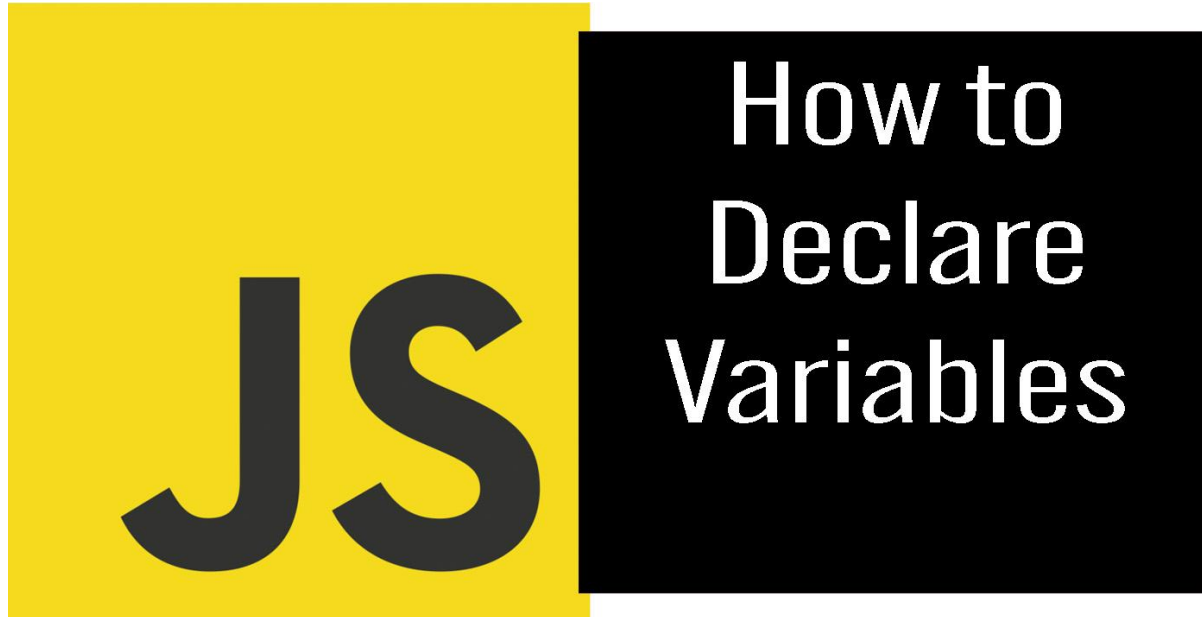


Declaring JavaScript Variables Declaring Variables in JavaScript



1. Declaration Keywords:	2
2. Variable Naming Rules:	2
3. Declaration and Initialization:	3
4. Reassignment:	3
5. Constants:	3
6. Scope:	3
7. Hoisting:	4
8. Naming Conventions:	5
10 coding exercises related to declaring variables in JavaScript	5
Exercise 1: Declare and Initialize Variables	5
Exercise 2: Variable Reassignment	6
Exercise 3: Constants	6
Exercise 4: Variable Scoping	7

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

Exercise 5: Variable Hoisting	8
Exercise 6: Template Literals	8
Exercise 7: Destructuring Objects	9
Exercise 8: Dynamic Typing	10
Exercise 9: Variable Naming	10
Exercise 10: Global Variables	11

Declaring variables is one of the fundamental concepts in JavaScript. Variables are used to store and manage data in your programs. To declare a variable means to create a named placeholder for data that you want to use later. Here's a detailed explanation of declaring variables in JavaScript, along with coding examples:

1. Declaration Keywords:

JavaScript provides three keywords to declare variables:

- `var` (historically used, but not recommended for modern JavaScript).
- `let` (introduced in ES6, recommended for block-scoped variables).
- `const` (introduced in ES6, recommended for block-scoped constants).

2. Variable Naming Rules:

- Variable names must begin with a letter, underscore (`_`), or dollar sign (`$`).
- They can contain letters, numbers, underscores, and dollar signs.
- Variable names are case-sensitive.

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

3. Declaration and Initialization:

To declare and initialize a variable, use the `var`, `let`, or `const` keyword followed by the variable name and an optional initial value.

```
var firstName = 'John'; // Using 'var'  
let age = 30;           // Using 'let'  
const pi = 3.14;       // Using 'const'
```

4. Reassignment:

Variables declared with `var` and `let` can be reassigned with new values.

```
let count = 10;  
count = 20; // Reassigning the variable 'count'
```

5. Constants:

Variables declared with `const` cannot be reassigned after their initial assignment.

```
const pi = 3.14;  
pi = 3.14159; // This will result in an error
```

6. Scope:

Variables have different scopes based on how and where they are declared:

- **Global Scope:** Variables declared outside of any function or block are accessible throughout the program.

