JavaScript Course

Course Topics:

Basics	Advance	Prov Level
How Website Works? What is JavaScript? History of JavaScript Values & Variables Data Types in JavaScript Concat & Type Coercion Operators & Expression If Statements & Loops Functions in JavaScript Arrays in JavaScript Strings in JavaScript Math Object Date & Time in JavaScript	EcmaScript 2015 - 2024 Window Objects – BOM vs DOM Events Objects in JavaScript localStorage in JavaScript Timing Based Events Objects in JavaScript OOPs in JavaScript Event Propagations Advanced Functions JSON & FETCH API & other APIs Promises, Async-Await Error Handling in JavaScript	How JavaScript Works ? Notes + List of Deprecated properties ECOM WEBSITE WITH HTML, CSS & JAVASCRIPT

What is JavaScript ?

JavaScriptimprovestheuserexperienceofthe webpageby convertingitfromastatic pageinto aninteractiveone.
OR
JavaScript is used to update and change both HTML and CSS.
Itaddsbehaviortowebpages

Ways to Write JavaScript Code.

Inline JavaScript

```
<button onclick="alert('Hello')">Click me </button>
```

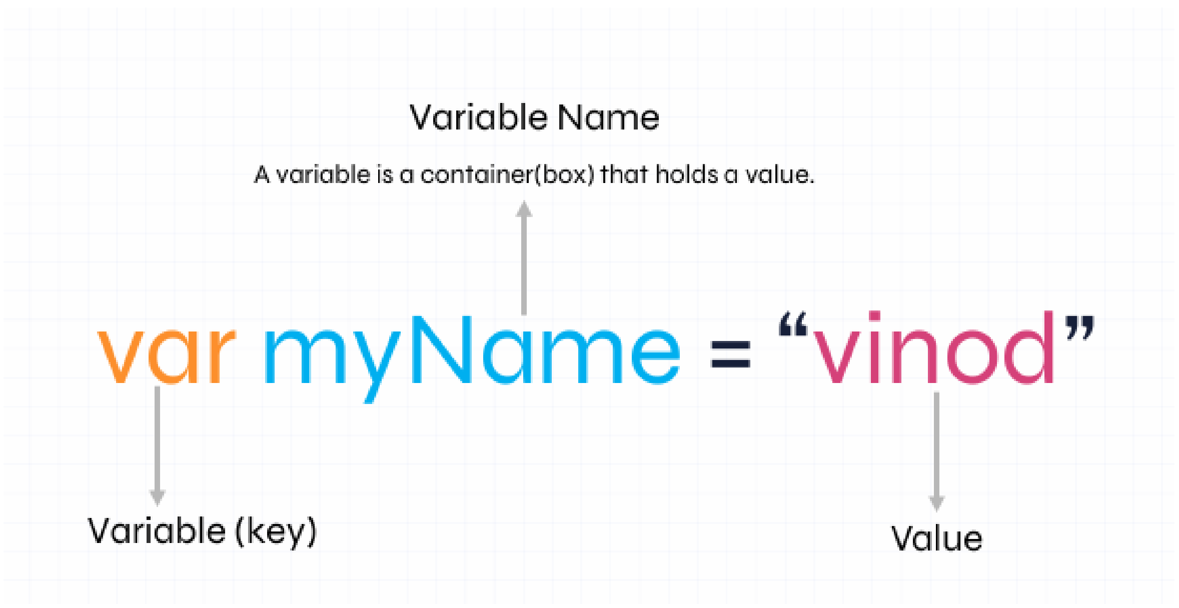
Internal JavaScript

```
<script> console.log('Hello, world!'); </script>
```

External JavaScript

```
<script src="script.js"></script>
```

JavaScript Values & Variables



Naming Variables: Rules and Best Practices

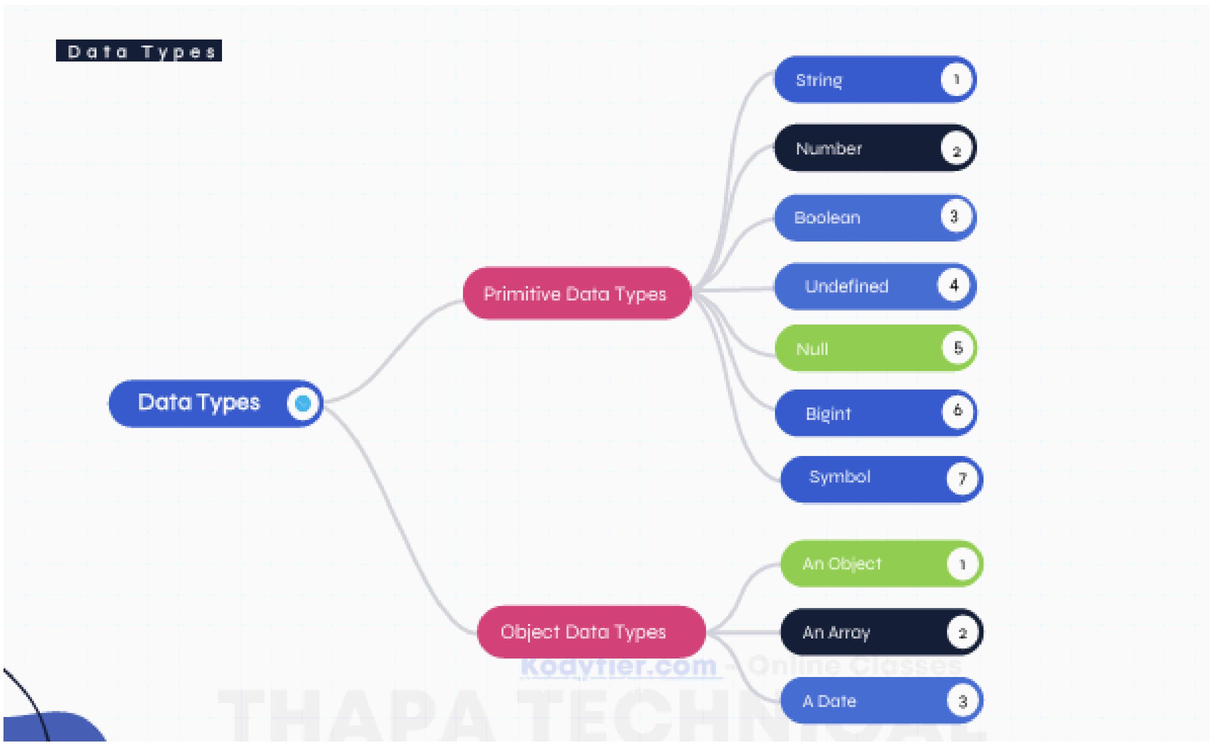
- Variable names must start with a letter, an underscore(_) or a dollar sign($).
- Variables cannot be the same as reserved keywords such as if or const.
- Variable names cannot contain spaces.
- variable names are case sensitive.

- By convention, JavaScript variable names are written in camelCase.
- Variable names can be as long as you need.

Questions

Questions	Answers
var my_firstName = "John";	This is a valid variable name.
var _myLastName$ = "Doe";	This is a valid variable name.
var 123myAge = 25;	This is not a valid variable name.
var $cityName = "New York";	This is a valid variable name.
var my@Email = “Thapa@me.com”	This is not a valid variable name.

