Chapter 3

Working with Variables and Data Type

Example:

var message; // Declaration

message = "Hello, Rupali!"; // Initialization

console.log(message); // Output: Hello, Rupali!

var number = 10; // Declaration and initialization

console.log(number); // Output: 10

var number = 20; // Redeclaring a variable with var

console.log(number); // Output: 20

Example:

let name; // Declaration

name = "Raju"; // Initialization

console.log(name); // Output: Raju

let age = 25; // Declaration and initialization

console.log(age); // Output: 25

// let age = 30; // Error: Identifier 'age' has already been declared in the same scope

age = 30; // Reassigning a new value (allowed)

console.log(age); // Output: 30

Example:

const pi = 3.14; // Declaration and initialization

console.log(pi); // Output: 3.14

// pi = 3.1415; // Error: Assignment to constant variable

// const gravity; // Error: Missing initializer in const declaration

Example Combining All Three

function example() {

var x = 10; // Function-scoped

if (true) {

let y = 20; // Block-scoped

const z = 30; // Block-scoped, cannot be reassigned

console.log(x, y, z); // Output: 10, 20, 30

}

console.log(x); // Output: 10

// console.log(y); // Error: y is not defined

// console.log(z); // Error: z is not defined

}

example();

* + **Example:**

let message = "Hello, World!";

console.log(message); // Output: Hello, World!

* + **Example:**

let score = 98.5;

console.log(score); // Output: 98.5

* + **Example:**

let isAvailable = true;

console.log(isAvailable); // Output: true

* + **Example:**

let x;

console.log(x); // Output: undefined

* + **Example:**

let empty = null;

console.log(empty); // Output: null

* + **Example:**

let sym = Symbol("id");

console.log(sym); // Output: Symbol(id)

* + **Example:**

let bigNumber = BigInt(123456789012345678901234567890);

console.log(bigNumber); // Output: 123456789012345678901234567890n

* + **Example:**

let user = { name: "Rupali", age: 25 };

console.log(user.name); // Output: Rupali

* + **Example:**

let colors = ["red", "green", "blue"];

console.log(colors[0]); // Output: red

* + **Example:**

function greet(name) {

return `Hello, ${name}!`;

}

console.log(greet("Bob")); // Output: Hello, Bob!

let x = 10;

let y = x; // Copy of x

y = 20;

console.log(x); // Output: 10

console.log(y); // Output: 20

**Reference Types:**

let obj1 = {name: "Rupali"};

let obj2 = obj1; // Reference to the same object

obj2.name = "Monali";

console.log(obj1.name); // Output: Monali

console.log(obj2.name); // Output: Monali

Values cannot be changed; any modification creates a new value.

let str = "Hello";

str = str + " Rupali";

console.log(str); // Output: Hello Rupali

let numbers = [1, 2, 3];

numbers.push(4); // Modifying the array

console.log(numbers); // Output: [1, 2, 3, 4]

**Primitive Example:**

let a = 50;

let b = a; // Copy the value of 'a'

b = 100;

console.log(a); // Output: 50 (original remains unchanged)

console.log(b); // Output: 100

**Reference Example:**

let obj1 = { value: 10 };

let obj2 = obj1; // Reference the same object

obj2.value = 20;

console.log(obj1.value); // Output: 20 (both point to the same memory location)

* + **Example:**

let num = 123;

let str = String(num);

console.log(str); // Output: "123"

console.log(typeof str); // Output: string

* + **Example:**

let str = "456";

let num = Number(str);

console.log(num); // Output: 456

console.log(typeof num); // Output: number

* + **Example:**

let value = 0;

let bool = Boolean(value);

console.log(bool); // Output: false

* + **Example:**

let result = 5 + " apples";

console.log(result); // Output: "5 apples"

