JavaScript Output

JavaScript can "display" data in different ways:

- Writing into an HTML element, using `innerHTML`.
- Writing into the HTML output using `document.write()`.
- Writing into an alert box, using `window.alert()`.
- Writing into the browser console, using `console.log()`.

JavaScript Variables

Variables are Containers for Storing Data

JavaScript Variables can be declared in 4 ways:

- Automatically
- Using `var`
- Using `let`
- Using `const`

Declaration: Variables in JavaScript are declared using the `var`, `let`, or `const` keywords.

- `var`: Historically used for variable declaration. It has function scope.
- `let`: Introduced in ES6, used for block-scoped variables. It allows reassignment of values.
- `const`: Also introduced in ES6, used for variables whose values will not change. It is also block-scoped.

```
x= 5;
y = 6

y = 7;
```

From the examples you can guess:

- x stores the value 5
- y stores the value 6
- z stores the value 11

Note

It is considered good programming practice to always declare variables before use.

Example using var

```
var x                                                    = 5;
var y                                                    = 6;
var z = x + y;
```

Example using let

```
let x                                                    = 5;
let y                                                    = 6;
let z = x + y;
```

Example using const

```
const x                                                  = 5;
const y                                                  = 6;
const z = x + y;
```

Note

The `var` keyword was used in all JavaScript code from 1995 to 2015.

The `let` and `const` keywords were added to JavaScript in 2015.

The `var` keyword should only be used in code written for older browsers.

Mixed Example

```
const price1                                             = 5;
const price2                                             = 6;
let total = price1 + price2;
```

The two variables `price1` and `price2` are declared with the `const` keyword.

These are constant values and cannot be changed.

The variable `total` is declared with the `let` keyword.

The value `total` can be changed.

When to Use var, let, or const?

1. Always declare variables
2. Always use `const` if the value should not be changed
3. Always use `const` if the type should not be changed (Arrays and Objects)
4. Only use `let` if you can't use `const`
5. Only use `var` if you MUST support old browsers.

JavaScript Identifiers

All JavaScript **variables** must be **identified** with **unique names**.

These unique names are called **identifiers**.

Identifiers can be short names (like x and y) or more descriptive names (age, sum, totalVolume).

The general rules for constructing names for variables (unique identifiers) are:

- Names can contain letters, digits, underscores, and dollar signs.
- Names must begin with a letter.
- Names can also begin with $ and _ (but we will not use it in this tutorial).
- Names are case sensitive (y and Y are different variables).
- Reserved words (like JavaScript keywords) cannot be used as names.

Note

JavaScript identifiers are case-sensitive.

Valid Identifiers:

Example

```
var myVariable = 10;
var userName = "John";
var _userAge = 25;
var $test = "Hello";
var myFunction = function() {};
```

Invalid Identifiers:

```
var 123invalid; // Invalid: Cannot start with a digit
var my-variable; // Invalid: Hyphens are not allowed
var let; // Invalid: Reserved word
var function; // Invalid: Reserved word
var break; // Invalid: Reserved word
```

Remembering these rules when naming variables, functions, or objects in JavaScript will help you write clean and error-free code. It's essential to choose descriptive and meaningful names for better readability and maintainability of your code.

Using var:

```
var x = 10;
var y = "Hello";
```

Using let:

```
let name = "John";
let age = 30;
```

Using const:

```
const PI = 3.14;
const daysOfWeek = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday'];
```

Multiple Variable Declaration:

```
var a = 5, b = 10, c = 15;
let firstName = "Alice", lastName = "Smith";
```

Destructuring Assignment (with Arrays and Objects):

```
// Array destructuring
let [first, second] = ['apple', 'banana'];
// Object destructuring
let { name, age } = { name: 'John', age: 30 };
```

Dynamic Variable Names:

```
let dynamicVarName = 'age';
let person = {};
person[dynamicVarName] = 25;
```

These examples demonstrate various ways to declare variables in JavaScript along with their initial values. Depending on your use case and the nature of the data you're working with, you can choose the most appropriate method for declaring and initializing your variables.

Task 1]

1. Explain the difference between console.log(), alert(), and document.write() in JavaScript regarding outputting information.
2. When should you use var, let, or const for declaring variables in JavaScript? Provide examples illustrating their respective use cases.
3. What are the characteristics of identifiers in JavaScript? Discuss the rules and conventions for naming identifiers in JavaScript.
4. Discuss the importance of variable naming conventions in JavaScript. Provide examples of meaningful and appropriate variable names.
5. Describe the concept of variable reassignment in JavaScript. How does it relate to the usage of const?
6. Discuss the implications of using var, let, or const in terms of memory management and performance in JavaScript applications.