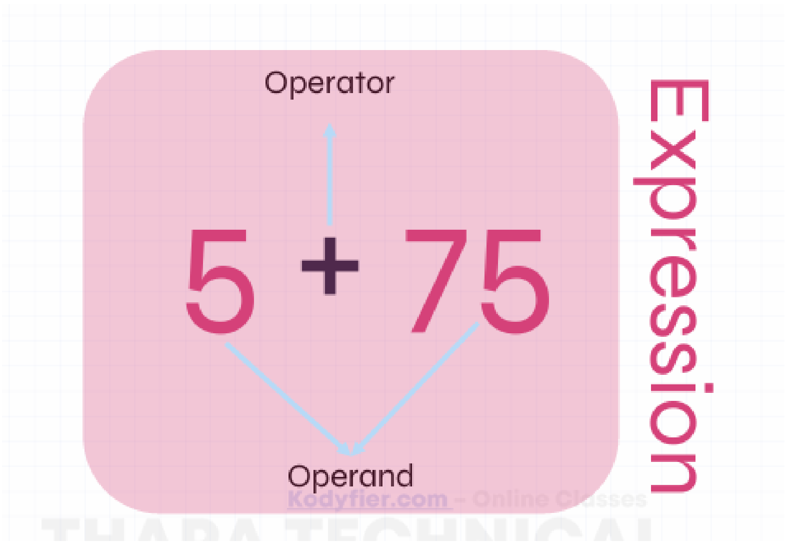
JavaScript Data Types



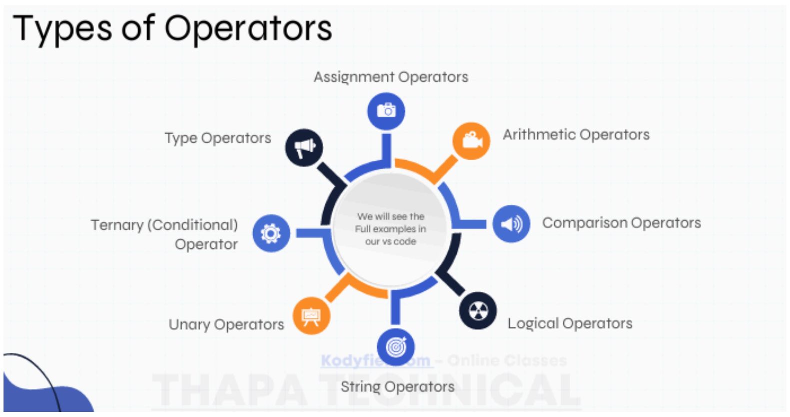
Interview Questions - DataTypes

1. What is the difference between a null and undefined in JavaScript?
2. What is the purpose of typeof operator in JavaScript?
3. What is the result of `typeof null` in JavaScript?
4. What are the primitive data types in JavaScript?
5. Explain the concept of truthy and falsy values in JavaScript. Provide examples?

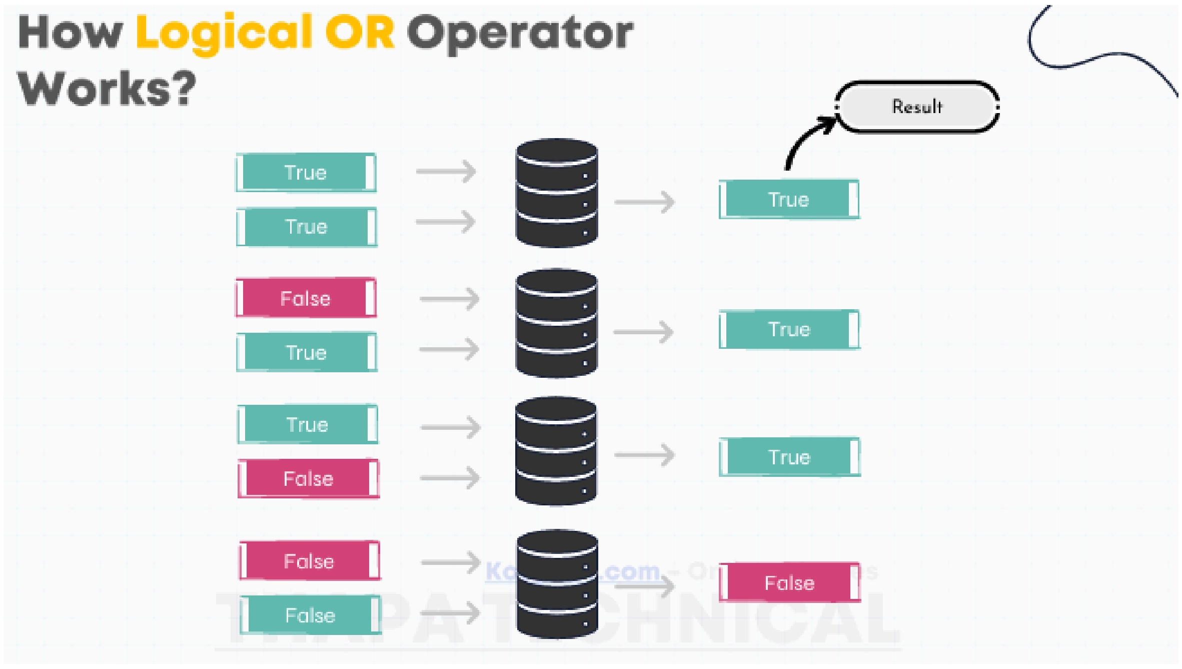
JavaScript Expressions & Operators



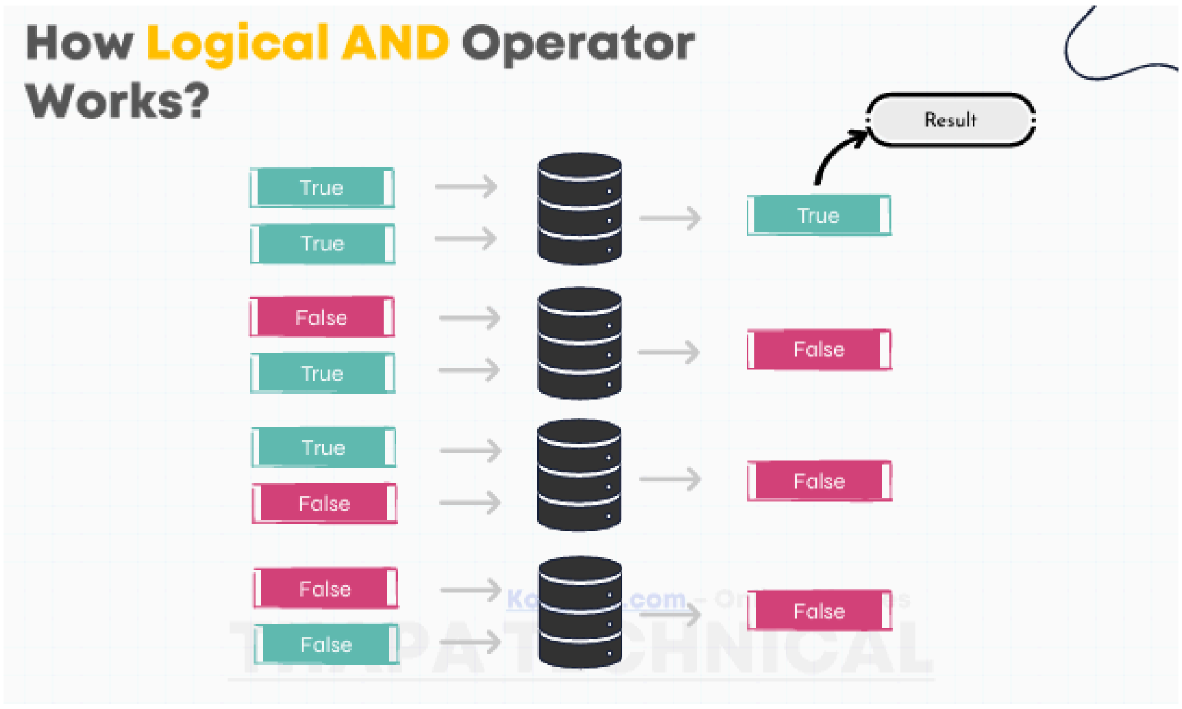
Types of Operators

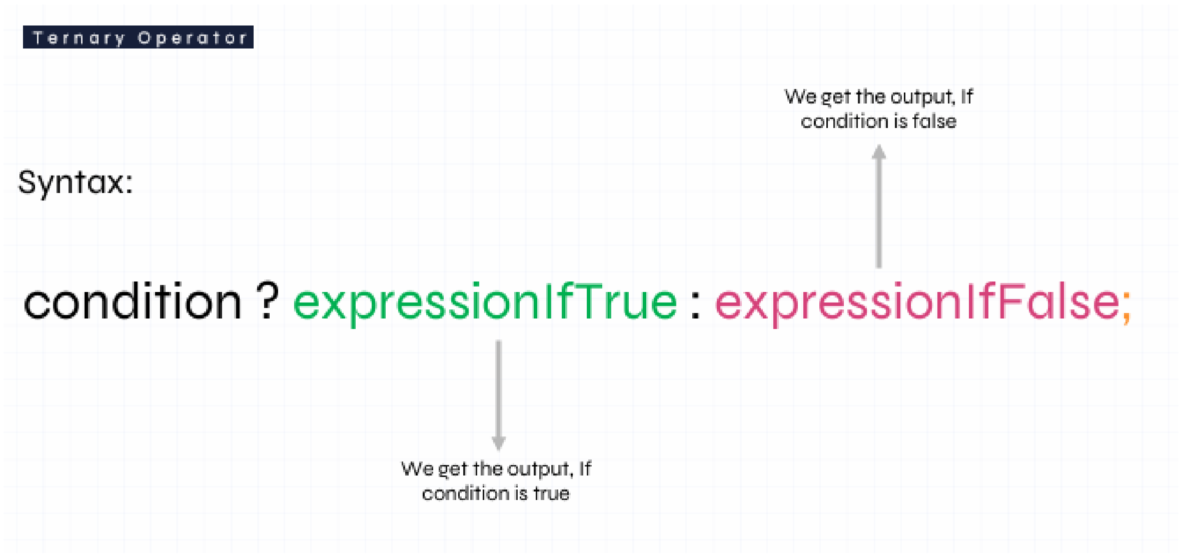


How Logical OR Operator Works



How Logical AND Operator Works?



