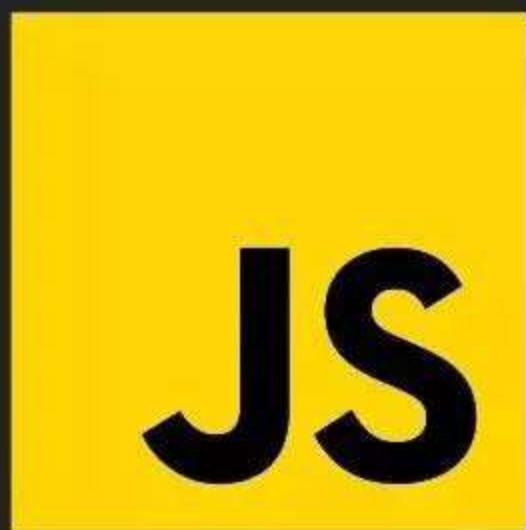




Save it



Like it



# Es6 **Guide**

Part 01



@coder\_aishya



# Let & const Keywords

- Variables were previously declared using **"var"** which had **function scope** and were hoisted to the top within its scope. It means that **a variable can be used before declaration.**
- But, the **"let"** variables and constants have **block scope** which is surrounded by curly-braces **"{}"**, they are **not hoisted** & **cannot be used before declaration.**
- The **new const** keyword makes it **possible to define constants.** Constants are read-only, you **cannot reassign new values to them.**



## Let example:



```
let x = 10; // Here x is 10

{
  let x = 2; // Here x is 2
}

document.getElementById("demo").innerHTML = x;
//Output : 10
```

## const example:



```
var x = 10; // Here x is 10
{
  const x = 2; // Here x is 2
}

document.getElementById("demo").innerHTML = x;
//Output : 10
```





# Arrow Functions

- It provides a more concise syntax for writing function expressions by removing the "function" and "return" keywords.
- Arrow functions are defined using the fat arrow (`=>`) notation.
- Unlike ordinary functions, arrow functions do not have their own `this` keyword.
- The value of `this` inside an arrow function is always bound to the value of `this` in the closest non-arrow function.
- **Arrow functions are not hoisted.** They must be defined before they are used.



## Arrow function Example :



```
// ES5 Function Expression
var sum = function(a, b) {
    return a + b;
}
console.log(sum(2, 3)); // Output : 5

// ES6 Arrow function
var sum = (a, b) => a + b;
console.log(sum(2, 3)); // Output : 5
```



# Multi-line Strings

Users can create multi-line strings by using back-ticks(`). In ES5 we needed to use '\n' for multi line statements

```
//ES5 Sytax
```

```
var poemData = 'Johnny Johnny Yes Papa,\n                + 'Eating sugar? No, papa!\n                + 'Telling lies? No, papa!\n                + 'Open your mouth Ah, ah, ah!'
```

```
//ES6 syntax
```

```
let poemData = `Johnny Johnny Yes Papa,\n                Eating sugar? No, papa!\n                Telling lies? No, papa!\n                Open your mouth Ah, ah, ah!`
```





# Template Literals

ES6 introduces very simple string templates along with **placeholders for the variables**.

The syntax for using the string template is **`${PARAMETER}`** and is used inside of the back-ticked string.



```
//ES5 Syntax
```

```
var name = 'Your name is ' + firstName + ' ' + lastName + '.'
```

```
//ES6 syntax
```

```
var name = `Your name is ${firstName} ${lastName}.`
```



@coder\_aishya



# Default Parameters

ES6 allows **function parameters to have default values**. But, in ES5, OR operator had to be used.



```
//ES5 syntax
```

```
var calculateArea = function(height, width) {  
  height = height || 50;  
  width = width || 80;  
  // logic  
}
```

```
//ES6 syntax
```

```
let calculateArea = function(height = 100, width = 50) {  
  // logic  
}
```

