

Variables

<code>let x = 10;</code>	block scope	most used
<code>const MAX = 100;</code>	constant values	
<code>var y = 10;</code>	global or function scope	old usage

Operators

<code>let x = 10;</code>	assignment
<code>x == " Hello"</code>	comparison
<code>x === " Hello"</code>	strict comparison
<code>&&</code>	logical AND
<code> </code>	logical OR
<code>!</code>	logical negation

Data Types

<code>let num = 10;</code>	Number
<code>let str = " Hello";</code>	String
<code>let isTrue = true</code>	Boolean
<code>let arr = [1, 2, 3]</code>	Array
<code>let obj = { a:1, b:2, c:" Hello" }</code>	Object

Array

<code>let array = ["a", " b","c "]</code>	array definition
<code>array.length //3</code>	number of elements in array
<code>array[0] // a</code>	access first element
<code>array[n] //0=<n <array.length</code>	access n th element
<code>array[4] // error</code>	4 grater then length of array
<code>array[array.length - 1] // c</code>	access last element

Comments

```
// this is a one line comment

/* this is a multiple
   lines comment
   also called block comment */
```

Object

```
let object = {
  name: " Alice",
  age: 30
}
```

Unlike JSON, a JavaScript object does not have double quotes (") in the property name

Object usage

<code>object.name // Alice</code>	dot notation
<code>object["name"] // Alice</code>	property access notation
<code>object.age // 30</code>	
<code>object.gender // undefined</code>	non-existent property

try...catch...finally

```
try{
  //code might throw an error
} catch(err) {
  console.e rro r("Error occur ed",
err.me ssage);
} finally {
  console.l og( " Ope ration comple ted ");
}
```

try: This block contains code that might throw an exception (an error)
 catch: This block handles the exception if an error occurs in the try block

finally: This block always executes, regardless of whether an error occurs or not

Function

```
function greet() {
  console.l og( " Hello")
}
```

function definition

Function with parameter

```
function greet(name="Alice") {
  console.l og( " Hello")
}
```

parameter name with default value "Alice"