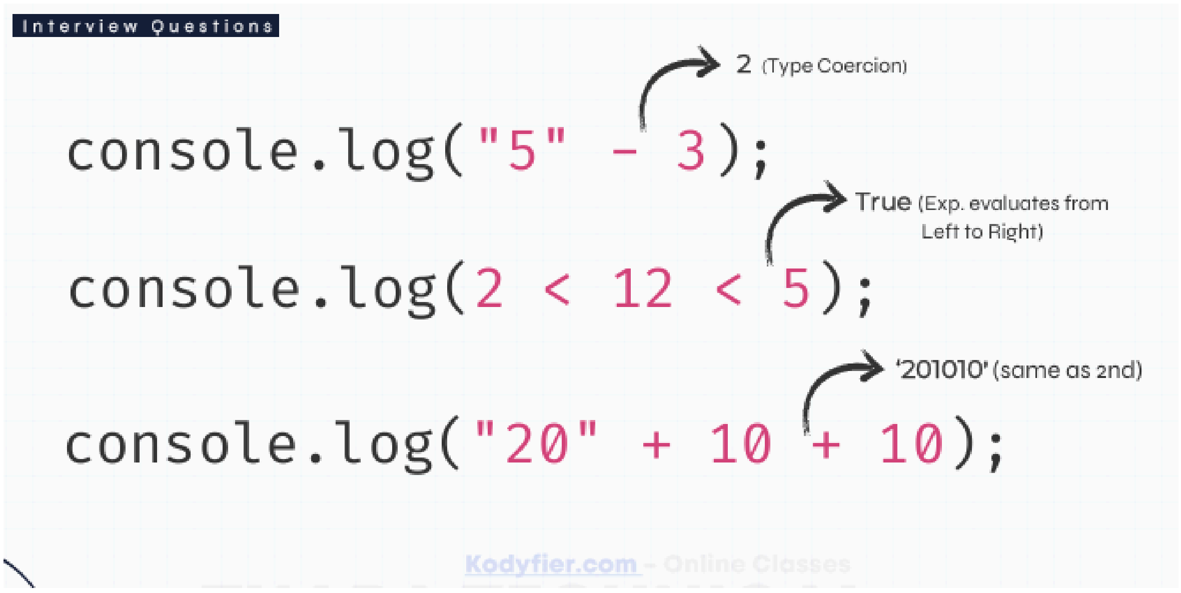

Ternary Operator

Syntax:

```
condition ? expressionIfTrue : expressionIfFalse;
```

We get the output, If condition is false
We get the output, If condition is true

Interview Questions


Interview Questions

```
console.log("5" - 3);
console.log(2 < 12 < 5);
console.log("20" + 10 + 10);
```

2 (Type Coercion)
True (Exp. evaluates from Left to Right)
'201010' (same as 2nd)
Kodyfier.com - Online Classes

JavaScript Control Statement & Loops

1. If.. Else Statement
2. Switch Statement
3. While Loop
4. Do While Loop

5. For Loop
6. For in / For Of Loop (Later in Arrays)

Syntax-if Else

```
if (condition) {
 // Code to be executed if the condition is true
} else {
 // Code to be executed if the condition is false
}
```

Example - If Statement

```
var temp = 40;
if (temp > 30) {
  console.log(“Let’s go to Beach  ”)
 } else {
  console.log(“Watch TV at Home ”)
 }
```

If Else Statements Quiz

1. Write a program to check if a number is even or odd.
2. Write a program to check if a number is prime.
3. Write a program to check if a number is positive, negative, or zero.

Switch Statement

Q: Write a JavaScript switch statement that takes a variable areaOfShapes representing
different shapes, and based on its value, calculates and logs the area of the corresponding
shape. Consider three shapes: 'Rectangle,' 'Circle,' and 'Square.' For

'Rectangle,' use variables a
and b as the sides; for 'Circle,' use a variable r as the radius; and for 'Square,' use variable a as the
side length. If the provided shape is not recognized, log a message saying, 'Sorry the shape is not
available.' Test your switch statement with areaOfShapes set to 'Square' and sides a and b set to
5 and 10, respectively. Ensure that the correct area (25 in this case) is logged to the console

---

Syntax - While Loop

```
while (condition) {
  // Code to be executed as long as the
condition is true
 }
```

Syntax - Do-While Loop

```
do {
  // Code to be executed at least once
 } while (condition);
```

Syntax - For Loop

```
 for(initializer; condition; iteration)
 {
   // Code to be executed
 }
```

Syntax - While Loop

```
let i=1; // Initialization

while(i<=10) { // (i<=0) condition
   console.log(i);
   i++; // iteration
}
```

Syntax - Do While Loop

```
let i=1; // initilization

do{
   console.log(i);
   i++
} while (i<=10) // condition
```

Syntax - For Loop

```
for (let i = 1; i<=10; i++){
   console.log(i);
}
```

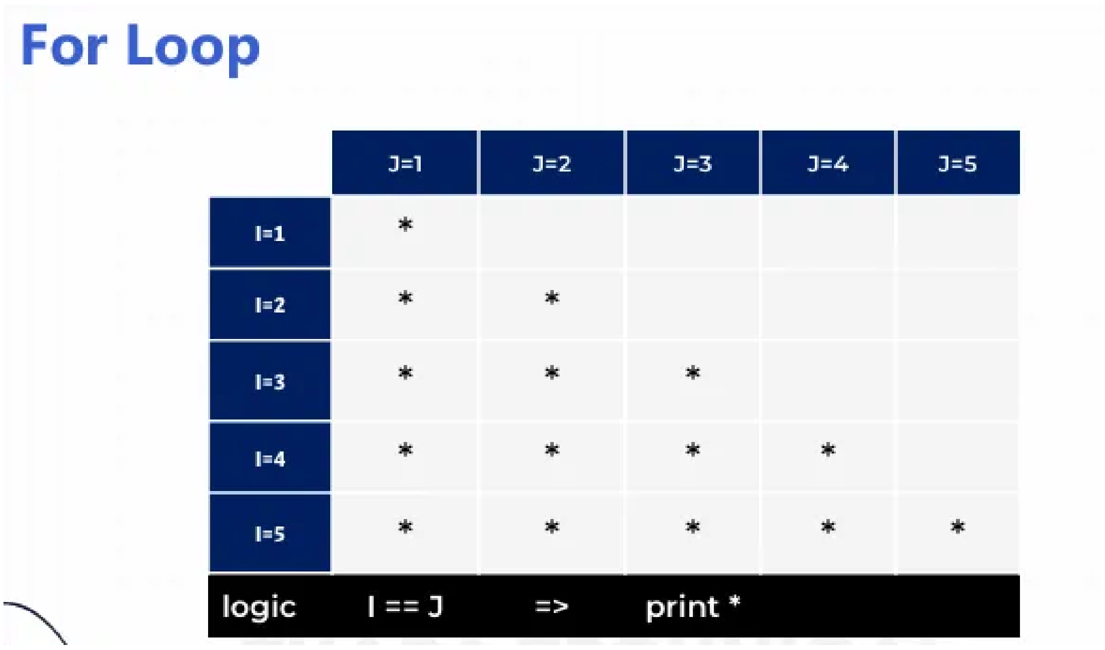
For Loop Task

Program to check if a year is a leap year,

If a year is divisible by 4 and not divisible by a 400, then it is a leap year.

Otherwise, it not a leap year.

