Module (JAVASCRIPT BASIC & DOM) – 4

1. **What is JavaScript?**

**Ans.** JavaScript is a scripting or programming language that allows you to implement complex features on web pages. Displaying timely content updates, interactive maps, animated 2D/3D graphics, scrolling video jukeboxes, etc.

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

**Ans.** NaN stands for “Not a Number” and is a Value in JavaScipt used to represent an undefined or unrepresentable value. NaN is the result of an operation thet was supposed to return a number, but couldn’t because of an error or undefined/empty value.

1. **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 that the largest representable number in JavaScript. JavaScript displays the NEGATIVE\_INFINITY value as -Infinity.

1. **Which company developed JavaScript?**

**Ans.** Netscape Communications Corporation developed JavaScript in 1995. Brendan Eich, a Netscape employee, created the language.

1. **What are undeclared and undefined variables?**

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

**Undeclared:** It occurs when we try to access any variable that is not initialization 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.

1. **Write the code for adding new elements dynamically?**

**Ans. Creation of new element:** New elements can be created in JS by using the createElement() method.

**Ex.** Syntax:

document**.**createElement(‘<tagname>’);

// Where <tagName> can be any HTML

// tagName like div, button, I, etc.

// newDiv element has been created

For Eg: **let newDiv=document.createElement(‘div’);**

1. **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.
* Used to allow the persistence of page-instance-specific data.
* ViewState values are lost/cleared when new page is loaded.

**SessionState:**

* Maintain at session level.
* Session state value availability is across all pages available in a user session.
* In session state, user data remainsin the server. Data is available to user until the browser is closed or there is session expiration.
* Information is stored on the server.
* Used for the persistence of user-specific data on the server’s end.
* SessionState can be cleared by programmer or user or in case of timeouts.

1. **What is === operators?**

**Ans.** In JavaScript, === is a strict equality operator. It checks if two values are aqual and also ensures they are of the same type.

**Exa.** Let a = 5;

b = ‘5’;

console.log (a===b);

This will return false because one is a number and the other is a string.

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

**Ans. Change Class:**

Function changeClass() {

let element= document.querySelector(‘#myDIv’);

element.classList.replace(‘oldClass’, ‘newClass’);

}

**Change Style:**

Document.getElementById(‘id’).style.fontStyle=”italic’;

1. **How to read and write a file using JavaScript?**

**Ans. readFile():** The file is read using thefs.readFile() function, which is an inbuilt method. This technique reads the full file into memory and stores it in a buffer.

**Exa. Fs.readFile( file\_name, encoding, callback\_function )**

**writeFile:** The fs.write() function is used to write data to a file in an asynchronous manner. If the file already exists, it will be replaced.

**Exa.** **Fs.writeFile( file\_name, data, options, callback )**

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

**Ans.** There are many loops in Javascript: for-loop, while-loop, while-loop, while-do-loop, for-in-loop, for-of-loop, for-each-loop etc.

**Exa.** **For loop:**

for (initialization; condition; increment / decrement (update)) {

code to execute

}

for (let counter = 1; counter <= 108; counter++) {

console.log('Ram', counter);

}

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

**Ans.** There’s a function called parseInt() in JavaScript, this is used for parsing a string as an argument and it returns an integer of the specified radix (basically the base of the numerical system) as output.

**Exa.** let num = '1234';

console.log(num);

console.log(parseInt(num))

console.log('type: ', typeof Number(num))

1. **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.

**Exa.** object.delete(property)

delete(object.property)

1. **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.

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

**Ans.** 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. By running void (0) in the URL JavaScript code, nothing is evaluated or returned.

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

**Ans.** In JavaScript, we can use window.location object to force a page to load another page. We can use the location object to set the URL of a page. There are different ways – window.location.href property, window.location.assign() and window.location.relace() methods, to set the URL of a new page using the location object.

**Exa.** window.location.href=‘new\_url’;

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

**Ans.** There are some disadvantages of using innerHTML in JavaScript:

1. 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.
2. **Preserves event handlers attached to any DOM elements**
3. **Content is replaced everywhere**
4. **Appending to innerHTML is not supported**
5. **Old content replaced issue**
6. **Can break the document**
7. **Can also be used for Cross-site Scripting**