



# LEARN JAVASCRIPT

🎯 Sharpen Your JavaScript Skills:  
Dive into Variables and Data Types  
with These 10 Challenges! 🎯  
*Coding Exercises and explanations!*

- Exercise: Declare two variables, num1 and num2, and assign them numeric values. Add these two variables and store the result in a third variable named sum. 2
- Exercise: Create a variable greeting and assign it a string value. Then create another variable name and assign it a string. Concatenate these two variables and print the result. 2
- Exercise: Declare a variable isValid and assign it a boolean value. Check if isValid is true, and if so, print "Valid". Otherwise, print "Invalid". 3
- Exercise: Write a JavaScript program to swap the values of two variables a and b. 3
- Exercise: Create a variable undefinedVar without assigning any value to it. Print its value and data type. 4
- Exercise: Declare a constant PI and assign it the value of 3.14. Attempt to reassign it to a different value and observe the result. 4
- Exercise: Create a variable nullVar and explicitly set its value to null. Print its type. 5
- Exercise: Convert the string "123" into a number without using any built-in JavaScript 5
- Exercise: Create an array with four different data types (string, number, boolean, and null), and print the array and its length. 5
- Exercise: Declare a variable count with a number value. Use the typeof operator to check its type, change the value to a string, and check its type again. 6

**Exercise:** Declare two variables, num1 and num2, and assign them numeric values. Add these two variables and store the result in a third variable named sum.

Solution:

```
let num1 = 10;
```

```
let num2 = 20;
```

```
let sum = num1 + num2;
```

Explanation: This exercise demonstrates basic variable declaration and arithmetic operations in JavaScript. The let keyword is used to declare variables that can be reassigned later.

**Exercise:** Create a variable greeting and assign it a string value. Then create another variable name and assign it a string. Concatenate these two variables and print the result.

Solution:

```
let greeting = "Hello, ";
```

```
let name = "Alice";
```

```
console.log(greeting + name);
```

Explanation: This exercise shows string concatenation. The + operator is used to concatenate two string variables.

Exercise: Declare a variable `isValid` and assign it a boolean value. Check if `isValid` is true, and if so, print "Valid". Otherwise, print "Invalid".

Solution:

```
let isValid = true;
if (isValid) {
  console.log("Valid");
} else {
  console.log("Invalid");
}
```

Explanation: This exercise introduces boolean data types and basic conditional statements. The if-else structure is used to execute code based on the boolean value of `isValid`.

Exercise: Write a JavaScript program to swap the values of two variables `a` and `b`.

Solution:

```
let a = 5, b = 10;
[a, b] = [b, a];
console.log(a, b); // 10, 5
```

Explanation: This exercise involves swapping variable values using array destructuring, an elegant way to exchange values without needing a temporary variable.

**Exercise: Create a variable `undefinedVar` without assigning any value to it. Print its value and data type.**

Solution:

```
let undefinedVar;  
console.log(undefinedVar); // undefined  
console.log(typeof undefinedVar); // "undefined"
```

Explanation: A variable declared without a value automatically gets the value `undefined`. The `typeof` operator is used to print the data type of the variable.

**Exercise: Declare a constant `PI` and assign it the value of `3.14`. Attempt to reassign it to a different value and observe the result.**

Solution:

```
const PI = 3.14;  
PI = 3.15; // TypeError
```

Explanation: This exercise shows that constants declared with `const` cannot be reassigned. Attempting to do so results in a `TypeError`.

**Exercise: Create a variable `nullVar` and explicitly set its value to `null`. Print its type.**

Solution:

```
let nullVar = null;  
console.log(typeof nullVar); // "object"
```

Explanation: In JavaScript, `null` is an object. It represents the intentional absence of any object value.

**Exercise: Convert the string `"123"` into a number without using any built-in JavaScript**

functions like `parseInt`.

Solution:

```
let str = "123";  
let num = +str;  
console.log(num); // 123
```

Explanation: The unary plus (+) operator is a quick way to convert strings to numbers in JavaScript. It's a shorthand method often used for type conversion.

**Exercise: Create an array with four different data types (string, number, boolean, and `null`), and print the array and its length.**

Solution:

```
let mixedArray = ["Hello", 42, true, null];  
console.log(mixedArray); // ["Hello", 42, true, null]  
console.log(mixedArray.length); // 4
```

Explanation: This exercise demonstrates an array that holds multiple data types. In JavaScript, arrays can contain elements of different types. The length property of an array returns the number of elements it contains.

**Exercise: Declare a variable count with a number value. Use the typeof operator to check its type, change the value to a string, and check its type again.**

Solution:

```
let count = 10;  
console.log(typeof count); // "number"  
count = count.toString();  
console.log(typeof count); // "string"
```

Explanation: Initially, count is a number. The typeof operator is used to verify its data type. The toString() method is then used to convert the number to a string, and typeof confirms the type change.