WEB DESIGNING ASSIGNMENT

Module(JAVASCRIPT BASIC & DOM) - 4

1. What is JavaScript?

- JavaScript is a scripting language.
- JavaScript is a dynamic programming language that's used for web development, in web applications, for game development, and lots more. It allows you to implement dynamic features on web pages that cannot be done with only HTML and CSS.

2. What is the use of isNaN function?

- The isNaN() function is used to check whether a given value is an illegal number or not. It returns true if value is a NaN else returns false. It is different from the Number.

3. What is negative Infinity?

- The negative infinity in JavaScript is a constant value which is used to represent a value which is the lowest available. This means that no other number is lesser than this value. It can be generated using a self-made function or by an arithmetic operation.

4. Which company developed JavaScript?

- JavaScript start in 1995 and developed by **Netscape**.
- The first JavaScript engine was created by Brendan Eich at Netscape Communications Corporation, for the Netscape Navigator Web browser.

5. What are undeclared and undefined variables?

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

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

6. Write the code for adding new elements?

- Example:

```
- function login() {
-     let uname = document.form1.uname.value;
-     let pass = document.form1.upass.value;
-
-     if (uname == "") {
-         document.getElementById("uname").innerHTML = "Please Enter User Name";
-
-         // document.getElementById("u_name").innerHTML = "<style>#u_name{border:3px
  solid red; border-radius:10px;}</style>";
-         return false;
-     }
-     if (pass == "") {
-         document.getElementById("upass").innerHTML = "Please Enter Password";
-         // document.getElementById("pwd").innerHTML = "<style>#pwd{border:3px solid
  red; border-radius:10px;}</style>";
-         return false;
-     }
-     if (pass == "^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%*?&])[A-Za-
  z\d@$!%*?&]{8,10}$") {
-         document.getElementById("upass").innerHTML = "Please enter valid password";
-         // alert("Please enter valid password");
-         return false;
-     }
- }
```

7. What is the different between View State and Session State?

-

View State	Session State
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 post back 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.
used to allow the persistence of page-instance-specific data.	used for the persistence of user-specific data on the server's end.
It can be used to store information that you wish to access from same web page.	It can be used to store information that you wish to access on different web pages.

8. What is === operator?

- 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?

- There are two approaches that allow us to changed style/class element.
  1. **Changing CSS with the help of the style property:**
     Syntax: document.getElementById("id").style.property = new_style

  2. **Changing the class itself**: We can use two properties that can be used to manipulate the classes.
     A. **The ClassList Property**: The classList is a read-only property that returns the CSS class names of an element as a DOMTokenList object.
        Syntax: document.getElementById("id").classList
     B. **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

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

- The read and write operations in a file can be done by using some commands. But the module which is required to perform these operations is to be imported. The required module is ‘fs’ which is called as File System module in JavaScript.
- **Read File**: After the File System module is imported, the reading of the file in JavaScript can be done by using the readFile() function.
- **Syntax**: readFile(path, format, callBackFunc)
- **Write 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.
- **Syntax**: writeFile(path, inputData, callBackFunction)

11. What are all the looping structures in JavaScript?

- Loop is the count number of iteration. Loops can execute a block of code a number of times. Loops are handy, if you want to run the same code over and over again, each time with a different value.
- There are three types of loop:

1. <u>While Loop</u>

- The while loop loops through a block of code as long as a specified condition is true.
- **Syntax**: while (condition){
  Statements
  }

```
-    let a = 1;
-          while (a <= 10) {
-              document.writeln(a);
-              a++;
-          }
```

2. **<u>Do…While Loop</u>**:

- The do while loop is a variant of the while loop. This loop will execute the code block once, before checking if the condition is true, then it will repeat the loop as long as the condition is true.

- **Syntax**: do {
  Statements
  }
  while (condition)

```
let a = 11;
        do {
            document.writeln(a);
            a++;
        }
        while (a <= 30);
```

3. **<u>For Loop</u>**:
   - The for statement creates a loop with 3 optional expressions:
   - **Syntax**: for (expression 1; expression 2; expression 3)
     {
     Statements
     }
   - Expression 1 is executed (one time) before the execution of the code block.
   - Expression 2 is defines the condition for executing the code block.
   - Expression 3 is executed (every time) after the code block has been executed.

```
for (a = 1; a <= 20; a++) {
            document.writeln(a);
        }
```

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

- In JavaScript parseInt() function is used to convert the passed in string parameter or value to an integer value itself. This function returns an integer of base which is specified in second argument of parseInt() function.
- parseInt() function returns Nan(not a number) when the string doesn't contain number.
- **Syntax:** parseInt (value, radix)

13. What is the function of the delete operator?

- The delete operator removes a property from an object. If the property's value is an object and there are no more references to the object, the object held by that property is eventually released automatically.
- **Syntax:** delete object.property
  delete object[property]

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

- JavaScript has a three kind of pop-up boxes:
  1. Alert Box
  2. Confirm Box
  3. Prompt Box

- **1. 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.
- **Example:**

```
- // ALERT POP-UP BOX
-         function btn_alert() {
-             alert("Hello User");
-         }
```

- **2. 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.
- **Example:**

```
- function btn_confirm() {
-             if (confirm("Are you sure youn want to leave this page???")) {
-                 // document.getElementById("p1").innerHTML = "ok";
-                 document.body.style.backgroundColor = "gray";
-             }
-             else {
-                 // document.getElementById("p1").innerHTML = "cancel";
-                 document.body.style.backgroundColor = "maroon";
-             }
-         }
```

- **3. 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.
- **Example:**

```
- // PROMPT POP-UP BOX
-         function btn_prompt() {
-             let uname = prompt("Enter Your Name");
-
-             if (uname == "") {
-                 alert("Enter Your Name");
-             }
-             else {
-                 document.getElementById("p2").innerHTML = uname;
-             }
-         }
```

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

- JavaScript void 0 means returning undefined (void) as a primitive value. You might come across the term “JavaScript:void(0)” while going through HTML documents. It is used to prevent any side effects caused while inserting an expression in a web page.

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

- 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>

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

- Disadvantages of using innerHTML:
  1. Event handlers attached to any DOM element are preserved.

2. Replacement is done everywhere.
3. It is not possible to append innerHTML.
4. Breaks the document.
5. Used for Cross-site Scripting.