



Introduction



1. Write a JS program that demonstrate all examples of var, let and const variable declaration.

➤ Var:

```
var x = 10;  
console.log(x);
```

Ans: 10

```
var x = 10;  
var x = 20;  
console.log(x);
```

Ans: 20

```
var y;  
console.log(y);
```

Ans: Undefined

```
var z;  
var z = 10;  
console.log(z);
```

Ans: 10

```
var x = 50;  
{  
  var x = 500;  
}  
console.log(x);
```

Ans: 500

➤ Let:

```
let x = 10;  
let x = 10;  
console.log(x);
```

Ans = error

```
let z = 10;  
  
{  
  let z = 20;  
  console.log(z);  
}  
console.log(z);
```

Ans = 20
Ans = 10

```
let d;  
d = 10;  
d = d + 20;  
console.log(d);
```

Ans = 30

➤ **Const:**

```
const x;  
x = 10;  
console.log(x);
```

Ans = error

```
const x;  
x = 10;  
x = x + 20;  
console.log(x)
```

Ans = error

```
const x = 10;  
{  
  const x = 20;  
}  
console.log(x);
```

Ans = 10

2. Write a JS program that demonstrate all examples of different operators.

➤ Arithmetic Operators:

```
let x = 10, y = 20;  
let z = x + y;  
console.log(z);
```

Ans: 30;

```
let x = 20, y = 10;  
let z = x - y;  
console.log(z);
```

Ans: 10;

```
let x = 20, y = 10;  
let z = x * y;  
console.log(z);
```

Ans: 200;

```
let x = 20, y = 10;  
let z = x / y;  
console.log(z);
```

Ans: 2;

```
let x = 20, y = 10;  
let z = x % y;  
console.log(z);
```

Ans: 0;

```
let x = 2, y = 3;  
let z = x ** y;  
console.log(z);
```

Ans: $X^Y = 2^3 = 8$;

```
let x = 2;  
let z = ++x;  
console.log(z);
```

Ans: 3;

```
let x = 2;  
let z = --x;  
console.log(z);
```

Ans: 1;

➤ Assignment Operators:

```
let x = 10;  
console.log(x);
```

Ans: 10;

```
let x = 10;
```

x += 3; console.log(x); let x = 10; x -= 3; console.log(x);	Ans: $x = x + 3 = 10 + 3 = 13$; Ans: $x = x - 3 = 10 - 3 = 7$;
let x = 10; x *= 3; console.log(x);	Ans: $x = x * 3 = 10 * 3 = 30$;
let x = 10; x /= 2; console.log(x);	Ans: $x = x / 2 = 10 / 2 = 5$;
let x = 10; x %= 2; console.log(x);	Ans: $x = x \% 2 = 10 \% 2 = 0$;
let x = 2, y = 3; x **= y; console.log(x);	Ans: $x = x ** y = x^y = 2^3 = 8$;

➤ Assignment Operators:

The **+** operators can also be used to add (Concatenate) string.

➤ Comparison Operators:

let x = 10, y = '10'; console.log (x == y);	Ans: true;
let x = 10, y = '10'; console.log (x === y);	Ans: false;
let x = 10, y = '10'; console.log (x != y);	Ans: false;
let x = 10, y = '10'; console.log (x !== y);	Ans: true;
let x = 10, y = 8; console.log (x > y);	Ans: true;
let x = 10, y = 8; console.log (x < y);	Ans: true;

```
let x = 10, y = 10;  
console.log (x >= y);
```

Ans: true;

```
let x = 10, y = 8;  
console.log (x <= y);
```

Ans: false;

```
let x = 10, y = 20, z = 30;  
(z == x + y) ? (console.log("Yes")) : (console.log("No"));
```

Ans: Yes;

➤ Logical Operators:

```
let x = 10, y = 20, z = '20';  
if ((x < y) && (y === z) {  
    console.log ('Yes');  
} else {  
    console.log ('No');  
}
```

Ans: No;

```
let x = 10, y = 20, z = '20';  
if ((x < y) || (y === z) {  
    console.log ('Yes');  
} else {  
    console.log ('No');  
}
```

