



JavaScript Lecture 1 Notes

What is JavaScript?

JavaScript is a programming language used to make websites interactive. It allows you to add dynamic features like animations, buttons, forms, and other actions on web pages that respond to user input. It's widely used alongside HTML and CSS to create modern web applications.

What is meant by ECMA Script?

ECMAScript (often abbreviated as ES) is a scripting language specification that serves as the standard for JavaScript. It defines the core features and functionalities that a scripting language should have, and JavaScript is an implementation of this standard.

History and Purpose

JavaScript was originally created by Brendan Eich at Netscape in the mid-1990s. To ensure that the language remained consistent and widely adoptable across different browsers and environments, Netscape submitted JavaScript to ECMA for standardization. In 1997, the first edition of ECMAScript (ECMAScript 1 or ES1) was published.

Latest version of ECMA Script

The latest version of ECMAScript, as of 2024, is ECMAScript 2024 (ES2024). This edition introduces new features such as the `Object.groupBy()` and `Map.groupBy()` methods, along with new Temporal objects like `PlainDate`, `PlainTime`, and others, which are part of the ongoing efforts to enhance JavaScript's date and time handling.

Keywords and Words

Keywords:

Keywords are predefined words in JavaScript that have special functions or roles in the code. They are part of the language's syntax and help define the structure and behavior of the program. Here are some examples of JavaScript keywords:

if, else: Used for conditional statements.

for, while: Used for loops.

function: Declares a function.

return: Exits a function and possibly returns a value.

var, let, const: Declare variables.

class: Defines a class (introduced in ES6).

Words:

Anything in programming or in JS programming which doesn't have any "meaning" is called WORD like we include the word **wow**, **hi**, **baba**, **chacha** are all words.

What is Variable in JS?

In JavaScript, a variable is a container that holds data values. It allows you to store and manipulate data within your code. You can think of a variable as a "label" for a piece of data that you can use and change throughout your program.

Key Points:

Declaration: You declare a variable using keywords like **var**, **let**, or **const**.

var: Declares a variable that can be updated and has function scope.

let: Declares a block-scoped variable that can also be updated.

const: Declares a block-scoped variable that cannot be reassigned once set.

```
let name = "John"; // 'name' is the variable holding the value "John"
const birthYear = 1990; // 'birthYear' is a constant variable with value 1990
```

Rules for Specifying Variable Names

1. Start with a Letter, Underscore (_), or Dollar Sign (\$):

A variable name must begin with a letter (a-z or A-Z), an underscore (_), or a dollar sign (\$)

```
let name = "Alice"; // Valid
let _score = 100; // Valid
let $price = 25.50; // Valid
```

2. Use Letters, Numbers, Underscores, or Dollar Signs:

After the first character, the variable name can include letters, numbers (0-9), underscores (_), or dollar signs (\$).

```
let user123 = "John"; // Valid
let _temp_var = "temporary"; // Valid
```

3. Case-Sensitive:

JavaScript variable names are case-sensitive, meaning name and Name would be treated as two different variables.

```
let name = "Alice";
let Name = "Bob"; // Different from 'name'
```

4. Cannot Be a Reserved Keyword:

You cannot use JavaScript reserved keywords (e.g., var, let, if, else, etc.) as variable names.

```
let var = 5; // Invalid
let function = "test"; // Invalid
```

5. No Spaces:

Variable names cannot contain spaces. If you need to use multiple words, use camelCase (e.g., userName) or underscores (e.g., user_name).

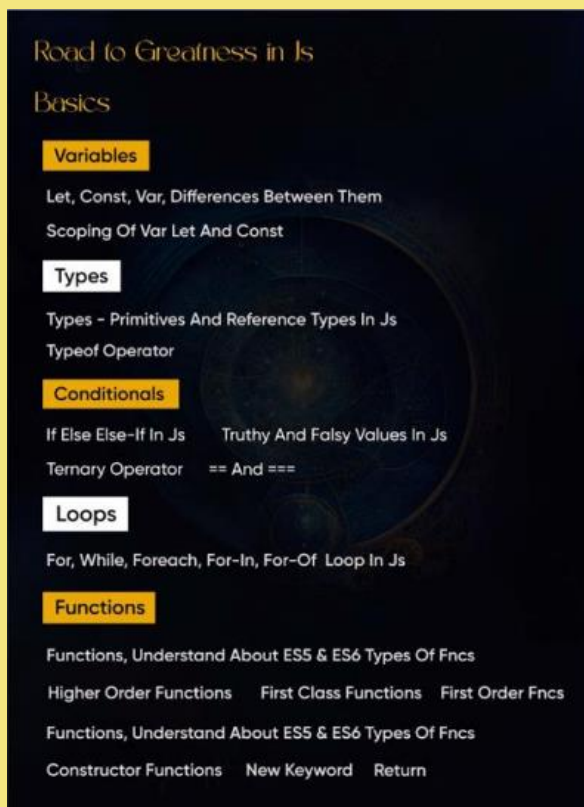
```
let userName = "Alice"; // Valid
let user name = "Alice"; // Invalid
```

