# ASSIGNMENT-1

1. What are the popup boxes 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.

Syntax:

window.alert(“-----");

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(“------”);

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");

2. What is JavaScript engine for Safari?

Safari's javescript engine is known as "Nitro" or "JavaScriptCore webkit".

3. Test the following requirement using JavaScript

a. Create two separate *.js files (Names : ClientScripts1.js, ClientScripts2.js)
b. Write some messages using alert() . Message should contains the file name
c. Import the above js files in other *.html document and observe.

4. Do some changes and observe
a. Give some wrong file name while import and observe the messages in console window

Failed to load resource:net::ERR_FILE_NOT_FOUND

b. Is Filename case sensitive check practically?
NO

5. What are the key differences between Java and JavaScript?

| **Java** | **JavaScript** |
|---|---|
| This is OOP or Object-Oriented programming language | This is an object-based scripting language |
| A stand-alone language | Not stand-alone, incorporated into HTML program for operation |
| Strongly typed language is used, and data type of variable is decided before declaring or using it | Language utilized is loosely typed, so that the user does not have to worry about the data type before the declaration |
| Code has to be compiled | The code is all text |
| Slightly more complex | Easier in comparison |
| Used to perform complex tasks | Complex tasks cannot be executed |
| Large amount of memory is required | Memory consumption is lesser |
| Programs are saved with ".java" extension | Programs are saved with JavaScript, ".js" extension |

| | |
|---|---|
| Stored in the / client host machine under the “Byte” code | Stored in host or client machine as “source” code |
| Compiled on the server before it is executed on the client side | JavaScript is interpreted on the client side |
| Is static and the code once written can be run on any computing platform | Dynamic and is a cross-platform language |

# ASSIGNMENT-2

1. Is it possible to disable JavaScript in browser?  If yes, how to disable javascript in browser?

Yes, it is possible to disable JavaScript in most web browsers.
Disabling JavaScript can be useful for various reasons, such as improving security, reducing the load on your device, or troubleshooting certain issues on websites.
Here's how you can disable JavaScript in some common web browsers:

Google Chrome:
Open Chrome.
Click the three dots menu (top-right corner).
Go to "Settings."
Scroll down and click "Privacy and security."
Click "Site settings.
"Under "Content," click "JavaScript."
Toggle the switch to turn off JavaScript.

Mozilla Firefox:
Open Firefox.
Type "about:config" in the address bar and press Enter.
Confirm that you'll be careful.
In the search bar, type "javascript.enabled.
"Double-click the "javascript.enabled" preference to set it to "false."

Microsoft Edge:
Open Edge.
Click the three dots menu (top-right corner).

Go to "Settings."
Scroll down and click "Cookies and site permissions.
"Click "JavaScript."
Toggle the switch to turn off JavaScript.Safari

(macOS):
Open Safari.
Click "Safari" in the menu bar.
Select "Preferences.
"Go to the "Websites" tab.
On the left sidebar, select "JavaScript."
Disable JavaScript for the websites you want, or uncheck "Enable JavaScript."

2.What is the difference between undefined and not defined in JavaScript?

The main difference between "undefined" *and* "not defined" *is that* "undefined" is a value that can be assigned to a variable, while "not defined" indicates that a variable does not exist.

Another difference is that "undefined" is usually caused by forgetting to assign a value to a variable, while "not defined" is usually caused by a typo or by trying to access a variable that is out of scope.

## 3. Difference between =, == and ===?

| = | == | === |
|---|---|---|
| = in JavaScript is used for assigning values to a variable. | == in JavaScript is used for comparing two variables, but it ignores the datatype of variable. | === is used for comparing two variables, but this operator also checks datatype and compares two values. |
| It is called as assignment operator | It is called as comparison operator | It is also called as comparison operator |
| The assignment operator can evaluate to the assigned value | Checks the equality of two operands without considering their type. | Compares equality of two operands with their types. |
| It does not return true or false | Return true if the two operands are equal. It will return false if the two operands are not equal. | It returns true only if both values and data types are the same for the two variables. |
| = simply assign one value of variable to another one. | == make type correction based upon values of variables. | === takes type of variable in consideration. |
| == will not compare the value of variables at all. | The == checks for equality only after doing necessary conversations. | If two variable values are not similar, then === will not perform any conversion. |

## 4. What does the isNaN() function?

The isNaN() function determines whether a value is NaN, first converting the value to a number if necessary.

isNaN() is a function property of the global object.

For number values, isNaN() tests if the number is the value NaN. When the argument to the isNaN() function is not of type Number, the value is first coerced to a number, and the resulting value is then compared against NaN.

The isNaN() function answers the question "is the input functionally equivalent to NaN when used in a number context". If isNaN(x) returns false, you can use x in an arithmetic expression as if it's a valid number that's not NaN.
If isNaN(x) returns true, x will get coerced to NaN and make most arithmetic expressions return NaN (because NaN propagates).

for example, to test whether an argument to a function is arithmetically processable (usable "like" a number), and handle values that are not number-like by throwing an error, providing a default value, etc. This way, you can have a function that makes use of the full versatility JavaScript provides by implicitly converting values depending on context.

## 5. Difference between Client-side JavaScript and Server-side JavaScript?

1. Server-side scripting is utilized in the backend when the source code is invisible or hidden on the client side. In contrast, client-side scripting is utilized at the front end, which users may access via the browser.
2. The files are inaccessible to the client-side script. In contrast, the server-side script has access to the files on the web server.
3. Server-side scripting is very useful for personalizing web pages and implementing dynamic updates to websites. In contrast, the client-side script may effectively minimize the load on the server.
4. The client-side script responds faster than the server-side script.
5. Client-side scripting is less secure than server-side scripting due to the server-side scripts are commonly concealed from the client end. In contrast, the client-side scripts are exposed to users.
6. The client-side script is run on a local computer. In contrast, the server-side script is run on a remote machine.
7. The client-side programming languages are *HTML, CSS, and JavaScript*. In contrast, server-side scripting programming languages, including *PHP, ColdFusion, Python, ASP.net, Java, C++, Ruby, C#,*

8. A server-side script communicates with the server when it is executed. In contrast, Client-side scripting doesn't need server involvement.
9. The script on the client side lacks security. However, the server-side script is protected.
10. The client-side script is dependent on the user's browser version. In contrast, the Server-side doesn't depend on the client.