জাভাস্ক্রিপ্টে **ভেরিয়েবল** হলো ডেটা সংরক্ষণের জন্য ব্যবহৃত কন্টেইনার। ভেরিয়েবল ব্যবহার করে আপনি বিভিন্ন ধরনের ডেটা (যেমন সংখ্যা, স্ট্রিং, অবজেক্ট) সংরক্ষণ করতে পারেন এবং পরে সেই ডেটা অ্যাক্সেস বা পরিবর্তন করতে পারেন। জাভাস্ক্রিপ্টে ভেরিয়েবল ডিক্লেয়ার করতে let, const, বা var ব্যবহার করা হয়। নিচে ভেরিয়েবল সম্পর্কে বিস্তারিত ব্যাখ্যা দেওয়া হলো:

**১. ভেরিয়েবল ডিক্লেয়ারেশন:**

জাভাস্ক্রিপ্টে ভেরিয়েবল ডিক্লেয়ার করতে নিম্নলিখিত কিওয়ার্ডগুলি ব্যবহার করা হয়:

**let:**

* let ব্যবহার করে আপনি একটি **পরিবর্তনযোগ্য (mutable)** ভেরিয়েবল ডিক্লেয়ার করতে পারেন।
* এটি ব্লক-স্কোপড, অর্থাৎ এটি শুধুমাত্র যে ব্লকে ডিক্লেয়ার করা হয়েছে সেই ব্লকের মধ্যে অ্যাক্সেস করা যায়।

**সিনট্যাক্স:**

javascript

Copy

let variableName = value;

**উদাহরণ:**

javascript

Copy

let name = "John";

name = "Alice"; // ভেরিয়েবলের মান পরিবর্তন করা যায়

console.log(name); // আউটপুট: Alice

**const:**

* const ব্যবহার করে আপনি একটি **অপরিবর্তনীয় (immutable)** ভেরিয়েবল ডিক্লেয়ার করতে পারেন।
* এটি ব্লক-স্কোপড এবং একবার মান অ্যাসাইন করার পর তা পরিবর্তন করা যায় না।

**সিনট্যাক্স:**

javascript

Copy

const constantName = value;

**উদাহরণ:**

javascript

Copy

const age = 25;

// age = 30; // Error: ভেরিয়েবলের মান পরিবর্তন করা যায় না

console.log(age); // আউটপুট: 25

**var:**

* var হলো জাভাস্ক্রিপ্টের পুরানো সিনট্যাক্স। এটি ফাংশন-স্কোপড বা গ্লোবাল-স্কোপড।
* এটি এখন কম ব্যবহৃত হয় কারণ এটি হোইস্টিং এবং স্কোপিং ইস্যু তৈরি করতে পারে।

**সিনট্যাক্স:**

javascript

Copy

var variableName = value;

**উদাহরণ:**

javascript

Copy

var city = "Dhaka";

city = "Chittagong"; // ভেরিয়েবলের মান পরিবর্তন করা যায়

console.log(city); // আউটপুট: Chittagong

**২. ভেরিয়েবলের নামকরণ নিয়ম:**

* ভেরিয়েবলের নাম **কেস-সেনসিটিভ** (case-sensitive), অর্থাৎ name এবং Name আলাদা।
* নামের শুরুতে একটি অক্ষর, আন্ডারস্কোর (\_), বা ডলার সাইন ($) ব্যবহার করতে পারেন।
* নামের মধ্যে সংখ্যা ব্যবহার করা যায়, কিন্তু নামের শুরুতে নয়।
* জাভাস্ক্রিপ্টের রিজার্ভেড কিওয়ার্ড (যেমন let, const, if, for) ভেরিয়েবলের নাম হিসেবে ব্যবহার করা যায় না।

**উদাহরণ:**

javascript

Copy

let firstName = "John"; // বৈধ

let \_lastName = "Doe"; // বৈধ

let $age = 25; // বৈধ

let 1name = "Alice"; // অবৈধ (নামের শুরুতে সংখ্যা ব্যবহার করা যায় না)