6. Avoid Starting with Numbers:

Variable names cannot start with a number, though numbers can be included after the first character.

```
let 1stPlace = "Gold"; // Invalid
let firstPlace1 = "Gold"; // Valid
```

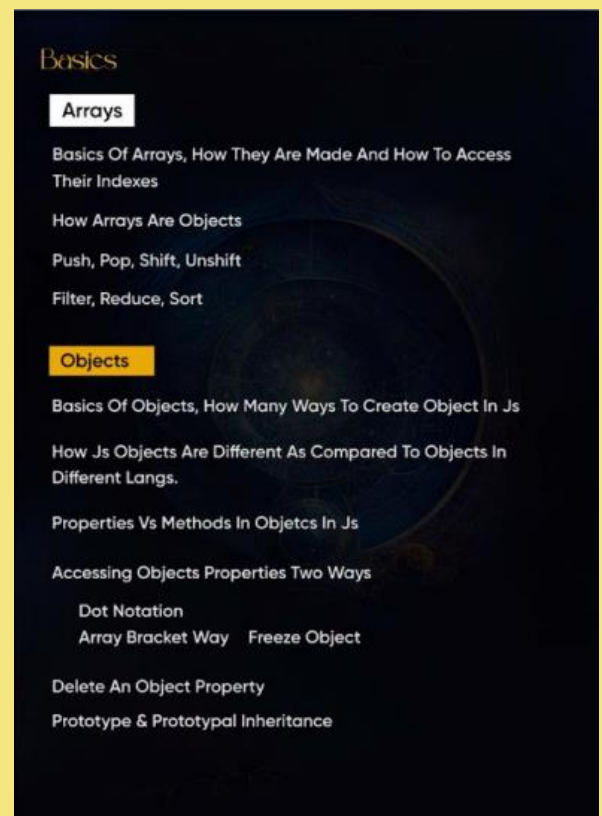
JAVASCRIPT BASIC CONCEPTS



Road to Greatness in Js

Basics

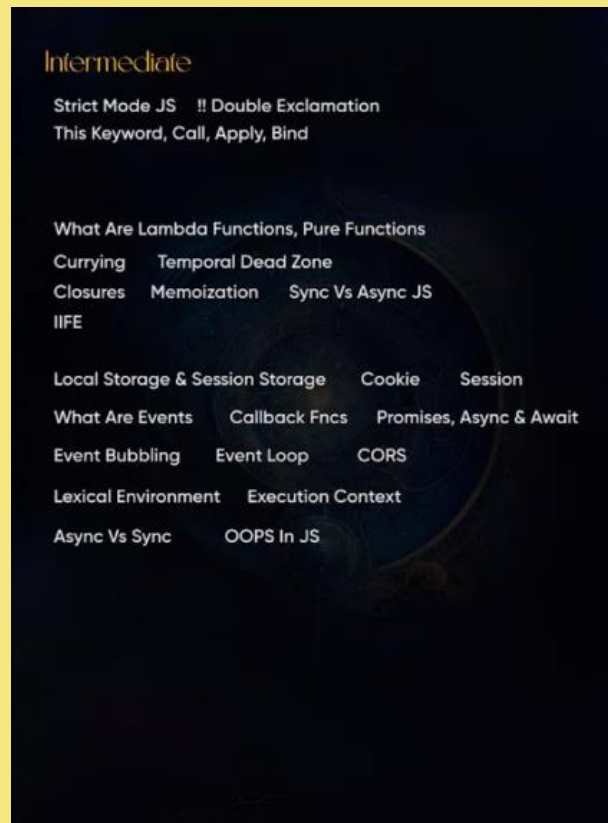
- Variables**
 - Let, Const, Var, Differences Between Them
 - Scoping Of Var Let And Const
- Types**
 - Types - Primitives And Reference Types In Js
 - Typeof Operator
- Conditionals**
 - If Else Else-If In Js
 - Truthy And Falsy Values In Js
 - Ternary Operator
 - == And ===
- Loops**
 - For, While, Foreach, For-In, For-Of Loop In Js
- Functions**
 - Functions, Understand About ES5 & ES6 Types Of Fncs
 - Higher Order Functions
 - First Class Functions
 - First Order Fncs
 - Functions, Understand About ES5 & ES6 Types Of Fncs
 - Constructor Functions
 - New Keyword
 - Return



Basics

- Arrays**
 - Basics Of Arrays, How They Are Made And How To Access Their Indexes
 - How Arrays Are Objects
 - Push, Pop, Shift, Unshift
 - Filter, Reduce, Sort
- Objects**
 - Basics Of Objects, How Many Ways To Create Object In Js
 - How Js Objects Are Different As Compared To Objects In Different Langs.
 - Properties Vs Methods In Objects In Js
 - Accessing Objects Properties Two Ways
 - Dot Notation
 - Array Bracket Way
 - Freeze Object
 - Delete An Object Property
 - Prototype & Prototypal Inheritance

JAVASCRIPT INTERMEDIATE CONCEPTS



JAVASCRIPT ADVANCED CONCEPTS

