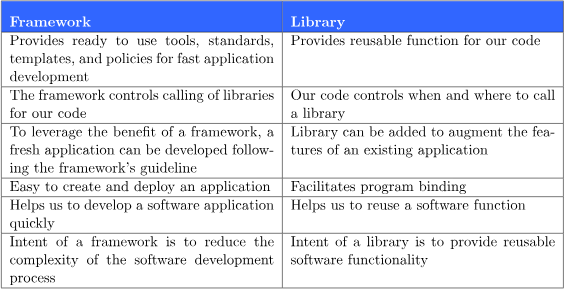
**2 Jan 2024**

1. Jquery is a javascript library
2. Library vs framework



1. Asynchronous vs Synchronous

Asynchronous is a non-blocking architecture, so the execution of one task isn't dependent on another. Tasks can run simultaneously.

Synchronous is a blocking architecture, so the execution of each operation depends on completing the one before it.

1. Document.queryselector = $
2. Minifier script
3. $(“h1”).css(“color”,”green”);
4. $(“h1”).addClass(“className1 className2”);
5. $(“h1”).removeClass(“className1 className2”);
6. $(“h1”).hasClass(“className”);

[Math.trunc()](https://developer.mozilla.org/en/docs/Web/JavaScript/Reference/Global_Objects/Math/trunc) (truncate fractional part, also see below)

[Math.floor()](https://developer.mozilla.org/en/JavaScript/Reference/Global_Objects/Math/floor) (round down)

[Math.ceil()](https://developer.mozilla.org/en/JavaScript/Reference/Global_Objects/Math/ceil) (round up)

[Math.round()](https://developer.mozilla.org/en/JavaScript/Reference/Global_Objects/Math/round) (round to nearest integer)

| onchange | An HTML element has been changed |
| --- | --- |
| onclick | The user clicks an HTML element |
| onmouseover | The user moves the mouse over an HTML element |
| onmouseout | The user moves the mouse away from an HTML element |
| onkeydown | The user pushes a keyboard key |
| onload | The browser has finished loading the page |

| [String length](https://www.w3schools.com/js/js_string_methods.asp#mark_length)  [String charAt()](https://www.w3schools.com/js/js_string_methods.asp#mark_charat)  [String charCodeAt()](https://www.w3schools.com/js/js_string_methods.asp#mark_charcodeat)  [String at()](https://www.w3schools.com/js/js_string_methods.asp#mark_at)  [String [ ]](https://www.w3schools.com/js/js_string_methods.asp#mark_propertyaccess)  [String slice()](https://www.w3schools.com/js/js_string_methods.asp#mark_slice)  [String substring()](https://www.w3schools.com/js/js_string_methods.asp#mark_substring)  [String substr()](https://www.w3schools.com/js/js_string_methods.asp#mark_substr) See Also: [String Search Methods](https://www.w3schools.com/js/js_string_search.asp)  [String Templates](https://www.w3schools.com/js/js_string_templates.asp) | [String toUpperCase()](https://www.w3schools.com/js/js_string_methods.asp#mark_touppercase)  [String toLowerCase()](https://www.w3schools.com/js/js_string_methods.asp#mark_tolowercase)  [String concat()](https://www.w3schools.com/js/js_string_methods.asp#mark_concat)  [String trim()](https://www.w3schools.com/js/js_string_methods.asp#mark_trim)  [String trimStart()](https://www.w3schools.com/js/js_string_methods.asp#mark_trimstart)  [String trimEnd()](https://www.w3schools.com/js/js_string_methods.asp#mark_trimend)  [String padStart()](https://www.w3schools.com/js/js_string_methods.asp#mark_padstart)  [String padEnd()](https://www.w3schools.com/js/js_string_methods.asp#mark_padend)  [String repeat()](https://www.w3schools.com/js/js_string_methods.asp#mark_repeat)  [String replace()](https://www.w3schools.com/js/js_string_methods.asp#mark_replace)  [String replaceAll()](https://www.w3schools.com/js/js_string_methods.asp#mark_replaceall)  [String split()](https://www.w3schools.com/js/js_string_methods.asp#mark_split) |
| --- | --- |

## Extracting String Parts

There are 3 methods for extracting a part of a string:

* slice(*start*, *end*)
* substring(*start*, *end*)
* substr(*start*, *length*)

## Extracting String Characters

There are 4 methods for extracting string characters:

* The at(*position*) Method
* The charAt(*position*) Method
* The charCodeAt(*position*) Method
* Using property access [] like in arrays

| [String indexOf()](https://www.w3schools.com/js/js_string_search.asp#mark_indexof)  [String lastIndexOf()](https://www.w3schools.com/js/js_string_search.asp#mark_lastindexof)  [String search()](https://www.w3schools.com/js/js_string_search.asp#mark_search) See Also: [Basic String Methods](https://www.w3schools.com/js/js_string_methods.asp)  [String Templates](https://www.w3schools.com/js/js_string_templates.asp) | [String match()](https://www.w3schools.com/js/js_string_search.asp#mark_match)  [String matchAll()](https://www.w3schools.com/js/js_string_search.asp#mark_matchall)  [String includes()](https://www.w3schools.com/js/js_string_search.asp#mark_includes)  [String startsWith()](https://www.w3schools.com/js/js_string_search.asp#mark_startswith)  [String endsWith()](https://www.w3schools.com/js/js_string_search.asp#mark_endswith) |
| --- | --- |

