ES6 Features

ES6 (ECMAScript 2015) features.

# let and const

let – Block scoped, can be updated but not re-declared.

const – Block scoped, cannot be updated or re-declared.

Avoid var unless necessary (function scoped, hoisted).

**Example**:

let x = 10;

const y = 20;

# Arrow Functions

Shorter syntax for function expressions.

Lexical this binding (no own this).

**Example**:

const add = (a, b) => a + b;

# Template Literals

Multi-line strings and interpolation using backticks (`).

**Example**:

const name = "Venkatesh";

console.log(Hello, ${name}!);

# Destructuring

Extract values from arrays or objects into variables.

**Example:**

// Object

const user = { name: "Ana", age: 25 };

const { name, age } = user;

// Array

const [a, b] = [1, 2];

# Default Parameters

Set default values for function parameters.

**Example**:

function greet(name = "Guest") {

console.log(Hello, ${name});

}

# Spread & Rest Operator (...)

**Spread**: Expand arrays/objects.

**Rest**: Collect arguments into an array.

**Example**:

const arr1 = [1, 2];

const arr2 = [...arr1, 3]; // Spread

function sum(...nums) { // Rest

return nums.reduce((a, b) => a + b, 0);

}

# Enhanced Object Literals

Shorthand for properties and methods.

**Example**:

const name = "JS";

const obj = {

name, // same as name: name

greet() { return "Hi"; }

};

# Promises

Handle async operations with .then() and .catch().

**Example**:

const fetchData = () =>

new Promise((resolve, reject) => {

resolve("Data loaded");

});

fetchData().then(console.log).catch(console.error);

# Classes

Syntactic sugar over prototypes.

**Example**:

class Person {

constructor(name) {

this.name = name;

}

greet() {

return Hello, ${this.name};

}

}

# Modules

Export/import code between files.

**Example**:

// module.js

export const PI = 3.14;

// main.js

import { PI } from './module.js';

# for...of Loop

Iterate over iterables like arrays or strings.

**Example**:

for (const val of [1, 2, 3]) {

console.log(val);

}

# Map and Set

**Map**: key-value pairs, maintains order.

**Set**: unique values, no duplicates.

**Example**:

const map = new Map();

map.set("a", 1);

const set = new Set([1, 2, 2]);