For Loop Task -2

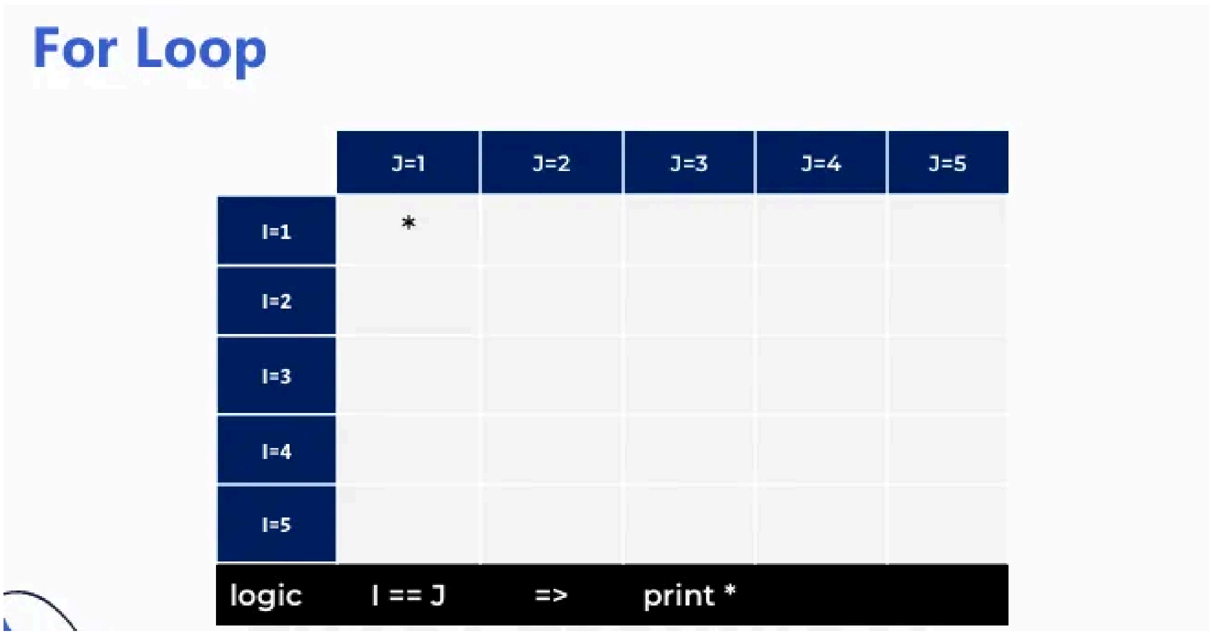


For Loop

	J=1	J=2	J=3	J=4	J=5
I=1	*				
I=2	*	*			
I=3	*	*	*		
I=4	*	*	*	*	
I=5	*	*	*	*	*

logic I == J => print *

For Loop Task - 3



For Loop

	J=1	J=2	J=3	J=4	J=5
I=1	*				
I=2					
I=3					
I=4					
I=5					

logic I == J => print *

JavaScript Functions

In JavaScript, a function is a block of reusable code that performs a specific task or set of tasks. Functions are used to organize code into modular and manageable pieces, promote code reuse, and make programs more readable.

```
function functionName(parameters) {
  // code to be executed
  return result; // optional return statement
 }
```

What we will cover

1. Function Declaration
2. Function Invocation
3. Function Parameter
4. Function Argument
5. Function expressions
6. Anonymous Function
7. Return Keywords
8. IIFE (immediatly invoked function expression)
9. More we will see in advanced.

Syntax - Function Declaration

```
 function greet() {
  console.log("Welcome to The JS Course ");
 }

// ==== Explanation ===
// function = function keyword
// greet = function name
// console.log .... = function Body
```

Syntax - Function Invocation

```
function greet () {
    console.log("hello i'm nazeem khan");
}

greet(); // we need to call the function name

// output: hello i'm nazeem khan
```

Syntax - Function Parameter

```
function greet(parameter1, parameter2){
    console.log("JavaScript is the best!");
}

greet() // call the function
```

Syntax - Function Argument

```
function greet (parameter1, parameter2, ...){
    console.log("Best JS Course");
}

greet(argument1)
```

Syntax - Function Argument

```
function greet(parameter1, parameter2, ...){
 console.log(" Best JS Course ");
}

greet(argument1, argument2)
```

Syntax - Function Argument

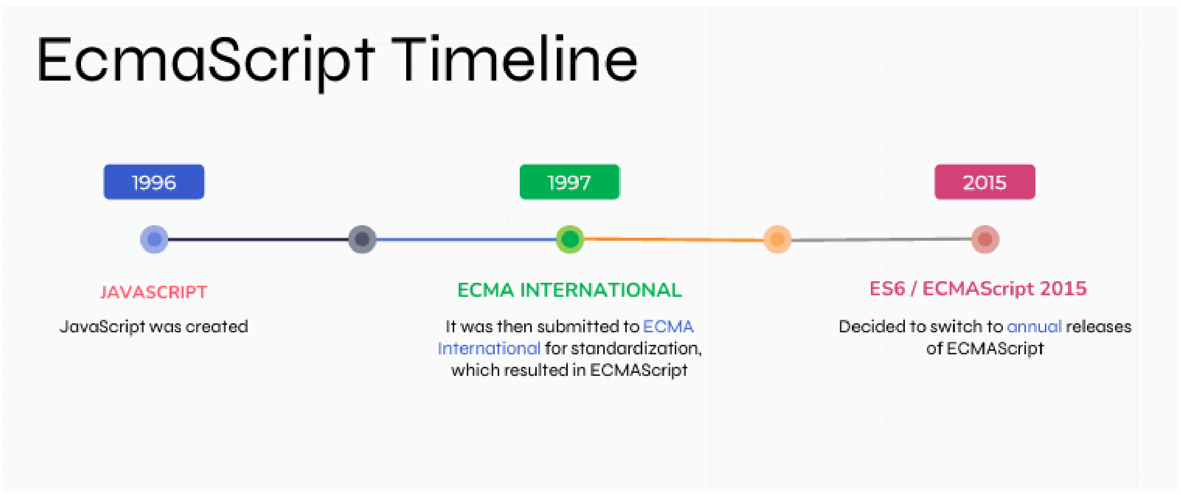
```
function greet(parameter1, parameter2, ...)
{
    console.log(" Best JS Course ");
}

greet(argument1, argument2, ...)
```

Interview Questions - Functions

1: Reverse a String: Write a function to reverse a given string without using built-in reverse methods.
2: Palindrome Check: Create a function to determine if a given string is a palindrome (reads the same backward as forward).
3: Calculator Function: Write a JavaScript function calculator that takes two numbers and an operator as parameters and returns the result of the operation. The function should support addition, subtraction, multiplication, and division.

JavaScript ECMAScript


EcmaScript Timeline
1996
1997
2015
JAVASCRIPT
JavaScript was created
ECMA INTERNATIONAL
It was then submitted to ECMA International for standardization, which resulted in ECMAScript
ES6 / ECMAScript 2015
Decided to switch to annual releases of ECMAScript

ECMAScript 2015 / ES6

Interview Questions-Function

1. **Reverse a String:** Write a function to reverse a given string without using built-in reverse methods.
2. **Palindrome Check:** Create a function to determine if a given string is a palindrome (reads the same backward as forward).
3. **Calculator Function:** Write a JavaScript function calculator that takes two numbers and an operator as a parameter and returns the result of the operation. The function should support addition, subtraction, multiplication, and division.

JavaScript Arrays

Imagine you want to store collection of people names.

ali, sayed, rehman, waqar, tayab,

```
const persons = ["Ali", "Sayed", "Rehman", "Waqar", "Tayab"]
```

That’s what array is for.

JavaScript array is an object that represents a collection of similar type of elements.

Each Value(name) will be called as an Element.

```
// Index Numbers    0       1        2        3        4
const persons =  ["Ali", "Sayed", "Rehman", "Waqar", "Tayab"]
persons[0]; // Ali
persons[1]; // Sayed
```

And we can access each element by using indexes.

```
// Index Number  Lower-Index (0)             Upper-Index(4)
const persons =  ["Ali", "Sayed", "Rehman", "Waqar", "Tayab"]
persons[-1] // Error
persons.at(-1) // Tayab
persons.at(-2) // Waqar
```

First element or head: Refers to the element at index 0.

Last element or tail: Refers to the element at the last index which can be obtained using array.length -1.

ECMAScript 2022 also introduces new .at() method in arrays which helps to index from last elements too easily.

What we Will cover in Arrays

- Creating Arrays / Accessing Elements/ Modifying Elements.
- Array Traversal / Iteration over Arrays
- How to insert, Add, Replace and Delete Elements in Arrays **(CRUD)**
- Searching in an Array
- Filter in an Array
- How to Sort and Compare an Array
- Very Important Array Methods

In Number

```
const persons = ["Ali","Rehman","Sayed", "Tayaab", "Shahid"]
persons[0] // = Ali, 0 is index number which hold the value Ali
```

In Arrays, each element is represented by an index which starts with zero.

Push()

Push Method: The Push Method that adds one or more elements to the end of an array.

Syntax:

```
push(Element)
```

Example:

```
const persons = ["Ali","Rehman","Sayed", "Tayaab", "Shahid","Sana"]
persons.push("Sana")
// Sana will be added at the end
// ["Ali","Rehman","Sayed", "Tayaab", "Shahid","Sana"]
```

Pop()

Pop Method: Method that removes the last element from an array.

Syntax: **pop(**Element**)**

```
const persons = ["Ali","Rehman","Sayed", "Tayaab", "Shahid","Sana"]
persons.pop("Sana") // Sana name will be removed
```

indexOf()

IndexOf Method: The indexOf method returns the first index at which a given element can be found in the array, or -1 if it is not present.

Syntax: **indexOf(SearchElement, fromIndex)**

```
const persons = ["Ali","Rehman","Sayed", "Tayaab", "Shahid"]
persons.indexOf("Shahid") // the index number of Shahid will be returned
```

Includes()

Includes Method: The includes method checks whether an array includes a certain element, returning true or false.

