Module 4 : JavaScript Basic & DOM

1. What is JavaScript?

→ JavaScript is the Programming Language for the Web. JavaScript can update and change both HTML and CSS. JavaScript can calculate, manipulate and validate data.

2. What is the use of isNaN function?

→ 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.

3. What is negative Infinity?

→ 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 (i.e., more negative than -Number. MAX\_VALUE). JavaScript displays the Negative\_infinity value as -Infinity. This value behaves mathematically like infinity; for example, anything multiplied by infinity is infinity, and anything divided by infinity is zero.

4. Which company developed JavaScript?

→ JavaScript was invented by **Brendan Eich** in 1995. It was developed for **Netscape** 2, and became the ECMA-262 standard in 1997. After Netscape handed JavaScript over to ECMA, the Mozilla foundation continued to develop JavaScript for the Firefox browser. Mozilla's latest version was 1.8.

5. What are undeclared and undefined variables?

→ An undeclared variable is one which has not previously been defined with the “let”, “var”, or “const” keywords. An undefined variable is one which has been declared but has not been assigned any value.

6. Write the code for adding new elements dynamically?

→ New elements can be created in JS by using the **createElement()** method.

**Syntax:** document.createElement("<tag Name>");

Where <tag Name> can be any HTML. Tag Name like div, ul, button, etc. newDiv element has been created For Eg: let newDiv = document.createElement("div");

7. What is the difference between ViewState and SessionState?

→ A ViewState is a state of a page within a browser wherein the values of controls persist when post back operation is done. When another page is loaded, the previous page data is no longer available. SessionState is the data of a user session and is maintained on the server side. This data available until user closes the browser or session time-outs.

8. What is === operator?

→ The strict equality (===) operator checks whether its two operands are equal, returning a Boolean result. Unlike the [equality](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Equality) operator, the strict equality operator always considers operands of different types to be different.

For Example :**[Comparing operands of the same type](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Strict_equality" \l "comparing_operands_of_the_same_type).**

"hello" === "hello"; // true

"hello" === "hola"; // false

3 === 3; // true

3 === 4; // false

true === true; // true

true === false; // false

null === null; // true

[**Comparing operands of different types**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Strict_equality#comparing_operands_of_different_types)**.**

"3" === 3; // false

true === 1; // false

null === undefined; // false

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

→ Another way to alter the style of an element is by changing its class attribute. class is a reserved word in JavaScript, so in order to access the element’s class, you use element.className. You can append strings to className if you want to add a class to an element, or you could just overwrite className and assign it a whole new class.

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

→ The[fs.readFile()](https://www.geeksforgeeks.org/node-js-fs-readfile-method/) and [rs.writeFile()](https://www.geeksforgeeks.org/node-js-fs-writefile-method/) 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. This technique reads the full file into memory and stores it in a buffer.

**Syntax**: fs.readFile( file\_name, encoding, callback\_function )

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

**Syntax**: fs.writeFile( file\_name, data, options, callback )

11. What are all the looping structures in JavaScript?

→ Loops are handy, if you want to run the same code over and over again, each time with a different value.

***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

**do/while** - also 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?

→ In JavaScript [parseInt()](https://www.geeksforgeeks.org/javascript-parseint-function/) 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.

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**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/delete#syntax): delete object.property

delete object[property]

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

→ 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.
* **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**.
* **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.

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](https://www.simplilearn.com/tutorials/html-tutorial/what-is-html) documents. It is used to prevent any side effects caused while inserting an expression in a web page. For instance, URLs or hyperlinks are the common examples of using JavaScript void 0. Suppose you insert a link and want to call some JavaScript through it. Usually, when you click on a link, the browser will either reload or open a new page. However, if you just want to call JavaScript through that link, you would not want the entire page to refresh. This is where the JavaScript:void(0) will come in handy.

When you use JavaScript void 0, it will return an undefined primitive value. This will prevent the browser from opening a new or reloading the web page and allowing you to call the JavaScript through it.

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

→ **Approach:**We can use **[window.location](https://www.geeksforgeeks.org/javascript-window-location-and-document-location-objects/)** 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?

→ 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. To change that, we have to track the event handlers and manually attach them to a new element. It means that first, we have to fetch the element property through innerHTML, and then we have to attach them to a new element.
* **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.