# Module (JAVASCRIPT BASIC & DOM) – 4

**1.What is JavaScript?**

**Ans.**

Javascript is a loght weight cross platform,and interpreted scripting language.

It is well known for the development web pages,many known browser environment also use it.

Javascript is use to make webpages interactive ( e.g having complex , animation , clickable buttons , pop up,menus ,etc

Javascript can be use for clint side development as well as server site development.

Javascript conatain a standard object like array , date , and math and core set of languge element like operator control structure & statement.

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

**Ans.**

In JavaScript NaN is short for "Not-a-Number".

The isNaN() method returns true if a value is NaN.

The isNaN() method converts the value to a number before testing it.

**example:**

[**https://github.com/chandaniacharya/Assignments/blob/main/assignment%20practicles/isNaN().html**](https://github.com/chandaniacharya/Assignments/blob/main/assignment%20practicles/isNaN().html)

**3. What is negative Infinity?**

**Ans**.

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.

In ECMAtScrip v1 and later, you can also use -Infinity instead of Number.NEGATIVE\_INFINITY.

**example:**

[**https://github.com/chandaniacharya/Assignments/blob/main/assignment%20practicles/nagative%20infinity.html**](https://github.com/chandaniacharya/Assignments/blob/main/assignment%20practicles/nagative%20infinity.html)

**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.

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. (Identical to ES5).

**Internet Explorer** (IE4) was the first browser to support ECMA-262 Edition 1 (ES1).

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

**Ans.**

**Undefined:**It occurs when a variable has been declared but has not been assigned with 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 *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.

**example:**

[**https://github.com/chandaniacharya/Assignments/tree/main/assignment%20practicles/undeclaired%20%26%20undefined**](https://github.com/chandaniacharya/Assignments/tree/main/assignment%20practicles/undeclaired%20%26%20undefined)

**6. Write the code for adding new elements dynamically**

**Ans.**

Javascript is a very important language when it comes to learning how the browser works. Often there are times we would like to add dynamic

elements/content to our web pages. This post deals with all of that.

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

**example:**

[**https://github.com/chandaniacharya/Assignments/tree/main/assignment%20practicles/adding%20new%20element%20in%20dynamically%20way**](https://github.com/chandaniacharya/Assignments/tree/main/assignment%20practicles/adding%20new%20element%20in%20dynamically%20way)

**7. What is the difference between ViewState and SessionState?**

**Ans.**

| **ViewState** | **SessionState** |
| --- | --- |
| 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 postback 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. |
| Information is stored on the client’s end only. | Information is stored on the server. |
| used to allow the persistence of page-instance-specific data. | used for the persistence of user-specific data on the server’s end. |
| ViewState values are lost/cleared when new page is loaded. | SessionState can be cleared by programmer or user or in case of timeouts. |

**8. What is === operator?**

**Ans.**

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.

**example:**

[**https://github.com/chandaniacharya/Assignments/blob/main/assignment%20practicles/%3D%3D%3D%20oparator.html**](https://github.com/chandaniacharya/Assignments/blob/main/assignment%20practicles/%3D%3D%3D%20oparator.html)

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

**Ans.**

The class name is used as a selector in HTML which helps to give some value to the element attributes.

The document.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.

**Syntax:**

document.getElementById('myElement').className = "myclass";

**example:**

[**https://github.com/chandaniacharya/Assignments/tree/main/assignment%20practicles/class%20of%20an%20element%20be%20changed**](https://github.com/chandaniacharya/Assignments/tree/main/assignment%20practicles/class%20of%20an%20element%20be%20changed)

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

**Ans.**

The fs.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.

This technique reads the full file into memory and stores it in a buffer.

**Syntax:**

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

**Parameters:**

**filename:** It contains the filename to be read, or the whole path if the file is saved elsewhere.

**encoding:** It stores the file’s encoding. ‘utf8’ is the default setting.

**callback function:**This is a function that is invoked after the file has been read. It requires two inputs:

**err:** If there was an error.

**data:**The file’s content.

**Return Value:**It returns the contents contained in the file, as well as any errors that may have occurred.

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 )

**Parameters:**

**file\_name**: It’s a string, a buffer, a URL, or a file description integer that specifies the location of the file to be written.