Syntax: **Includes(searchElement , FromIndex)**

```
const persons = ["Ali","Rehman","Sayed", "Tayaab", "Shahid"]
persons.Includes("Haris") // Look where Haris is.
```

Syntax - forEach

```
array.forEach(function
   callback(currentValue, index, array){
      // your logic here
} , ThisValue);
```

Breakdown of each part:

array: The array on which the foreach method is called.

callback: A function that will be called once for each element in the array.

currentValue: the current element being processed in the array.

index(optional): The index of the current element being processed.

array (optional): The array foreach was called upon.

thisValue: (optional): A value to use as this when executing the callback function.

Syntax - forEach

```
array.forEach((currentValue, index, array)=>{
      // your logic here
}, thisValue);
```

Syntax - Map()

```
array.map(function callback(currentValue, index, array){
   // your logic here
}, thisValue);
```

Break down of map

array: the array on which the map method is called.

callback: A function that will be called once for each element in the array.

currentValue: the current element being processed in the array.

index (optional): The index of the current element being processed.

array (optional): The array map was called upon.

thisValue (optional): A value to use as this when executing the callback function.

Interview Questions-Array CRUD

1. Add Dec at the end of an array?
2. what is the return value of splice method?
3. Update march to March (update)?
4. Delete June from an array?

```
Const months=['jan', 'march', 'april', 'june', 'july'];
```

Interview Questions-Array Filter

Q: Given an array of products where each product has a name and a price, write a function that uses the filter method to return an array containing only the products with a price less than or equal to 500.

```
const products=[
    {name:"Laptop", price: 1200},
    {name:"Phone", price: 800},
    {name:"Tablet", price: 300},
    {name:"SmartWatch", price: 150},
];
```

Interview question - Array Reduce

Write a JavaScript function that calculates the total price of items in a shopping cart. The function should take an array of item prices as input and return the total price.

JavaScript Strings

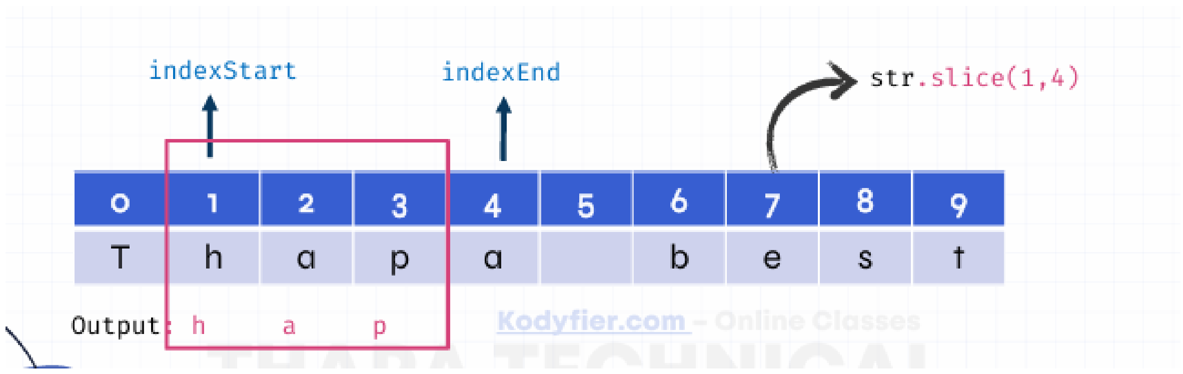
Topics:

- String & it's properties
- Escape character
- String Search Methods
- Extracting String Parts
- Extracting String Characters
- Replacing String Content
- Other Useful Methods

slice()

slice() extracts a part of string and returns the extracted part in a new string.

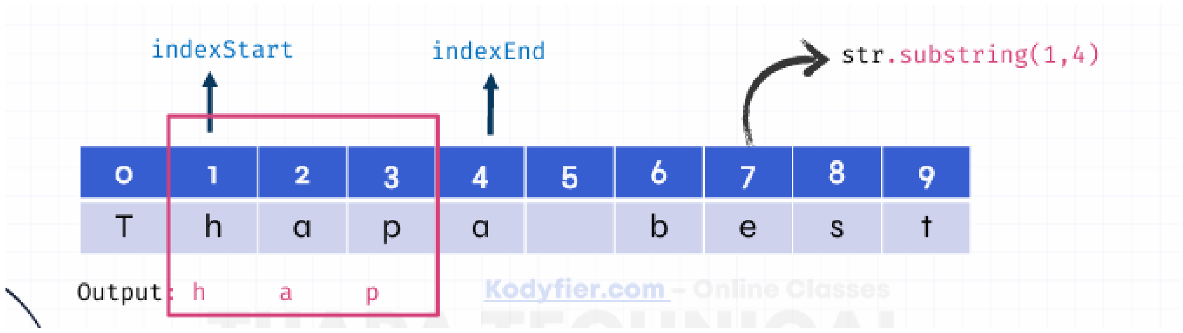
1. JavaScript counts positions from zero.
2. slice() extractsup to but not including indexEnd.



substring()

substring() extracts a part of a string and returns the extracted part in a new string.

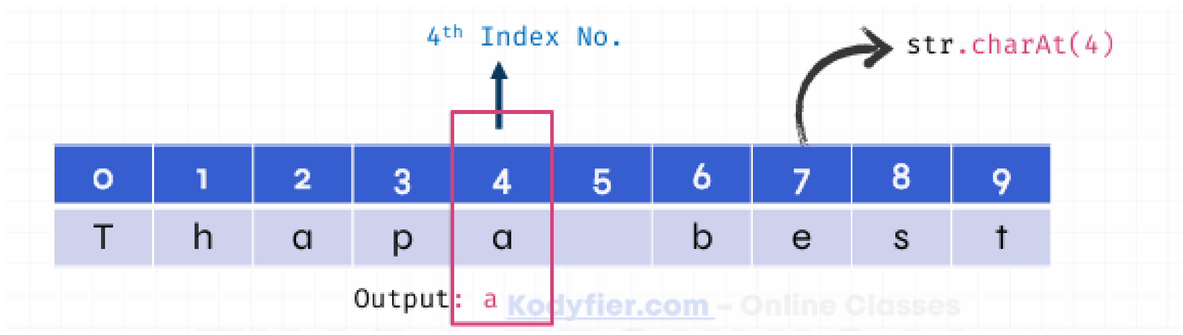
1. JavaScript counts positions from zero.
2. substring() extracts up to but not including indexEnd.



charAt()

The charAt() methods returns the characters at a specified index (position) in a string .

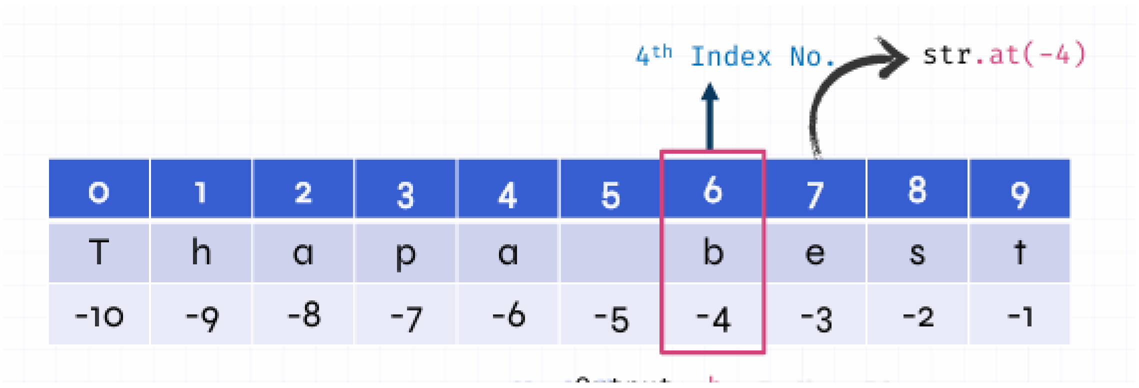
1. JavaScript counts in positions from zero.



at()

