জাভার সিনট্যাক্স (Syntax) হল জাভা প্রোগ্রাম লিখার নিয়ম এবং গঠন। জাভা প্রোগ্রামিং ভাষার সিনট্যাক্স C এবং C++ এর মতো, তবে এটি আরও সহজ এবং সংগঠিত। নিচে জাভার সিনট্যাক্সের মূল অংশগুলো বাংলায় ব্যাখ্যা করা হলো:

**১. ক্লাস ডিক্লারেশন (Class Declaration)**

জাভা একটি অবজেক্ট-ওরিয়েন্টেড ভাষা, তাই প্রতিটি জাভা প্রোগ্রাম একটি ক্লাস দিয়ে শুরু হয়। ক্লাস হল জাভা প্রোগ্রামের বিল্ডিং ব্লক।

java

Copy

public class ClassName {

// ক্লাসের ভিতরে কোড লেখা হয়

}

* public: এটি একটি অ্যাক্সেস মডিফায়ার, যা নির্দেশ করে এই ক্লাসটি যেকোনো জায়গা থেকে অ্যাক্সেস করা যাবে।
* class: এটি একটি কীওয়ার্ড, যা দিয়ে ক্লাস ডিক্লার করা হয়।
* ClassName: ক্লাসের নাম। ক্লাসের নাম ক্যাপিটাল লেটার দিয়ে শুরু করা ভালো প্র্যাকটিস।

**২. মেইন মেথড (Main Method)**

প্রতিটি জাভা প্রোগ্রামে একটি main মেথড থাকতে হবে, কারণ এটি প্রোগ্রামের এন্ট্রি পয়েন্ট (Entry Point)। প্রোগ্রাম রান করলে main মেথড থেকে এক্সিকিউশন শুরু হয়।

java

Copy

public static void main(String[] args) {

// প্রোগ্রামের লজিক এখানে লেখা হয়

}

* public: মেথডটি যেকোনো জায়গা থেকে অ্যাক্সেস করা যাবে।
* static: মেথডটি ক্লাসের অবজেক্ট ছাড়াই কল করা যাবে।
* void: মেথডটি কোনো ভ্যালু রিটার্ন করে না।
* main: মেথডের নাম। এটি জাভা প্রোগ্রামের এন্ট্রি পয়েন্ট।
* String[] args: এটি কমান্ড লাইন আর্গুমেন্ট (Command Line Arguments) গ্রহণ করে।

**৩. ভেরিয়েবল ডিক্লারেশন (Variable Declaration)**

জাভাতে ভেরিয়েবল ডিক্লার করার জন্য ডেটা টাইপ এবং ভেরিয়েবলের নাম উল্লেখ করতে হয়।

java

Copy

int number = 10;

String name = "Java";

* int: ডেটা টাইপ (ইন্টিজার টাইপের ভেরিয়েবল)।
* number: ভেরিয়েবলের নাম।
* 10: ভেরিয়েবলের ভ্যালু।

**৪. কন্ট্রোল স্টেটমেন্ট (Control Statements)**

জাভাতে কন্ট্রোল স্টেটমেন্ট ব্যবহার করে প্রোগ্রামের ফ্লো কন্ট্রোল করা হয়। যেমন: if, else, for, while ইত্যাদি।

**উদাহরণ:**

java

Copy

if (number > 5) {

System.out.println("Number is greater than 5");

} else {

System.out.println("Number is less than or equal to 5");

}

**৫. মেথড ডিক্লারেশন (Method Declaration)**

মেথড হল একটি ব্লক অফ কোড, যা একটি নির্দিষ্ট টাস্ক সম্পাদন করে। মেথড ডিক্লার করার সিনট্যাক্স নিচের মতো:

java

Copy

public int add(int a, int b) {

return a + b;

}

* public: অ্যাক্সেস মডিফায়ার।
* int: রিটার্ন টাইপ (মেথডটি একটি ইন্টিজার ভ্যালু রিটার্ন করবে)।
* add: মেথডের নাম।
* (int a, int b): প্যারামিটার লিস্ট।

**৬. কমেন্টস (Comments)**

জাভাতে কমেন্টস ব্যবহার করে কোডের ব্যাখ্যা লেখা হয়। কমেন্টস প্রোগ্রাম এক্সিকিউশনে কোনো প্রভাব ফেলে না।

java

Copy

// এটি একটি সিঙ্গেল লাইন কমেন্ট

/\*

এটি একটি

মাল্টি-লাইন কমেন্ট

\*/

**৭. প্রিন্ট স্টেটমেন্ট (Print Statement)**

জাভাতে System.out.println() ব্যবহার করে কনসোলে আউটপুট দেখানো হয়।

java

Copy

System.out.println("Hello, World!");

* System.out.println(): এটি কনসোলে টেক্সট প্রিন্ট করে এবং নতুন লাইন যোগ করে।
* System.out.print(): এটি নতুন লাইন যোগ করে না।

**৮. জাভা প্রোগ্রামের উদাহরণ**

একটি সম্পূর্ণ জাভা প্রোগ্রামের উদাহরণ:

java

Copy

public class HelloWorld {

public static void main(String[] args) {

int number = 10;

System.out.println("Number is: " + number);

}

}

* এই প্রোগ্রামটি Number is: 10 আউটপুট দেবে।

**জাভা সিনট্যাক্সের মূল নিয়ম:**

1. জাভা কেস-সেনসিটিভ (Case-Sensitive), অর্থাৎ Hello এবং hello আলাদা।
2. প্রতিটি স্টেটমেন্ট শেষ করতে সেমিকোলন (;) ব্যবহার করতে হয়।
3. ক্লাসের নাম এবং ফাইলের নাম একই হতে হবে (যদি ক্লাস public হয়)।
4. কোড ব্লক { } এর মধ্যে লেখা হয়।

জাভার সিনট্যাক্স সহজ এবং সংগঠিত, তাই নিয়মিত অনুশীলন করলে এটি দ্রুত আয়ত্ত করা যায়। 😊