JAVASCRIPT VARIABLES

Presented By,

KIRUTHIKA V

Introduction to Variables

What is a JS Variable?

- A variable is a container that stores data values.
- In JavaScript, variables are used to hold information that can be referenced and manipulated.

Example:

```
var x =  5;
var y = 10;
```

Naming Variables

Rules for Naming:

- Must start with a letter, underscore (_), or dollar sign ($).
- Cannot start with a number.
- You declare JavaScript variable with the var keyword.
- Case-sensitive (e.g., a and A are different).

Correct Way:

```
var a=10;
var _name=“Keerthi”;
```

Incorrect Way:

```
var 1a=20;
var *C=50;
```

Declaring Variables

Keywords Used for Declaration:

- var
- let
- const

var Keyword

- Variables declared with var are function-scoped.
- They can be re-assigned and re-declared within their scope.

Example:

```
var a = 10;
var b = 15;
var c = a + b;
```

let Keyword

- Block-scoped variable, meaning it exists only within the {} block where it is defined.
- let variables can be re-assigned but not re-declared in the same scope.

Example:

```
let mark = 80;
mark = 95;
```

const Keyword

- Variables declared with const are block-scoped and cannot be re-assigned.
- Best used for values that should remain constant.

Example:

Correct Way:

```
const PI = 3.14159;
```

Incorrect Way:

```
const PI;
PI = 3.14159;
```

Differences between var, let, and const

var

- Function-scoped.
- Can be re-declared within the same scope.
- Can be re-assigned

let

- Block-scoped.
- Cannot be re-declared within the same scope.
- Cannot be re-assigned

const

- Block-scoped.
- Cannot be re-declared within the same scope.
- Cannot be re-assigned

Redeclaring Variables

Using var:

- Redeclaring a variable inside a block will also redeclare the variable outside the block.

Example:

```
var a = 5;
//Here x is 5.
{
var x = 15;
//Here x is 15.
}
//Here x is 15.
```

Using let, const:

- Redeclaring a variable inside a block will not also redeclare the variable outside the block.

Example:

```
let a = 5;
//Here x is 5.
{
let x = 15;
//Here x is 15.
}
//Here x is 5.
```

Undefined Value

Undefined Value:

- A Variable declared without a value will have the value undefined.

Example:

```
let  x;
let y;
```

One Statement , Many Variables

- You can declare many variables in one statement.

Example:

```
let  _fname=“Keerthi” , _lname=“V” , age=22;
```

Variable Scope

What is Scope?

- Scope determines where variables can be accessed.

Types of Scope

Global Scope:

- Declared outside of any function/block; accessible anywhere.

Function Scope:

- Variables inside a function; not accessible outside.

Block Scope:

- Only variables declared inside a block(e.g., within {}); cannot be accessed outside a block.

Example of Global and Block Scope

- Example of Global Scope:

```
{
    var x=3;
}
// x can be used here.
```

- Example of Block Scope:

```
{
    let x = 3;
}
// x can not be used here.
```

Dynamic Typing in JavaScript

Dynamic Typing:

- Explain that JavaScript variables can change types based on assigned values.

Example:

```
let x = 5;
 x = "Hello";
```

Hoisting Variables

- var Hoisting:
  - Variables defined with var are hoisted to the top and can be initialized at any time.

Example:

```
name = “keerthi”;
var name;
```

- let Hoisting:
  - Variables defined with let are also hoisted to the top of the block, but not initialized.

Example:

```
name = “keerthi”;
let name;
```

- const Hoisting:
  - Variables defined with const are also hoisted to the top, but not initialized.

Example:

```
name = “keerthi”;
const name;
```

Example: Using Variables

Example Code:

```
let f_name = “Keerthi";
let l_name = “V";
const birthYear = 2002;
let currentYear = 2024;
let age = currentYear - birthYear;
document.write(f_name+l_name+”is”+age+”years old.”);
```

Output: Keerthi V is 22 years old.

THANK YOU