The at() method returns the character at a specified index(position) in a string

1. It allows the use of negative indexes while charAt() do not.



Interview Questions - Strings

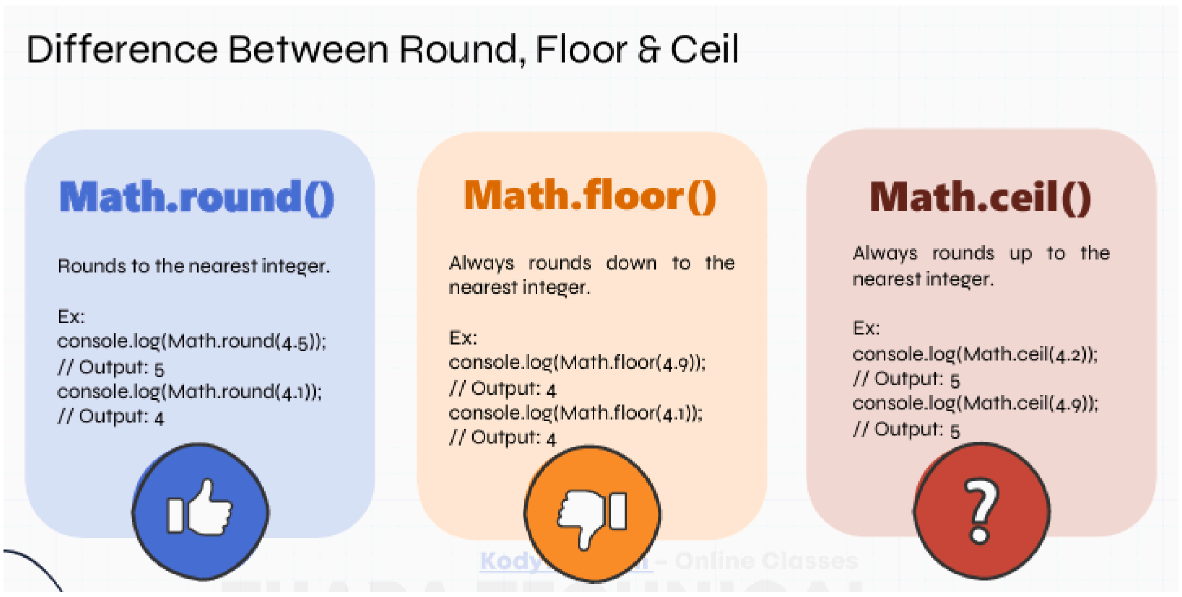
1. Write a JavaScript function that prints the letters 'a' through 'z' in the console. You should use a loop to iterate through the letters and print each one on a new line.
2. Write a function to count the number of vowels in String.
3. Write a function to chec if all the vowels presents in a String or not?

write a JavaScript function isPangram that takes a string as input and returns true if the string is pangram (contains all letters of the alphabet) and false otherwise. The function should be case- insensitive and ignore spaces.

Constraints:

1. The input string will consist of alphabetic characters and spaces.
2. The function should handle both uppercase and lowercase letters.
3. Consider the English alphabet with 26 letters.

Math Objects in JS



Difference Between Round, Floor & Ceil

Math.round()

Rounds to the nearest integer.

Ex:
```
console.log(Math.round(4.5));
// Output: 5
console.log(Math.round(4.1));
// Output: 4
```

Math.floor()

Always rounds down to the nearest integer.

Ex:
```
console.log(Math.floor(4.9));
// Output: 4
console.log(Math.floor(4.1));
// Output: 4
```

Math.ceil()

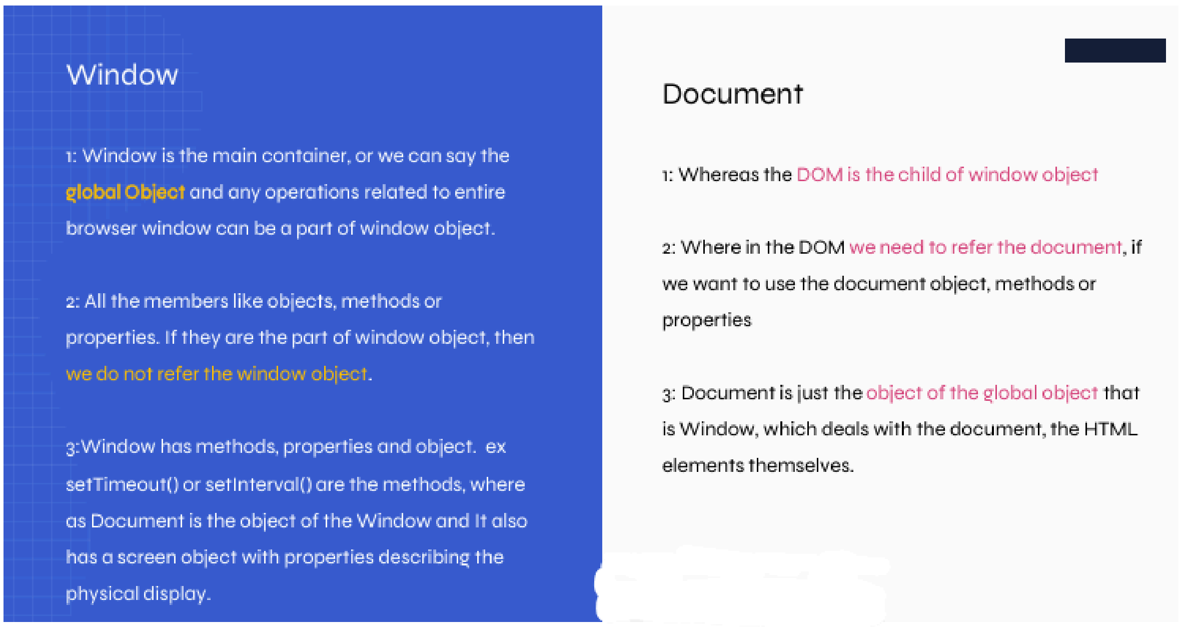
Always rounds up to the nearest integer.

Ex:
```
console.log(Math.ceil(4.2));
// Output: 5
console.log(Math.ceil(4.9));
// Output: 5
```

Interview Questions - Strings & Functions

1: Write a JavaScript function that prints the letters 'a' through 'z' in the console. You should use a loop to iterate through the letters and print each one on a new line.
2: Write a JavaScript function isPangram that takes a string as input and returns true if the string is a pangram (contains all letters of the alphabet) and false otherwise. The function should be case insensitive and ignore spaces.
Constraints:
1: The input string will consist of alphabetic characters and spaces.
2: The function should handle both uppercase and lowercase letters.
3: Consider the English alphabet with 26 letters.

JavaScript Window in JS DOM & BOM



Window

1: Window is the main container, or we can say the global Object and any operations related to entire browser window can be a part of window object.

2: All the members like objects, methods or properties. If they are the part of window object, then we do not refer the window object.

3:Window has methods, properties and object. ex setTimeout() or setInterval() are the methods, where as Document is the object of the Window and It also has a screen object with properties describing the physical display.