* + **Example:**

let result = "10" - 2; // Subtraction forces "10" to become 10

console.log(result); // Output: 8

* + **Example:**

if ("text") {

console.log("Truthy!"); // Output: "Truthy!" (non-empty strings are truthy)

}

* **Example:**

console.log(5 == "5"); // Output: true

* **Example:**

console.log(5 === "5"); // Output: false

* **Example:**

console.log("10" > 5); // Output: true

let value = "123";

let number = Number(value); // Explicitly convert string to number

console.log(number + 1); // Output: 124

let value = "123";

console.log(value + 1); // Output: "1231" (String concatenation due to coercion)

* + **Example:**

if (0) {

console.log("Won't run");

} else {

console.log("Falsy!"); // Output: "Falsy!"

}

* + **Example:**

console.log(Number("abc")); // Output: NaN

console.log(NaN == NaN); // Output: false

* + **Example:**

let obj = { name: "Alice" };

console.log(String(obj)); // Output: "[object Object]"

console.log("10" \* "2"); // Output: 20

console.log("10" + "2"); // Output: "102" (string concatenation)

console.log(false == 0); // Output: true

console.log(null == undefined); // Output: true

Declaration Examples:

let singleQuote = 'Hello, World!';

let doubleQuote = "JavaScript Strings";

let templateLiteral = `Template Literals`;

let text = "Hello, World!";

console.log(text.length); // Output: 13

toUpperCase() and toLowerCase()

let message = "MyIndia";

console.log(message.toUpperCase()); // Output: "MYINDIA"

console.log(message.toLowerCase()); // Output: "myindia"

charAt(index)

let str = "Hello";

console.log(str.charAt(1)); // Output: "e"

indexOf(searchValue)

let text = "Hello, World!";

console.log(text.indexOf("World")); // Output: 7

lastIndexOf(searchValue)

let text = "Hello, Hello!";

console.log(text.lastIndexOf("Hello")); // Output: 7

* Using the + operator:

let firstName = "Rupali";

let lastName = "Chopade";

console.log(firstName + " " + lastName); // Output: "Rupali Chopade"

* Using concat() method:

let str1 = "Hello";

let str2 = "Rupali";

console.log(str1.concat(", ", str2)); // Output: "Hello, Rupali"

Substring Extraction

let text = "JavaScript";

console.log(text.slice(0, 4)); // Output: "Java"

* substring(start, end):

let text = "JavaScript";

console.log(text.substring(0, 4)); // Output: "Java"

* substr(start, length) (deprecated):

let text = "JavaScript";

console.log(text.substr(0, 4)); // Output: "Java"

* split(separator):

let text = "apple,banana,cherry";

let fruits = text.split(",");

console.log(fruits); // Output: ["apple", "banana", "cherry"]

includes(searchValue)

let text = "Hello, World!";

console.log(text.includes("World")); // Output: true

startsWith(searchValue) and endsWith(searchValue)

let text = "JavaScript";

console.log(text.startsWith("Java")); // Output: true

console.log(text.endsWith("Script")); // Output: true

replace(searchValue, newValue)

let text = "Hello, World!";

console.log(text.replace("World", "JavaScript")); // Output: "Hello, JavaScript!"

replaceAll(searchValue, newValue)

let text = "Hello, Hello!";

console.log(text.replaceAll("Hello", "Hi")); // Output: "Hi, Hi!"

let name = "Rupali";

let greeting = `Hello, ${name}!`;

console.log(greeting); // Output: "Hello, Rupali!"

let message = `This is a

multiline string.`;

console.log(message);

Example:

let text = " Hello, World! ";

console.log(text.trim()); // Output: "Hello, World!"