When you use a file descriptor, it will function similarly to the fs. write() method.

**data**: The data that will be sent to the file is a string, Buffer, TypedArray, or DataView.

**options:** It’s a string or object that may be used to indicate optional output options. It includes three more parameters that may be selected.

**encoding**: It’s a string value that indicates the file’s encoding. ‘utf8’ is the default setting.

**mode**: The file mode is specified by an integer number called mode. 0o666 is the default value.

**flag**: This is a string that indicates the file-writing flag. ‘w’ is the default value.

**callback**: This function gets invoked when the method is run.

**err**: If the process fails, this is the error that will be thrown.

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

**Ans**.

Loops are used in JavaScript to perform repeated tasks based on a condition.

Conditions typically return true or false. A loop will continue running until the defined condition returns false.

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

**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

**nested loop**-The most common type of nested loops will be one loop inside another.

The first loop is usually called the outer loop while the second loop is called the inner loop.

The outer loop always executes first, and the inner loop executes inside the outer loop each time the outer loop executes once.

**example:**

[**https://github.com/chandaniacharya/Assignments/tree/main/assignment%20practicles/loops**](https://github.com/chandaniacharya/Assignments/tree/main/assignment%20practicles/loops)

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

**Ans.**

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**.

**Syntax:**

parseInt(Value, radix)

**example:**

[**https://github.com/chandaniacharya/Assignments/blob/main/assignment%20practicles/parseint().html**](https://github.com/chandaniacharya/Assignments/blob/main/assignment%20practicles/parseint().html)

**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.

**example:**

[**https://github.com/chandaniacharya/Assignments/blob/main/assignment%20practicles/delete%20operator.html**](https://github.com/chandaniacharya/Assignments/blob/main/assignment%20practicles/delete%20operator.html)

**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, and 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.

**2.conform** **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**.

**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:**

[**https://github.com/chandaniacharya/Assignments/tree/main/assignment%20practicles/pop%20up%20boxes**](https://github.com/chandaniacharya/Assignments/tree/main/assignment%20practicles/pop%20up%20boxes)

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

**ans.**

javascript: void(0) means return undefined as a primitive value.

We use this to prevent any negative effects on a webpage when we insert some expression.

For example,

in the case of URL hyperlinks. Hyperlinks open by reloading the page when the user clicks on the link.

When you need to run some other code in such cases, you can use javascript: void(0).

**example:**

[**https://github.com/chandaniacharya/Assignments/blob/main/assignment%20practicles/void(0).html**](https://github.com/chandaniacharya/Assignments/blob/main/assignment%20practicles/void(0).html)

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

**Ans.**

**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>

**example:**

[**https://github.com/chandaniacharya/Assignments/blob/main/assignment%20practicles/window.location.html**](https://github.com/chandaniacharya/Assignments/blob/main/assignment%20practicles/window.location.html)

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

**Ans.**

Inner HTML is just like a property of every HTML element.

It means that let’s say we have created a div tag in HTML and inside the div tag,

we have created the paragraph tag then we can say that the paragraph tag is an inner HTML and the whole div is an outer HTML.

Also, we can use innerHTML in JavaScript to fetch the property of any tag by using is’t tag name, id, or by class name.

Following is the snippet of code to show an inner HTML −

<div><p>Hello world</p><div>

In the above snippet of code, you can say that the whole div tag is an outer HTML and the whole p tag is an inner HTML.

Let’s try to understand with a suitable **example**

[**https://github.com/chandaniacharya/Assignments/blob/main/assignment%20practicles/inner%20html.html**](https://github.com/chandaniacharya/Assignments/blob/main/assignment%20practicles/inner%20html.html)

**thank you**