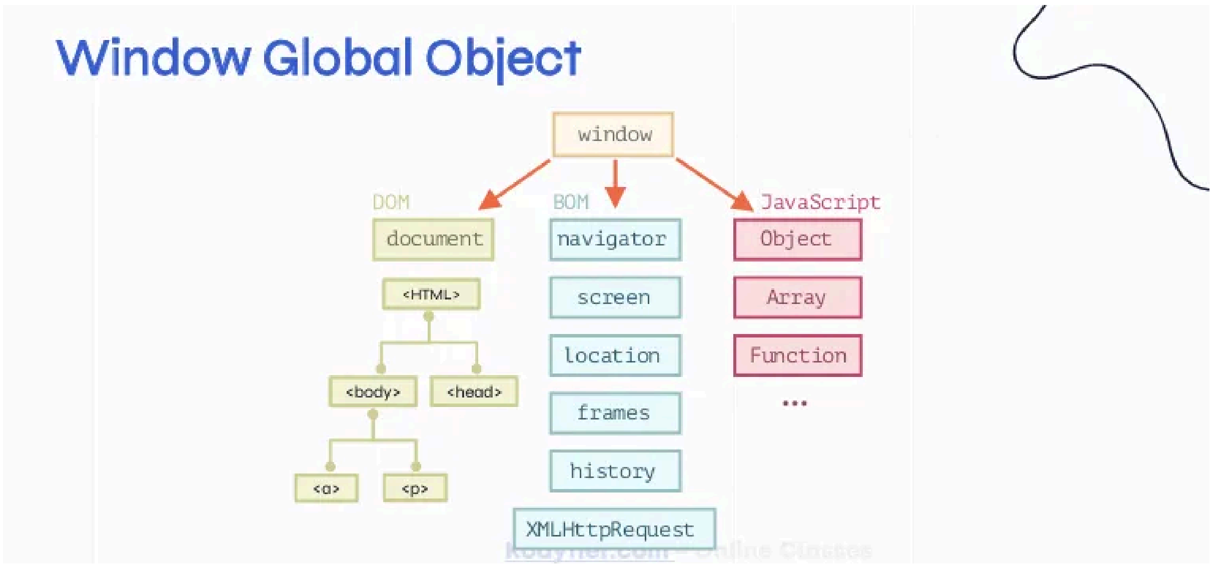
Document

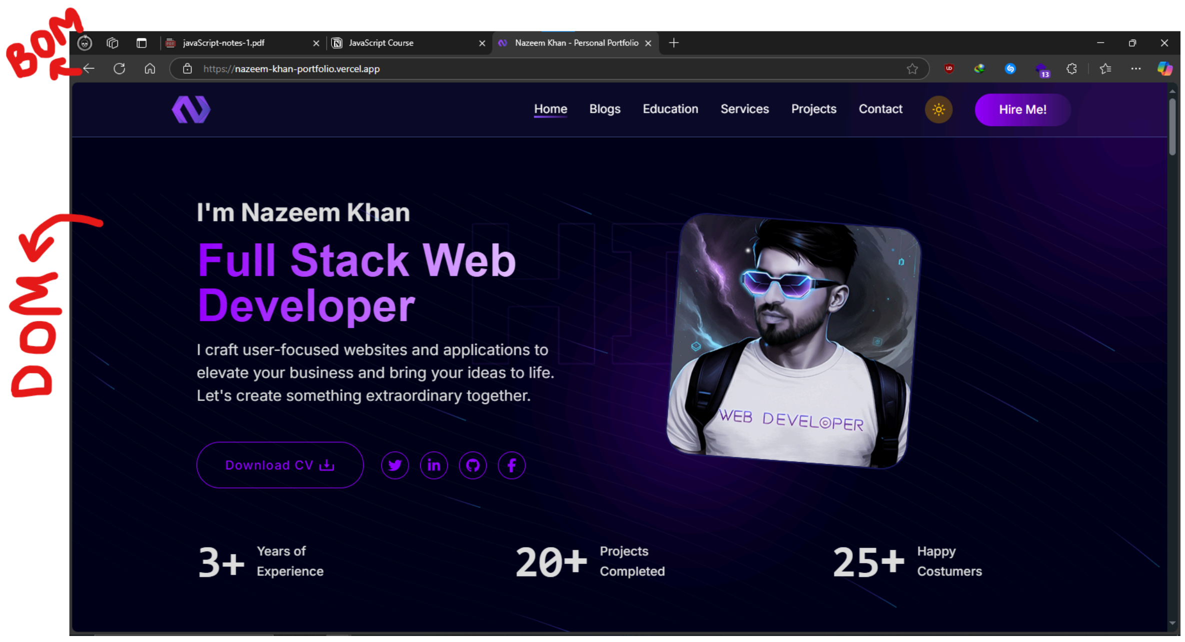
1: Whereas the DOM is the child of window object

2: Where in the DOM we need to refer the document, if we want to use the document object, methods or properties

3: Document is just the object of the global object that is Window, which deals with the document, the HTML elements themselves.

Window Global Object




BOM
DOM
javaScript-notes-1.pdf
JavaScript Course
Nazeem Khan - Personal Portfolio
https://nazeem-khan-portfolio.vercel.app
Home
Blogs
Education
Services
Projects
Contact
Hire Me!
I'm Nazeem Khan
Full Stack Web Developer
I craft user-focused websites and applications to elevate your business and bring your ideas to life. Let's create something extraordinary together.
Download CV
3+ Years of Experience
20+ Projects Completed
25+ Happy Costumers

Window Object:

The window object represents the global window in a browser.
Both the Browser Object Model (BOM) and the Document Object Model (DOM) are part of the window object.

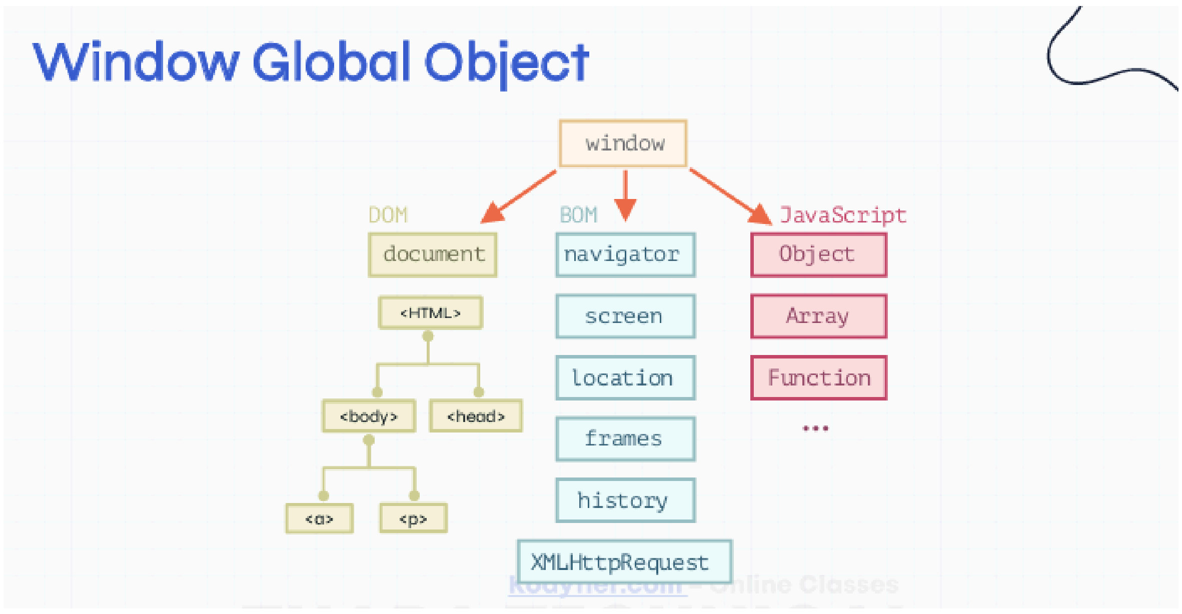
Window Object:
The window object represents the global window in a browser. Both the Browser Object Model (BOM) and the Document Object Model (DOM) are part of the window object.
Browser Object Model (BOM):
The BOM represents the browser as an object and provides methods and properties for interacting with the browser itself (not directly related to the content of a web page).
Examples of BOM features include window.navigator for browser information, window.location for URL manipulation, and window.alert for displaying alerts.

Window Object:
The window object represents the global window in a browser.
Both the Browser Object Model (BOM) and the Document Object Model (DOM) are part of the window object.
Document Object Model (DOM):
The DOM represents the structured document as a tree of objects, where each object corresponds to a part of the document (such as elements, attributes, and text).

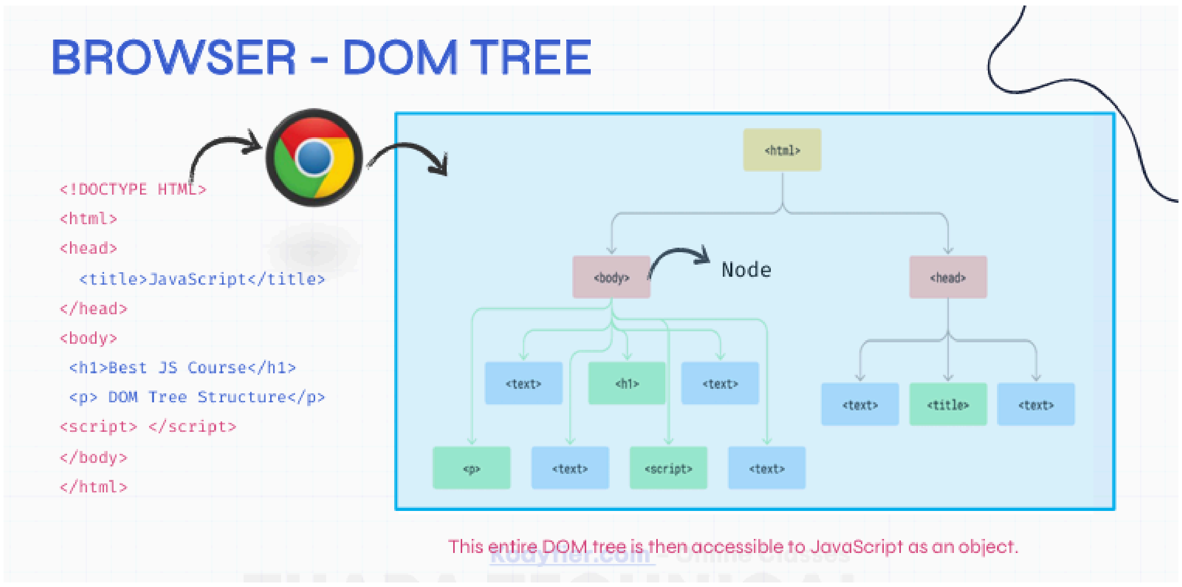
The DOM is primarily concerned with the content of the web page and allows JavaScript to interact with and manipulate the HTML elements

So, while the DOM is focused on the content of the page, the BOM is focused on the browser environment. The window object serves as the global object that encompasses both the BOM and the DOM when working in a browser environment