Ans: Yes;

```
let x = 10, y = 20, z = '20';  
if (! (y === z) {  
    console.log ('Yes');  
} else {  
    console.log ('No');  
}
```

Ans: Yes;

```
let x = 10;  
console.log(typeof x);
```

Ans: number;

```
let x = 'Hello';  
console.log(typeof x);
```

Ans: string;

```
let x;  
console.log(typeof x);
```

Ans: undefined;

```
let x = null;  
console.log(typeof x);
```

Ans: object;

Interview Questions

1. What is JavaScript?

- JavaScript is **programming language** also called a **scripting language**. A Scripting language is type of programming language.
- It is **Object based, Lightweight, Cross platformed translated language**.
- JavaScript was developed by **Brendam Eich**, who was a **Netscape programmer**.
- This is run by JavaScript engine, that is already available in web browser.

2. What is Advantage and Disadvantage of JavaScript?

➤ Advantages:

- **Interactivity:** JavaScript enables you to add interactive elements to your website, such as drop-down menus, pop-up windows, and other features that make your site more engaging and user-friendly.
- **Client-side scripting:** JavaScript runs on the client-side, which means that it is processed by the user's browser rather than the web server, reducing the load on the server and improving website performance.
- **Easy to learn:** JavaScript is a relatively easy programming language to learn, especially for beginners who have some experience with HTML and CSS.
- **Versatility:** JavaScript can be used for a wide range of applications, including web development, mobile app development, desktop app development, and game development.

➤ Disadvantages:

- **Security risks:** JavaScript can be vulnerable to security threats, such as cross-site scripting (XSS) attacks and injection attacks, which can compromise the security of your website and user data.

- **Browser compatibility:** Not all browsers support the latest version of JavaScript, and some users may have disabled JavaScript on their browsers, which can affect the functionality of your website.
- **Performance issues:** JavaScript can slow down your website if it's not optimized properly, especially if you use complex scripts or large libraries.
- **Debugging can be difficult:** Debugging JavaScript code can be challenging, especially for complex applications, due to the dynamic nature of the language.

3. What is the purpose of the let keyword?

- The let keyword in JavaScript is used to declare block-scoped variables. Block-scoped variables are variables that are only **accessible within the block of code where they are declared**. And we cannot declare **same variable let for multiple time**.

4. Give the difference between var, let and const.

- **Scope:** **var** has function scope, meaning that it is accessible throughout the function in which it is declared, including any nested functions. **let** and **const** have block scope, meaning that they are only accessible within the block of code where they are declared, including any nested blocks.
- **Hoisting:** **var** variables are hoisted to the top of their function scope or global scope, which means that they can be accessed before they are declared. However, their value will be undefined until they are assigned a value. **let** and **const** variables are also hoisted, but they are not accessible before they are declared, which can help prevent bugs.
- **Mutability:** **var** and **let** variables can be reassigned a new value after they are declared. **const** variables are constants and cannot be reassigned a new value after they are declared. However, the object or array that they refer to can still be mutated.

5. What is the difference between =, == and === operator?

- The **=** operator is the assignment operator, which is used to **assign a value** to a variable or property.
- The **==** operator is the equality operator, which **compares the values of two operands**.

- The **===** operator is the strict equality operator, which also **compares the values and type of two operands**.

6. What is identifier? Give the rules to declare identifier.

An identifier is a **name** that is used to identify a **variable, function, or any other user-defined item** in a program. It is a sequence of characters that begins with a letter or underscore and can be followed by any **combination of letters, digits, or underscores**. However, there are some rules that must be followed while declaring an identifier in most programming languages.

Here are some general rules for declare identifier:

- The first character must be a letter or an underscore (_).
- The identifier must consist of only letters, digits, and/or underscores.
- Identifiers are case-sensitive, meaning "foo" and "FOO" are different identifiers.
- Identifiers must not be a reserved keyword or a predefined name in the programming language.
- There is typically a limit on the length of an identifier.