| [Array length](https://www.w3schools.com/js/js_array_methods.asp#mark_length)  [Array toString](https://www.w3schools.com/js/js_array_methods.asp#mark_tostring)()  [Array at()](https://www.w3schools.com/js/js_array_methods.asp#mark_at)  [Array join()](https://www.w3schools.com/js/js_array_methods.asp#mark_join)  [Array pop()](https://www.w3schools.com/js/js_array_methods.asp#mark_pop)  [Array push()](https://www.w3schools.com/js/js_array_methods.asp#mark_push) Se Also: [Search Methods](https://www.w3schools.com/js/js_array_search.asp)  [Sort Methods](https://www.w3schools.com/js/js_array_sort.asp)  [Iteration Methods](https://www.w3schools.com/js/js_array_iteration.asp) | [Array shift()](https://www.w3schools.com/js/js_array_methods.asp#mark_shift)  [Array unshift()](https://www.w3schools.com/js/js_array_methods.asp#mark_unshift)  [Array delete()](https://www.w3schools.com/js/js_array_methods.asp#mark_delete)  [Array concat()](https://www.w3schools.com/js/js_array_methods.asp#mark_concat)  [Array copyWithin()](https://www.w3schools.com/js/js_array_methods.asp#mark_copywithin)  [Array flat()](https://www.w3schools.com/js/js_array_methods.asp#mark_flat)  [Array splice()](https://www.w3schools.com/js/js_array_methods.asp#mark_splice)  [Array toSpliced()](https://www.w3schools.com/js/js_array_methods.asp#mark_tospliced)  [Array slice()](https://www.w3schools.com/js/js_array_methods.asp#mark_slice) |
| --- | --- |

1. let temp={

a:10,

b:20,

sum(){

return this.a+this.b;

},

multi:()=>{

return this.a\*this.b;

},

};

console.log(temp.sum());

console.log(temp.multi())

1. The multi method is an arrow function (=>). Arrow functions do not have their own this context; instead, they inherit it from the enclosing scope at the time of their definition. In this case, the arrow function is defined within the global context, so this inside multi refers to the global object
2. console.log(+true);

console.log(!'abc'); explain its output

The unary plus (+) is used to convert the operand to a number.

1. \_proto\_
2. One object inherit feature of another object
3. If we use prototype instead of creating individual code then output will be same but when we access the object then in case of prototype the information/properties will load when required and this will improve the speed of browser and reduce object complexity.
4. Stateless vs statefull
5. Memory lifecycle
6. Static vs dynamic memory
7. Stack and heap
8. Try catch
9. Oops
10. Array and string method
11. Comments in js
12. How processor
13. Data segmentation vs heap vs static vs dynamic memory allocation
14. What is readonly
15. Segmentation in RAM
16. **What is js and why we are using**

**JavaScript, a programming language commonly used for web development. It's primarily used to add interactivity and dynamic content to websites. JS runs in web browsers and allows developers to manipulate elements on a webpage, respond to user actions, and create dynamic, interactive user experiences. It's an essential tool for building modern, responsive, and engaging web applications.**

1. **How many ways we can pass argument**

**Positional Argument**

**Named Argument**

**Argument Object**

1. **Function/Types of function**
2. **Named Function : Name Function ahs a Name Identifier**

**function sum(a,b){**

**Return a+b;**

**};**

**Console.log(sum(5,3));**

1. **Anonymous Function : They Do not have a name Identifier and can not be referenced directly by the name**

**console.log(function(a,b){**

**return a\*b;**

**}(4,5));**

**So, if u want to reuse or having a comlex code use named function.**

1. **Arrow Function: It is a short cut of writing a normal function**

**(Parameter list)=>{finction body}**

**anonymous functions and arrow functions serve similar purposes, but arrow functions provide a more concise syntax and handle the this keyword differently.**

1. **Call Back Function: A function that is passed as a argument to another function**
2. **Function Expression:A function expression is a way of define a function by assigning it to a variable and this function can be named or anonymous both**

**But there is no any benefit of using named function so we generally use anonymous function as we are calling with the help of variable name**

**const add=function(a,b){**

**return a+b;**

**};**

**console.log(add(5,3));**

1. **IIFE: Immediately Invoked Function Expression used to remove global scope pollution and to execute immidiately (function definition)()**
2. **Higher-Order Function:This function take one or more function as a argument or return function as a result**
3. **Difference between Exception and error**

**JavaScript throws Errors**

**Developers throws Exceptions**

**In summary, "error" is a more specific term in JavaScript and is often associated with the Error object and runtime issues, while "exception" is a broader term that can encompass various unexpected conditions, especially in asynchronous programming. The use of these terms can vary depending on the context and the type of issues encountered in a JavaScript program.**

**Exceptions are expected failures, which we should recover from.**

**Errors are unexpected failures. By definition, we cannot recover elegantly from unexpected failures.**

1. **How to clone and copy of object**

**copy: replicate to existing instance (shallow or deep)**

**We had a instance of object and we are just replicating that instance**

**clone: replicate to new instance (always deep)**

**We are creating a new instance of that object**

**Object.assign({},objectname)**

**Spread Operator**

**JSON.stringify() and JSON.parse():**

**Object.create()**

1. **How to select, modify, create and remove DOM elements**
2. **Event Listner using J Query**
3. **Synchronize/Asynchronize in Javascript**

**08-JAN-24**

1. Why we use JASON- Data will be compressed, secure and easy to access
2. Instead of call back function what are the function we can use

Promises and async/await are two alternatives to using callbacks that can make code more readable and maintainable

1. A promise is simply a function that returns an Object which you can attach callbacks to.
2. Why we use DOM

allows us to create, change, or remove elements from the document

1. How to modify a table row using DOM
2. Dynamic table we can change data permanently but not in static here we can change only temporarily
3. Without using constructor declare a class
4. Without void and no return anything create a class