Front-end Assignment

Name: Mahavirsinh P.Dodiya

Course: Front End Development

Assignment-4(JavaScript Basic & DOM)

1) What is JavaScript?

Ans)

- JavaScript is the Programming Language for the Web. JavaScript can update and change both HTML and CSS.
- JavaScript can calculate, manipulate and validate data.
- JavaScript, often abbreviated as JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS.

2) What is the use of isNaN function?

Ans)

- The JavaScript isNaN() 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. IsNaN() Method.
- **Syntax:** isNaN(value)
- **Parameter Values:** This method accepts a single parameter as mentioned above and described below:
- **Value:** It is a required value passed in the isNaN() function.
- **Return Value:** It returns a Boolean value i.e. returns true if the value is NaN else returns false.

3) What is negative Infinity?

Ans)

NEGATIVE_INFINITY is a special numeric value that is returned when an arithmetic operation or mathematical function generates a negative value greater than the largest representable number in JavaScript.

4) Which company developed JavaScript?

Ans)

JavaScript was invented by **Brendan Eich** in 1995. It was developed for Netscape 2, and became the ECMA-262 standard in 1997.

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 'type of' 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.

6) Write the code for adding new elements dynamically?

Ans)

Syntax: document. createElement("tagName");

7) What is the difference between ViewState and SessionState?

Ans)

The basic difference between these two is that the **ViewState** is to manage state at the client's end, making state management easy for end-user while **SessionState** manages state at the server's end, making it easy to manage content from this end too.

8) What is === operator?

Ans)

- It is known as strict equality operator, it compares both the value and the type which is why the name "strict equality".
- 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) **getElementById ()** method is used to return the element in the document with the "id" attribute and the "className" attribute can be used to change/append the class of the element.

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

Ans) readFile() and **rs. writeFile()** methods are used to read and write of a file using javascript. The file is read using the fs. readFile() function, which is an inbuilt method.

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

while - loops through a block of code while a specified condition is true

do/while - also loops through a block of code while a specified condition is true

Syntax:

For-

for (initialization; Condition; Increment/decrement)

{ // code block to be executed }

While

initialization

while(condition)

{ // code block to be executed

Increment/decrement };

Do….While

Do

{ initialization

// code block to be executed

Increment/decrement }

while(conditon).

Ex:

```
<!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>
    <script>
   for(x=1;x>=3;x++)
        {
            document.write("for ");
        }
        document.write("<br><br>");
    y=1;
    while(y>=3)
        {
            document.write("while ");
            y++;
        }
        document.write("<br><br>");
    z=1;
    do
```

```
        {
            document.write("do_while ");
            z++;
        }while(z>=3)
    </script>
</body>
</html>
```

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

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

13) What is the function of the delete operator?

Ans)

- 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.
- Delete is comparatively a lesser-known operator in JavaScript. This operator is more specifically used to delete JavaScript object properties.
- The JavaScript **pop(), shift(),** or **splice()** methods are available to delete an element from an array. But because of the key-value pair in an object, deleting is more complicated. Note that, the delete operator only works on objects and not on variables or functions.

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
- 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*");
- The window.alert() method can be written without the window prefix.

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*");
- The window.confirm() method can be written without the window prefix.

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*");
- The window.prompt() method can be written without the window prefix.

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

Ans) The void operator is used to evaluate an expression and returns the **undefined**. Generally, this operator is used for obtaining the undefined primitive value. It is often used with hyperlinks. Usually the browser refreshes the page or loads a new page on clicking a link. The **javascript:void(0)** can be used when we don't want to refresh or load a new page in the browser on clicking a hyperlink.

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

Ans) It 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?

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.
- **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 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.
- **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. String concatenation just does not scale when dynamic DOM elements need to be created as the plus' and quote openings and closings becomes difficult to track.
- **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.
- **Can also be used for Cross-site Scripting(XSS):** The fact that innerHTML can add text and elements to the webpage, can easily be used by malicious users to manipulate and display undesirable or harmful elements within other HTML element tags. Cross-site Scripting may also lead to loss, leak and change of sensitive information.