFile() function.

## Example:-

const M = require('M')

M.readFile('tp.txt', (err, inputD) => {

   if (err) throw err;

      console.log(inputD.toString());

})

### **Output:-**

You are reading the content from Tutorials Point

The text which is displayed in the console is the text which is in the given file.

## Write operation on a file

After the File System file is imported then, the writeFile() operation is called. The writeFile() method is used to write into the file in JavaScript.

Example:-

const M = require('M')

let fInput = "You are reading the content from Tutorials Point"

M.writeFile('tp.txt', fInput, (err) => {

   if (err) throw err;

   else{

      console.log("The file is updated with the given data")

   }

})

**11.What are all the looping structures in JavaScript?**

**Ans:-**

* 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:-

|  |
| --- |
| <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 in JavaScript is used to delete an object’s property.

### **Syntax:-**

delete object.property;  
// OR  
delete object["property"];

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

Ans:- There are three types of pop-up boxes in javascript namely**Alert Box**,**Confirm Box** and**Prompt Box**.

**Alert Box:** It is used when a warning message is needed to be produced. When the alert box is displayed to the user, the user needs to press ok and proceed.

**Prompt Box:** It is a type of pop up box which is used to get the user input for further use. After entering the required details user have to click ok to proceed next stage else by pressing the cancel button user returns the null value.

**Confirm Box:** It is a type of pop-up box that is used to get authorization or permission from the user. The user has to press the ok or cancel button to proceed.

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

**Ans:-**  “javascript:void(0);” in anchor tag can prevent the page to reload and JavaScript functions can be called on single or double clicks easily.

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

**Ans:- Step 1:** Create a file named ***index.html***.

* index.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">  </**head**>    <**body**>      <**h3**>This is the original page</**h3**>      <**br**>        <**button** onclick="force\_load\_gfg()">          Force Load GFG Page      </**button**>      <**br**><**br**>        <**button** onclick="force\_load\_local()">          Force Load Local HTML page      </**button**>        <**script**>          function force\_load\_gfg() {              window.location =                  "<https://www.geeksforgeeks.org/>"          }            function force\_load\_local() {              window.location =                  "F:/gfg/PageRedirect/newPage.html"          }      </**script**>  </**body**>    </**html**> |

**Step 2:** Create a file named ***newPage.html***.

* newPage.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**> New Page </**title**>  </**head**>    <**body**>      <**h3**>This is the new loaded page</**h3**>  </**body**>    </**html**> |

**17.What are the disadvantages of using innerHTML in JavaScript?**

**Ans:- Disadvantages of using innerHTML property in JavaScript:**

* **The use of innerHTML very slow:** The process of using innerHTML is much slower as its contents as slowly built, also already parsed contents and elements are also re-parsed which takes time.
* **Content is replaced everywhere:** Either you add, append, delete or modify contents on a webpage using innerHTML, all contents is 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

**Example:**

|  |
| --- |
| <!DOCTYPE html>  <**html**>    <**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**>          Using innerHTML in JavaScript      </**title**>  </**head**>    <**body** style="text-align: center">        <**h1** style="color:green">          GeeksforGeeks      </**h1**>        <**p** id="P">          A computer science          portal for geeks.      </**p**>        <**button** onclick="geek()">          Try it      </**button**>        <**p** id="p"></**p**>        <**script**>          function geek() {              var x = document.getElementById("P")                          .innerHTML;                document.getElementById("p")                          .innerHTML = x;                document.getElementById("p")                          .style.color = "green";          }      </**script**>  </**body**>    </**html**> |

***complete***