জাভাতে **ভেরিয়েবল (Variable)** হল ডেটা স্টোর করার জন্য একটি কন্টেইনার। ভেরিয়েবল ব্যবহার করে প্রোগ্রামে বিভিন্ন ধরনের ডেটা (যেমন: নাম্বার, টেক্সট, বুলিয়ান ইত্যাদি) সংরক্ষণ করা যায় এবং পরবর্তীতে সেই ডেটা ব্যবহার করা যায়। জাভাতে প্রতিটি ভেরিয়েবলের একটি নির্দিষ্ট **ডেটা টাইপ (Data Type)** থাকে, যা নির্দেশ করে ভেরিয়েবলটি কী ধরনের ডেটা স্টোর করতে পারে।

**ভেরিয়েবল ডিক্লারেশনের সিনট্যাক্স:**

জাভাতে ভেরিয়েবল ডিক্লার করার সাধারণ সিনট্যাক্স নিচের মতো:

java

Copy

data\_type variable\_name = value;

* **data\_type**: ভেরিয়েবলের ডেটা টাইপ (যেমন: int, String, double ইত্যাদি)।
* **variable\_name**: ভেরিয়েবলের নাম (যেকোনো ভ্যালিড নাম হতে পারে)।
* **value**: ভেরিয়েবলে সংরক্ষিত ডেটা।

**উদাহরণ:**

java

Copy

int age = 25;

String name = "John";

double salary = 50000.50;

**ভেরিয়েবলের প্রকারভেদ:**

জাভাতে ভেরিয়েবল তিন ধরনের হতে পারে:

1. **লোকাল ভেরিয়েবল (Local Variable)**:
   * এটি মেথড, কনস্ট্রাক্টর বা ব্লকের ভিতরে ডিক্লার করা হয়।
   * এটি শুধুমাত্র সেই ব্লকের ভিতরে ব্যবহার করা যায়।
   * লোকাল ভেরিয়েবল ইনিশিয়ালাইজ (Initialize) করা বাধ্যতামূলক।

java

Copy

public void display() {

int number = 10; // লোকাল ভেরিয়েবল

System.out.println(number);

}

1. **ইনস্ট্যান্স ভেরিয়েবল (Instance Variable)**:
   * এটি ক্লাসের ভিতরে ডিক্লার করা হয়, কিন্তু মেথডের বাইরে।
   * এটি অবজেক্ট লেভেলের ভেরিয়েবল, অর্থাৎ প্রতিটি অবজেক্টের জন্য আলাদা আলাদা ভ্যালু থাকে।
   * ইনস্ট্যান্স ভেরিয়েবল ডিফল্ট ভ্যালু (Default Value) নিয়ে শুরু হয় (যেমন: int এর জন্য 0, String এর জন্য null)।

java

Copy

public class Student {

String name; // ইনস্ট্যান্স ভেরিয়েবল

int age; // ইনস্ট্যান্স ভেরিয়েবল

}

1. **স্ট্যাটিক ভেরিয়েবল (Static Variable)**:
   * এটি static কীওয়ার্ড ব্যবহার করে ডিক্লার করা হয়।
   * এটি ক্লাস লেভেলের ভেরিয়েবল, অর্থাৎ সব অবজেক্টের জন্য একই ভ্যালু শেয়ার করে।
   * স্ট্যাটিক ভেরিয়েবল প্রোগ্রামের শুরুতে মেমোরিতে লোড হয় এবং প্রোগ্রাম শেষ হওয়া পর্যন্ত থাকে।

java

Copy

public class Counter {

static int count = 0; // স্ট্যাটিক ভেরিয়েবল

}

**ভেরিয়েবলের ডেটা টাইপ:**

জাভাতে ভেরিয়েবলের ডেটা টাইপ দুই ধরনের হতে পারে:

1. **প্রিমিটিভ ডেটা টাইপ (Primitive Data Types)**:
   * এগুলি বিল্ট-ইন ডেটা টাইপ, যা সরাসরি ডেটা স্টোর করে।
   * উদাহরণ: int, double, char, boolean ইত্যাদি।

| **ডেটা টাইপ** | **সাইজ** | **ডেসক্রিপশন** |
| --- | --- | --- |
| int | 4 bytes | পূর্ণ সংখ্যা (Integer) |
| double | 8 bytes | দশমিক সংখ্যা (Floating Point) |
| char | 2 bytes | একটি ইউনিকোড ক্যারেক্টার |
| boolean | 1 bit | true বা false ভ্যালু |

1. java
2. Copy
3. int number = 10;
4. double price = 19.99;
5. char grade = 'A';
6. boolean isJavaFun = true;
7. **নন-প্রিমিটিভ ডেটা টাইপ (Non-Primitive Data Types)**:
   * এগুলি রেফারেন্স টাইপ, যা অবজেক্টকে রেফার করে।
   * উদাহরণ: String, Array, Class, Interface ইত্যাদি।

java

Copy

String name = "Java";

int[] numbers = {1, 2, 3, 4, 5};

**ভেরিয়েবলের নামকরণের নিয়ম:**

1. ভেরিয়েবলের নাম লেটার, আন্ডারস্কোর (\_) বা ডলার সাইন ($) দিয়ে শুরু করতে হবে।
2. নামে স্পেস বা স্পেশাল ক্যারেক্টার ব্যবহার করা যাবে না (শুধু \_ এবং $ ছাড়া)।
3. নাম জাভার কীওয়ার্ড (যেমন: int, class, public ইত্যাদি) হতে পারবে না।
4. নাম কেস-সেনসিটিভ (Case-Sensitive), অর্থাৎ age এবং Age আলাদা।

**উদাহরণ:**

java

Copy

int age = 25; // ভ্যালিড

String firstName; // ভ্যালিড

double \_salary; // ভ্যালিড

int $count; // ভ্যালিড

int 1number; // ইনভ্যালিড (নাম্বার দিয়ে শুরু করা যাবে না)

**ভেরিয়েবলের ব্যবহার:**

ভেরিয়েবল ব্যবহার করে প্রোগ্রামে ডেটা স্টোর এবং ম্যানিপুলেট করা যায়। নিচে একটি উদাহরণ দেওয়া হলো:

java

Copy

public class Main {

public static void main(String[] args) {

int a = 10; // লোকাল ভেরিয়েবল

int b = 20; // লোকাল ভেরিয়েবল

int sum = a + b; // ভেরিয়েবল ব্যবহার করে যোগ করা

System.out.println("Sum is: " + sum); // আউটপুট: Sum is: 30

}

}

**ভেরিয়েবলের গুরুত্ব:**

1. ভেরিয়েবল প্রোগ্রামে ডেটা স্টোর এবং ম্যানেজ করতে সাহায্য করে।
2. এটি কোডকে ডায়নামিক এবং রিইউজেবল করে তোলে।
3. ভেরিয়েবল ব্যবহার করে প্রোগ্রামের রিডাবিলিটি (Readability) বাড়ে।

জাভাতে ভেরিয়েবল শেখা খুবই গুরুত্বপূর্ণ, কারণ এটি প্রোগ্রামিং এর বেসিক কনসেপ্ট। নিয়মিত অনুশীলন করলে ভেরিয়েবলের ব্যবহার সহজেই আয়ত্ত করা যায়। 😊