Window Global Object



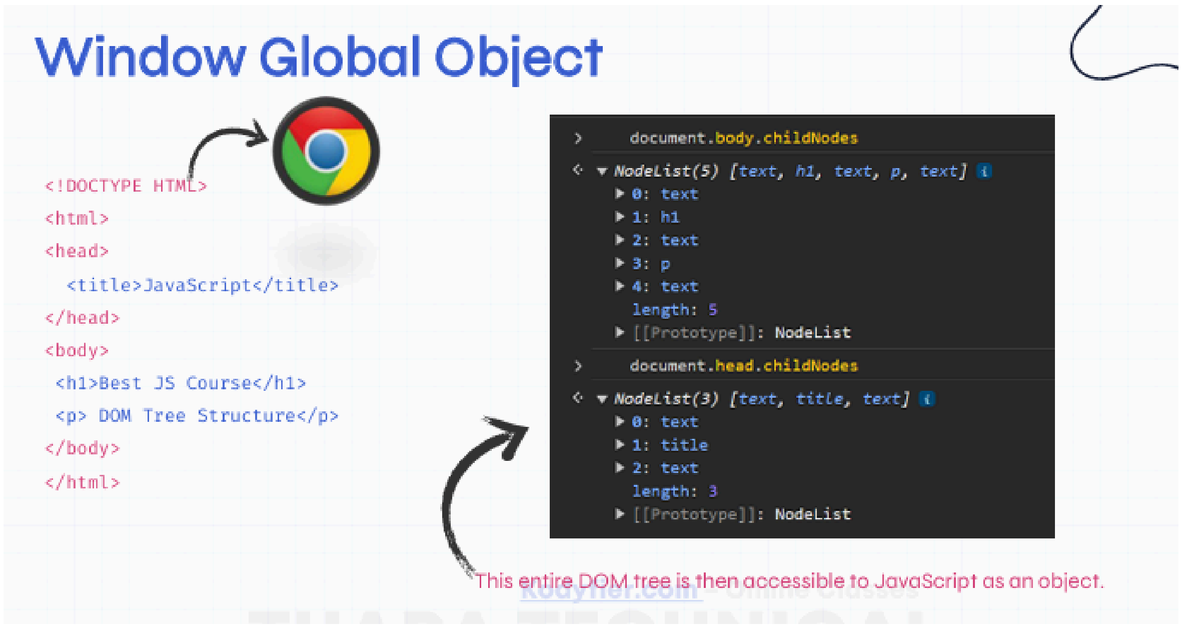
Broswer- DOM tree



```
<!DOCTYPE HTML>
<html>
<head>
  <title>JavaScript</title>
</head>
<body>
 <h1>Best JS Course</h1>
 <p> DOM Tree Structure</p>
<script> </script>
</body>
</html>
```

This entire DOM tree is then accessible to JavaScript as an object.

Window Global Object

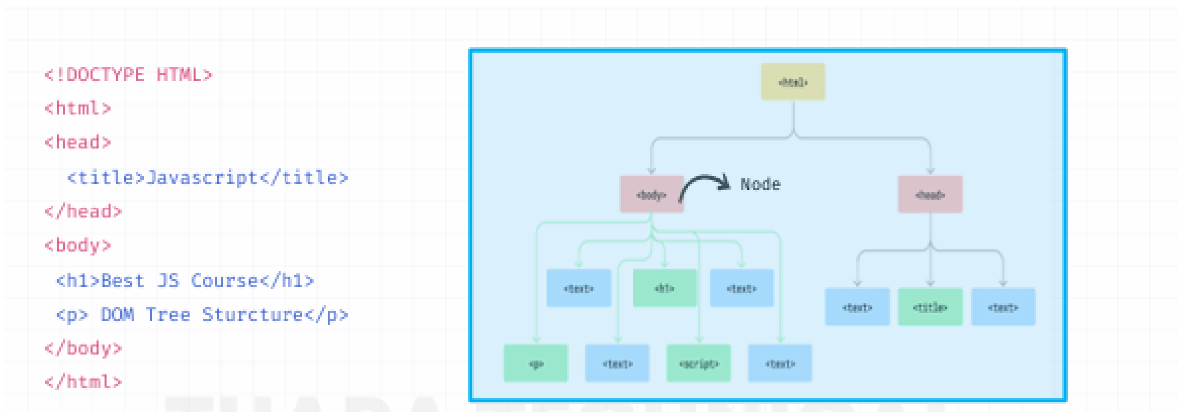


```
<!DOCTYPE HTML>
<html>
<head>
  <title>JavaScript</title>
</head>
<body>
 <h1>Best JS Course</h1>
 <p> DOM Tree Structure</p>
</body>
</html>
```

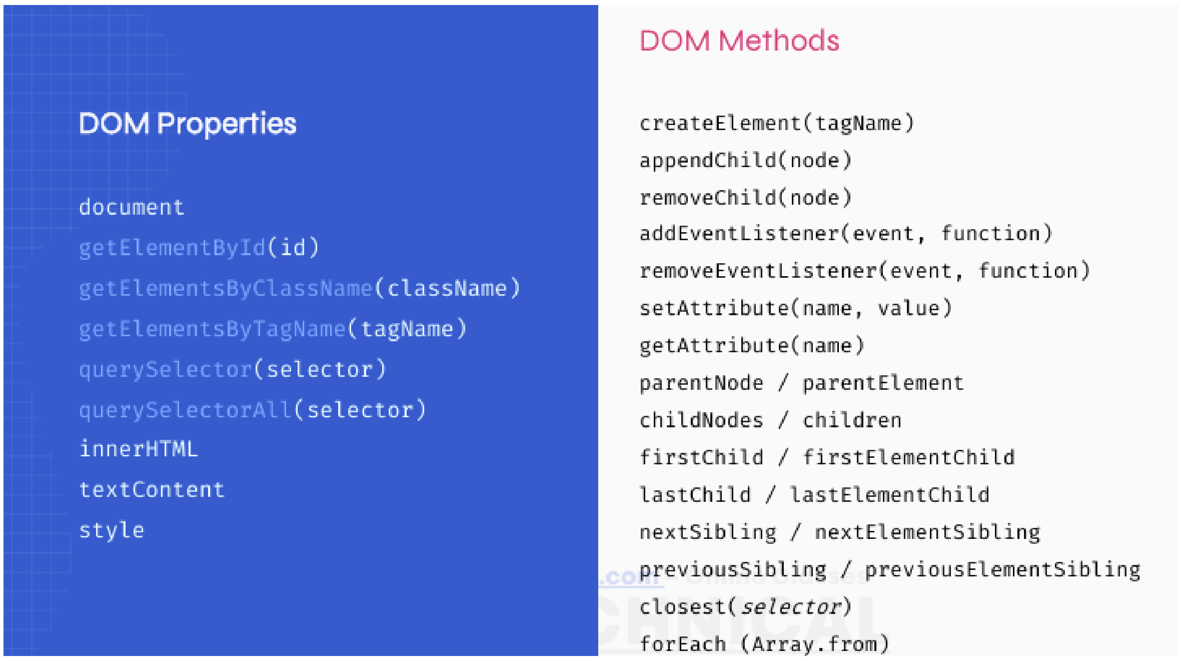
```
>    document.body.childNodes
<  ▼ NodeList(5) [text, h1, text, p, text]
     ▶ 0: text
     ▶ 1: h1
     ▶ 2: text
     ▶ 3: p
     ▶ 4: text
       length: 5
     ▶ [[Prototype]]: NodeList
>    document.head.childNodes
<  ▼ NodeList(3) [text, title, text]
     ▶ 0: text
     ▶ 1: title
     ▶ 2: text
       length: 3
     ▶ [[Prototype]]: NodeList
```

This entire DOM tree is then accessible to JavaScript as an object.

The Document Object Model (DOM) is a tree-like representation of the HTML document. It provides a way to interact with the HTML document using JavaScript. The DOM provides multiple properties and methods to dynamically change the content of the HTML document using JavaScript



```
<!DOCTYPE HTML>
<html>
<head>
  <title>Javascript</title>
</head>
<body>
 <h1>Best JS Course</h1>
 <p> DOM Tree Sturcture</p>
</body>
</html>
```



DOM Properties

```
document
getElementById(id)
getElementsByClassName(className)
getElementsByTagName(tagName)
querySelector(selector)
querySelectorAll(selector)
innerHTML
textContent
style
```

DOM Methods

```
createElement(tagName)
appendChild(node)
removeChild(node)
addEventListener(event, function)
removeEventListener(event, function)
setAttribute(name, value)
getAttribute(name)
parentNode / parentElement
childNodes / children
firstChild / firstElementChild
lastChild / lastElementChild
nextSibling / nextElementSibling
previousSibling / previousElementSibling
closest(selector)
forEach (Array.from)
```

Thank You 😊