**৩. ভেরিয়েবলের স্কোপ:**

* **গ্লোবাল স্কোপ**: ভেরিয়েবলটি পুরো প্রোগ্রামে অ্যাক্সেস করা যায়।
* **লোকাল স্কোপ**: ভেরিয়েবলটি শুধুমাত্র একটি নির্দিষ্ট ব্লক বা ফাংশনের মধ্যে অ্যাক্সেস করা যায়।

**উদাহরণ:**

javascript

Copy

let globalVar = "I am global"; // গ্লোবাল ভেরিয়েবল

function myFunction() {

let localVar = "I am local"; // লোকাল ভেরিয়েবল

console.log(globalVar); // গ্লোবাল ভেরিয়েবল অ্যাক্সেস করা যায়

console.log(localVar); // লোকাল ভেরিয়েবল অ্যাক্সেস করা যায়

}

myFunction();

// console.log(localVar); // Error: লোকাল ভেরিয়েবল বাইরে অ্যাক্সেস করা যায় না

**৪. ভেরিয়েবলের ডেটা টাইপ:**

জাভাস্ক্রিপ্টে ভেরিয়েবলের ডেটা টাইপ ডাইনামিক, অর্থাৎ ভেরিয়েবল যে কোনো ধরনের ডেটা সংরক্ষণ করতে পারে। জাভাস্ক্রিপ্টে প্রধান ডেটা টাইপগুলি হলো:

* **প্রিমিটিভ**: String, Number, Boolean, null, undefined, Symbol
* **নন-প্রিমিটিভ**: Object, Array, Function

**উদাহরণ:**

javascript

Copy

let name = "John"; // String

let age = 25; // Number

let isStudent = true; // Boolean

let person = { name: "John", age: 25 }; // Object

let fruits = ["Apple", "Banana", "Mango"]; // Array

**৫. ভেরিয়েবলের হোইস্টিং:**

জাভাস্ক্রিপ্টে ভেরিয়েবল ডিক্লেয়ারেশন কোডের সর্বোচ্চে হোইস্ট হয়, কিন্তু মান অ্যাসাইনমেন্ট হোইস্ট হয় না। এটি শুধুমাত্র var এর ক্ষেত্রে প্রযোজ্য।

**উদাহরণ:**

javascript

Copy

console.log(x); // আউটপুট: undefined (হোইস্টিং)

var x = 10;

**সারসংক্ষেপ:**

| **বৈশিষ্ট্য** | **let** | **const** | **var** |
| --- | --- | --- | --- |
| **স্কোপ** | ব্লক-স্কোপড | ব্লক-স্কোপড | ফাংশন-স্কোপড বা গ্লোবাল-স্কোপড |
| **পরিবর্তনযোগ্যতা** | পরিবর্তনযোগ্য | অপরিবর্তনীয় | পরিবর্তনযোগ্য |
| **হোইস্টিং** | হোইস্টেড (TDZ) | হোইস্টেড (TDZ) | হোইস্টেড |
| **ইউজ কেস** | পরিবর্তনযোগ্য ভেরিয়েবল | অপরিবর্তনীয় ভেরিয়েবল | পুরানো সিনট্যাক্স |

**কোনটি কখন ব্যবহার করবেন?:**

* যদি ভেরিয়েবলের মান পরিবর্তন করার প্রয়োজন হয়, তাহলে let ব্যবহার করুন।
* যদি ভেরিয়েবলের মান পরিবর্তন করার প্রয়োজন না হয়, তাহলে const ব্যবহার করুন।
* var ব্যবহার করা এড়িয়ে চলুন কারণ এটি স্কোপিং এবং হোইস্টিং ইস্যু তৈরি করতে পারে।

ভেরিয়েবল বুঝলে আপনি জাভাস্ক্রিপ্টে ডেটা সংরক্ষণ এবং ম্যানিপুলেট করতে পারবেন। 😊