

| Variables

```
// Declaring variables using `let`, `const`, or `var`:  
let variableName = value; // Block-scoped variable  
const constantName = value; // Constant block-scoped variable  
                             (cannot be reassigned)  
var variableName = value; // Function-scoped or globally-scoped  
                             variable (older syntax)
```

| Operators

```
a + b // addition  
a - b // subtraction  
a * b // multiplication  
a / b // division  
a % b // modulus  
a === b // strict equality  
a !== b // strict inequality  
a > b // greater than  
a < b // less than  
a >= b // greater than or equal to  
a <= b // less than or equal to  
a && b // logical AND  
a || b // logical OR  
!a // logical NOT
```

| Data Types

```
string: 'hello' or "hello" // string literal  
number: 42 // numeric literal  
boolean: true or false // boolean literal  
array: [1, 2, 3] or ['a', 'b', 'c'] // array literal  
object: { key: 'value', key2: 'value2' } // object literal  
null  
undefined
```

| Conditionals

```
// If statement
if (condition) {
    // code to execute if condition is true
}

// If-else statement
if (condition) {
    // code to execute if condition is true
} else if (anotherCondition) {
    // code to execute if anotherCondition is true
} else {
    // code to execute if none of the above conditions are true
}

// Switch statement
switch (expression) {
    case value1:
        // code to be executed
        break;
    case value2:
        // code to be executed
        break;
    default:
        // code to be executed
}
```

| Loops

```
// For loop
for (init: let i = 0; condition: i < array.length; increment: i++)
{
    // code to be executed
}

// For-of loop
for (let element of array) {
    // code to be executed
}

// While loop
let i = 0;
while (i < array.length) {
    // code to be executed
    i++;
}

// Do-while loop
let i = 0;
do {
    // code to be executed
    i++;
} while (i < array.length);

// Breake and Continue
for (let i = 0; i < array.length; i++) {
    if (array[i] === 3) {
        break; // Exit the loop entirely when 3 is found }
    }
for (let i = 0; i < array.length; i++) {
    if (array[i] === 3) {
        continue; // Skip this iteration when 3 is found }
    }

break; // break out of a loop or switch
continue; // skip to the next iteration of a loop
```

| Functions

```
// Function definition with parameters
function functionName(parameters) {
    // code to execute when the function is called
}

functionName(arguments); // Function call with arguments
// 'arguments' are the actual values that are supplied to the
// 'parameters'

// Assigning a function to a variable
const functionName = function(parameters) {
    // code to execute when function is called
};

// Arrow Function | Shorthand for the above code
const functionName = (parameters) => {
    // code to execute
};

// Can be done without assigning to a variable
button.addEventListener('click', () => {
    console.log('Button clicked!');
});
```

| Arrays

```
// array declaration
const array = [element1, element2, element3 ...];
//Elements' respective index number; starting from 0
const array = [0, 1, 2, ...];
//Accessing an array element by index
let firstElement = arrayName[0];
// Common array methods
array.push(element); // add an element to the end of an array
array.pop(); // remove the last element of an array
array.shift(); // remove the first element of an array
array.unshift(element); // add an element to the beginning of an array
array.length; // get the length of an array
array[index]; // access an element of an array by index
```

| Objects

```
// object declaration | Last one doesn't need a ',' comma
const object = {
  key1: value,
  key2: value2,
  key3: value3,
  ... };

object.key; // access a property of an object
object.key = value; // set a property of an object
delete object.key; // delete a property of an object

// an object can have multiple objects and functions nested inside.

let objectName = {
  key1: value1,
  key2: value2,
  methodName: function() {
    // method code
  }
};
```

| String Methods

```
// Common string methods
let text = "Hello, World!";
text.length; // Returns the length of the string
text.toUpperCase(); // Converts to uppercase
text.toLowerCase(); // Converts to lowercase
text.indexOf("World"); // Returns the index of the first
occurrence of "World"
text.slice(0, 5); // Extracts a portion of the string

// Template Literals
let name = "John";
let greeting = `Hello, ${name}!`; // Output: Hello, John!
```

| Console

```
console.log("Message"); // Logs a message to the console
```

| DOM and Events

```
document.getElementById('id'); // get an element by ID
document.querySelector('selector'); // get an element by CSS
selector
element.addEventListener('event', function); // add an event
listener to an element
element.removeEventListener('event', function); // remove an event
listener from an element
```