Use split() to break a string into an array based on a specified delimiter.

let text = "a,b,c,d";

let array = text.split(",");

console.log(array); // Output: ["a", "b", "c", "d"]

Example 1: Checking Palindrome

function isPalindrome(str) {

let reversed = str.split("").reverse().join("");

return str === reversed;

}

console.log(isPalindrome("madam")); // Output: true

console.log(isPalindrome("hello")); // Output: false

Example 2: Word Frequency Counter

function wordFrequency(sentence) {

let words = sentence.toLowerCase().split(" ");

let frequency = {};

words.forEach(word => {

frequency[word] = (frequency[word] || 0) + 1;

});

return frequency;

}

console.log(wordFrequency("Hello world hello")); // Output: { hello: 2, world: 1 }

let integer = 42; // Integer

let float = 3.14; // Floating-point number

let scientific = 1.2e3; // Scientific notation (1.2 \* 10^3)

**Examples:**

console.log(5 / 0); // Output: Infinity

console.log(-5 / 0); // Output: -Infinity

console.log("abc" \* 2); // Output: NaN

**Examples:**

let a = 10, b = 3;

console.log(a + b); // Output: 13

console.log(a % b); // Output: 1

console.log(a \*\* b); // Output: 1000

**Examples:**

let num = 3.14159;

console.log(num.toFixed(2)); // Output: "3.14"

console.log((255).toString(16)); // Output: "ff"

**Examples:**

console.log(Math.PI); // Output: 3.141592653589793

console.log(Math.max(1, 10, -5)); // Output: 10

console.log(Math.sqrt(16)); // Output: 4

console.log(Math.random()); // Output: A random value between 0 and 1

function getRandomInt(min, max) {

return Math.floor(Math.random() \* (max - min + 1)) + min;

}

console.log(getRandomInt(1, 10)); // Output: Random integer between 1 and 10

**Example:**

console.log(0.1 + 0.2); // Output: 0.30000000000000004

let result = (0.1 + 0.2).toFixed(2);

console.log(result); // Output: "0.30"

**Example 1: Calculating Area of a Circle**

function calculateCircleArea(radius) {

return Math.PI \* Math.pow(radius, 2);

}

console.log(calculateCircleArea(5)); // Output: 78.53981633974483

**Example 2: Finding the Larger of Two Numbers**

function findLarger(a, b) {

return Math.max(a, b);

}

console.log(findLarger(10, 20)); // Output: 20

**Example:**

if (!0) {

console.log("0 is falsy"); // Output: 0 is falsy

}

if (!"") {

console.log("Empty string is falsy"); // Output: Empty string is falsy

}

**Example:**

if ("hello") {

console.log("Non-empty strings are truthy"); // Output: Non-empty strings are truthy

}

if (42) {

console.log("Non-zero numbers are truthy"); // Output: Non-zero numbers are truthy

}

**Example 1: Using if statements**

let value = ""; // Empty string

if (value) {

console.log("This is truthy");

} else {

console.log("This is falsy"); // Output: This is falsy

}

**Example 2: Combining Logical Operators**

let a = "hello" && 42; // Both are truthy, so result is 42

console.log(a); // Output: 42

let b = 0 || "fallback"; // 0 is falsy, so result is "fallback"

console.log(b); // Output: fallback

**Example: Boolean Coercion**

let value1 = Boolean(0); // Explicit coercion to boolean

console.log(value1); // Output: false

let value2 = !!"text"; // Double negation to check truthiness

console.log(value2); // Output: true

**Empty arrays and objects are truthy**:

if ([]) {

console.log("Empty array is truthy"); // Output: Empty array is truthy

}

if ({}) {

console.log("Empty object is truthy"); // Output: Empty object is truthy

}

**Beware of NaN and undefined**:

let notANumber = NaN;

if (!notANumber) {

console.log("NaN is falsy"); // Output: NaN is falsy

}

**Misinterpreting 0 and empty strings**:

let num = 0;

if (num) {

console.log("Truthy");

} else {

console.log("Falsy"); // Output: Falsy

}

**Practical Use Cases**

**Case 1: Default Values**

let username = "";

let displayName = username || "Guest";

console.log(displayName); // Output: Guest

**Case 2: Validating Inputs**

function validateInput(input) {

if (!input) {

console.log("Invalid input"); // Falsy values trigger this

} else {

console.log("Valid input");