7. List features of JavaScript.

- **Dynamically typed:** JavaScript is a dynamically typed language, which means that the data type of a variable can be changed at runtime.
- **Object-oriented:** JavaScript supports object-oriented programming (OOP) concepts such as encapsulation, inheritance, and polymorphism.
- **Event-driven:** JavaScript is an event-driven language, which means that it can respond to user actions and other events that occur in the browser.
- **Client-side scripting:** JavaScript is primarily used for client-side scripting in web browsers, allowing developers to create interactive user interfaces and dynamic web pages.
- **Cross-platform:** JavaScript can run on a variety of platforms, including web browsers, servers, desktops, and mobile devices.
- **Asynchronous programming:** JavaScript supports asynchronous programming, allowing developers to write non-blocking code that can handle multiple tasks simultaneously.
- **Functional programming:** JavaScript also supports functional programming, allowing developers to write code that is easier to reason about, test, and maintain.

- **Libraries and frameworks:** There are many popular libraries and frameworks built on top of JavaScript, such as React, Angular, and Vue, which can greatly simplify the process of building complex web applications.

8. What is the difference between null and undefined?

- **undefined** is a value that is automatically assigned to a variable that has been declared but not initialized with a value. It also represents the return value of a function that does not explicitly return anything.
- **null** is a value that is explicitly assigned to a variable to indicate that it has no value. It is often used as a placeholder value to initialize a variable that will be assigned a value later.

9. What is the difference between window and document?

➤ Window:

- The window object is the top-level object in the browser's hierarchy of objects, and it represents the current browser window or tab. It provides access to various properties and methods related to the browser window, such as **window.location** for getting or setting the current URL, **window.alert()** for displaying a message box, **window.setTimeout()** for scheduling a function to run after a certain amount of time, and so on.

➤ Document

- The document object is a property of the window object, and it represents the current HTML document that is displayed in the browser window. It provides access to various properties and methods related to the document, such as **document.getElementById()** for finding an element by its ID, **document.createElement()** for creating a new HTML element, **document.title** for getting or setting the document title, and so on.

10. What is local variable and global variable?

➤ Local variable:

- A local variable is a variable that is **declared inside a function or a block of code**, and is only **accessible within that function or block**. Local variables have function or block scope, which means they can only be accessed from within the function or block where they are declared.

➤ Local variable:

- A global variable is a variable that is **declared outside of any function or block**, and is **accessible from anywhere in the code**, including inside functions and blocks. Global variables have global scope, which means they can be accessed from anywhere in the code.

11. What is NaN property?

In JavaScript, NaN stands for "**Not a Number**" and is a special value that represents an invalid number. The NaN value is a property of the **global Number object**, and it is typically returned as the result of an arithmetic operation that has failed, such as dividing by zero or trying to calculate the square root of a negative number.

- The NaN value has some unique characteristics that make it different from other values in JavaScript.
- NaN is a value of the **Number type**, but it is **not equal to any other value**, including itself. This means that you cannot use the `==` or `===` operator to compare a value to NaN, and you must use the **special `isNaN ()` function** to check if a value is NaN.
- Any arithmetic operation that involves NaN as an operand will result in NaN. For example, `NaN + 1` is NaN, and `Math.sqrt(-1)` is NaN.
- The `typeof` operator returns **'number'** for NaN, which can be misleading. To check if a value is NaN, you should use the `isNaN ()` function, which **returns true if the value is NaN**, and **false** otherwise.

12. Is JavaScript a case-sensitive language?

- **Yes**, JavaScript is a case-sensitive language, which means that **uppercase and lowercase letters are treated as distinct characters**. For example, `myVariable` and `myvariable` are two **different variable** names in JavaScript, and `console.log` and `console.Log` are two different function calls.

- This means that when you write JavaScript code, you need to be careful to use the correct capitalization for all **variable names, function names, and other identifiers**. Mixing up uppercase and lowercase letters can result in syntax errors, undefined variables, or other unexpected behavior in your code.

13. What is ECMAScript?

- The full form of ECMA is **European computer manufacture association**.
- ECMA Script is JavaScript standard intended is a JavaScript the interoperability of web pages across different browsers.

14. What are the benefits of initializing variables?