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

- Function Scope: Variables declared inside a function are only accessible within that function.
- Block Scope: Variables declared with let and const are block-scoped, meaning they are only accessible within the block they are declared in.

```
var globalVar = 'I am global'; // Global scope
```

```
function myFunction() {  
  var localVar = 'I am local'; // Function scope  
}
```

```
if (true) {  
  let blockVar = 'I am block-scoped'; // Block scope  
}
```

7. Hoisting:

JavaScript variables are hoisted, which means they are moved to the top of their containing scope during compilation.

```
console.log(hoistedVar); // undefined  
var hoistedVar = 5;
```

8. Naming Conventions:

It's a good practice to use meaningful variable names to make your code more readable and maintainable.

```
var age = 30;  
var firstName = 'John';
```

In summary, declaring variables in JavaScript is a crucial step in writing code. Understanding the declaration keywords (var, let, const), naming rules, scoping, and the concept of hoisting will help you effectively work with variables in your JavaScript programs.

10 coding exercises related to declaring variables in JavaScript

Exercise 1: Declare and Initialize Variables

Declare two variables, num1 and num2, and initialize them with numbers. Then, calculate and log their sum.

```
// Step 1: Declare and Initialize Variables  
let num1 = 5;  
let num2 = 7;  
  
// Step 2: Calculate Sum  
let sum = num1 + num2;
```

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

```
// Step 3: Log the Result
console.log(`The sum of ${num1} and ${num2} is
${sum}`);
```

Exercise 2: Variable Reassignment

Declare a variable count and initialize it with a number. Then, reassign it to a different value and log the result.

```
// Step 1: Declare and Initialize Variable
let count = 10;

// Step 2: Reassign Variable
count = 20;

// Step 3: Log the Result
console.log(`The new value of count is ${count}`);
```

Exercise 3: Constants

Declare a constant variable PI and assign it the value of Pi (3.14). Attempt to reassign it and handle the error.

```
// Step 1: Declare a Constant
const PI = 3.14;
```

```
// Step 2: Attempt to Reassign (Expect Error)
try {
  PI = 3.14159; // Error: Cannot reassign a const
variable
} catch (error) {
  console.log(`Error: ${error.message}`);
}
```

Exercise 4: Variable Scoping

Declare a variable x inside a function and log its value both inside and outside the function.

```
// Step 1: Declare Variable Inside a Function
function myFunction() {
  let x = 5;
  console.log(`Inside function: x = ${x}`);
}
```

```
// Step 2: Call the Function
myFunction();
```

```
// Step 3: Log Variable Outside the Function (Error
Expected)
```

```
console.log(`Outside function: x = ${x}`);
```

Exercise 5: Variable Hoisting

Declare a variable `hoistedVar` after attempting to log it before declaration.

Observe the result.

```
// Step 1: Attempt to Log Variable Before Declaration
```

```
console.log(hoistedVar); // undefined
```

```
// Step 2: Declare the Variable
```

```
var hoistedVar = 5;
```

```
// Step 3: Log Variable After Declaration
```

```
console.log(hoistedVar); // 5
```

Exercise 6: Template Literals

Create a template literal to generate a personalized greeting message.

```
// Step 1: Declare Variables
```

```
const name = 'Alice';
```

```
const age = 30;
```

```
// Step 2: Generate Greeting Message
```



```
const greeting = `Hello, ${name}! You are ${age} years old.`;
```

```
// Step 3: Log the Greeting  
console.log(greeting);
```

Exercise 7: Destructuring Objects

Declare an object person with firstName and lastName properties. Use object destructuring to extract and log these properties.

```
// Step 1: Declare an Object  
const person = { firstName: 'John', lastName: 'Doe' };
```

```
// Step 2: Destructure the Object  
const { firstName, lastName } = person;
```

```
// Step 3: Log the Extracted Properties  
console.log(`First Name: ${firstName}`);  
console.log(`Last Name: ${lastName}`);
```

Exercise 8: Dynamic Typing

Declare a variable value and initialize it with a number. Later, reassign it with a string and log its type.

```
// Step 1: Declare and Initialize Variable
let value = 42;

// Step 2: Reassign Variable with a String
value = 'Hello';

// Step 3: Log the Type
console.log(`The type of value is ${typeof value}`);
```

Exercise 9: Variable Naming

Declare a variable with an invalid name and observe the error message.

```
// Step 1: Declare a Variable with an Invalid Name
let 123abc = 'Invalid';

// Step 2: Expect Syntax Error
```

Exercise 10: Global Variables

Declare a global variable `total` and write a function that modifies it. Log the variable before and after the function call.

```
// Step 1: Declare a Global Variable
let total = 0;

// Step 2: Function to Modify the Variable
function addToTotal(value) {
  total += value;
}

// Step 3: Log Variable Before Function Call
console.log(`Total before: ${total}`);

// Step 4: Call the Function
addToTotal(10);

// Step 5: Log Variable After Function Call
console.log(`Total after: ${total}`);
```

These exercises cover various aspects of declaring variables in JavaScript, including declaration, initialization, reassignment, scoping, constants, hoisting, template literals, destructuring, dynamic typing, and variable naming conventions.

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>