}

}

validateInput(""); // Output: Invalid input

validateInput("User input"); // Output: Valid input

**Examples of undefined:**

// Example 1: Uninitialized variable

let name;

console.log(name); // Output: undefined

// Example 2: Missing function argument

function greet(message) {

console.log(message); // Output: undefined if no argument is passed

}

greet();

// Example 3: Missing object property

let user = { age: 25 };

console.log(user.name); // Output: undefined

**Examples of null:**

// Example 1: Explicitly assigning null

let user = null;

console.log(user); // Output: null

// Example 2: Resetting a variable

let result = "Success";

result = null; // Indicates that the value is intentionally cleared

console.log(result); // Output: null

**Example Comparing undefined and null:**

let a; // Declared but not initialized

let b = null; // Explicitly assigned no value

console.log(a); // Output: undefined

console.log(b); // Output: null

console.log(typeof a); // Output: undefined

console.log(typeof b); // Output: object

**Checking for undefined or null:**

// Using strict equality to differentiate between undefined and null

let value = null;

if (value === undefined) {

console.log("Value is undefined");

} else if (value === null) {

console.log("Value is null");

} else {

console.log("Value has a value");

}

let username = null;

let defaultName = "Guest";

console.log(username ?? defaultName); // Output: "Guest"

**Example with API Data:**

let apiResponse = {

data: null, // No data available

error: undefined, // Error state is not defined

};

if (apiResponse.data === null) {

console.log("No data available");

}

if (apiResponse.error === undefined) {

console.log("Error state not initialized");

}

**Syntax:**

typeof operand;

**Example of typeOf:**

console.log(typeof "Hello"); // Output: "string"

console.log(typeof 42); // Output: "number"

console.log(typeof true); // Output: "boolean"

console.log(typeof undefined); // Output: "undefined"

console.log(typeof { key: "value" }); // Output: "object"

console.log(typeof null); // Output: "object" (a known quirk in JavaScript)

console.log(typeof function () {}); // Output: "function"

console.log(typeof Symbol("id")); // Output: "symbol"

console.log(typeof 123n); // Output: "bigint"

**Syntax:**

object instanceof Constructor;

**Examples of instanceof:**

// Example with built-in objects

let arr = [1, 2, 3];

console.log(arr instanceof Array); // Output: true

console.log(arr instanceof Object); // Output: true (Array is a subtype of Object)

// Example with custom objects

function Person(name) {

this.name = name;

}

let john = new Person("John");

console.log(john instanceof Person); // Output: true

console.log(john instanceof Object); // Output: true

// Example with classes

class Car {}

let myCar = new Car();

console.log(myCar instanceof Car); // Output: true

console.log(myCar instanceof Object); // Output: true

**Practical Use Cases**

**Use Case: Validating Input Types**

When building a function that accepts various data types, you can use typeof to validate inputs.

function processInput(input) {

if (typeof input === "string") {

console.log("Processing string:", input);

} else if (typeof input === "number") {

console.log("Processing number:", input);

} else {

console.log("Unsupported data type.");

}

}

processInput(42); // Output: "Processing number: 42"

processInput("Hello"); // Output: "Processing string: Hello"

processInput([1, 2, 3]); // Output: "Unsupported data type."

**Use Case: Verifying Object Types**

When working with objects, you can use instanceof to confirm their type.

function checkInstance(obj) {

if (obj instanceof Array) {

console.log("It's an array!");

} else if (obj instanceof Object) {

console.log("It's a generic object.");

} else {

console.log("Unknown type.");

}

}

checkInstance([1, 2, 3]); // Output: "It's an array!"

checkInstance({ key: "value" }); // Output: "It's a generic object."

checkInstance(42); // Output: "Unknown type."