**Conditional Statement**

জাভাতে **কন্ডিশনাল স্টেটমেন্ট** ব্যবহার করে প্রোগ্রামের ফ্লো কন্ট্রোল করা হয়। কন্ডিশনাল স্টেটমেন্টগুলি নির্দিষ্ট শর্ত অনুযায়ী কোড ব্লক এক্সিকিউট করে। জাভাতে প্রধান কন্ডিশনাল স্টেটমেন্টগুলি হলো:

### **১.**if**স্টেটমেন্ট**:

এটি একটি সাধারণ কন্ডিশনাল স্টেটমেন্ট যা শর্ত সত্য (true) হলে একটি কোড ব্লক এক্সিকিউট করে।

#### সিনট্যাক্স:

java

Copy

if (condition) {

// Code to execute if condition is true

}

#### উদাহরণ:

java

Copy

int age = 18;

if (age >= 18) {

System.out.println("You are an adult.");

}

### **২.**if-else**স্টেটমেন্ট**:

এটি if স্টেটমেন্টের একটি এক্সটেনশন। যদি শর্ত সত্য (true) হয়, তাহলে if ব্লক এক্সিকিউট হয়; অন্যথায় else ব্লক এক্সিকিউট হয়।

#### সিনট্যাক্স:

java

Copy

if (condition) {

// Code to execute if condition is true

} else {

// Code to execute if condition is false

}

#### উদাহরণ:

java

Copy

int age = 16;

if (age >= 18) {

System.out.println("You are an adult.");

} else {

System.out.println("You are a minor.");

}

### **৩.**if-else-if**ল্যাডার**:

এটি একাধিক শর্ত চেক করার জন্য ব্যবহৃত হয়। যদি প্রথম শর্ত সত্য না হয়, তাহলে পরবর্তী শর্ত চেক করা হয়।

#### সিনট্যাক্স:

java

Copy

if (condition1) {

// Code to execute if condition1 is true

} else if (condition2) {

// Code to execute if condition2 is true

} else {

// Code to execute if all conditions are false

}

#### উদাহরণ:

java

Copy

int marks = 85;

if (marks >= 90) {

System.out.println("Grade: A");

} else if (marks >= 80) {

System.out.println("Grade: B");

} else if (marks >= 70) {

System.out.println("Grade: C");

} else {

System.out.println("Grade: D");

}

### **৪.**switch**স্টেটমেন্ট**:

এটি একাধিক শর্ত চেক করার জন্য ব্যবহৃত হয় এবং if-else-if ল্যাডারের একটি বিকল্প। এটি একটি ভেরিয়েবলের মান অনুযায়ী বিভিন্ন কেস এক্সিকিউট করে।

#### সিনট্যাক্স:

java

Copy

switch (expression) {

case value1:

// Code to execute if expression == value1

break;

case value2:

// Code to execute if expression == value2

break;

default:

// Code to execute if no case matches

}

#### উদাহরণ:

java

Copy

int day = 3;

switch (day) {

case 1:

System.out.println("Monday");

break;

case 2:

System.out.println("Tuesday");

break;

case 3:

System.out.println("Wednesday");

break;

default:

System.out.println("Invalid day");

}

### **৫.**ternary**অপারেটর**:

এটি একটি শর্টহ্যান্ড কন্ডিশনাল স্টেটমেন্ট যা if-else এর একটি সংক্ষিপ্ত রূপ।

#### সিনট্যাক্স:

java

Copy

variable = (condition) ? expression1 : expression2;

#### উদাহরণ:

java

Copy

int age = 20;

String status = (age >= 18) ? "Adult" : "Minor";

System.out.println(status); // Output: Adult

### **৬.**nested if**স্টেটমেন্ট**:

এটি একটি if স্টেটমেন্টের ভিতরে আরেকটি if স্টেটমেন্ট ব্যবহার করা।

#### সিনট্যাক্স:

java

Copy

if (condition1) {

if (condition2) {

// Code to execute if both conditions are true

}

}

#### উদাহরণ:

java

Copy

int age = 20;

boolean isStudent = true;

if (age >= 18) {

if (isStudent) {

System.out.println("You are an adult student.");

}

}

### **৭.**assert**স্টেটমেন্ট**:

এটি ডিবাগিং এর সময় ব্যবহার করা হয় যাতে প্রোগ্রামের কোনো শর্ত সত্য কিনা তা চেক করা যায়। যদি শর্ত মিথ্যা (false) হয়, তাহলে একটি AssertionError থ্রো হয়।

#### সিনট্যাক্স:

java

Copy

assert condition : "Error message";

#### উদাহরণ:

java

Copy

int age = 15;

assert age >= 18 : "Age must be 18 or above";

### **সারসংক্ষেপ**:

| **কন্ডিশনাল স্টেটমেন্ট** | **ব্যবহার** |
| --- | --- |
| if | একটি শর্ত চেক করে কোড এক্সিকিউট করে। |
| if-else | দুটি শর্ত অনুযায়ী কোড এক্সিকিউট করে। |
| if-else-if | একাধিক শর্ত চেক করে কোড এক্সিকিউট করে। |
| switch | একটি ভেরিয়েবলের মান অনুযায়ী বিভিন্ন কেস এক্সিকিউট করে। |
| ternary অপারেটর | শর্টহ্যান্ড if-else স্টেটমেন্ট। |
| nested if | একটি if স্টেটমেন্টের ভিতরে আরেকটি if স্টেটমেন্ট। |
| assert | ডিবাগিং এর সময় শর্ত চেক করে। |

### **কোনটি কখন ব্যবহার করবেন?**:

* যদি শুধুমাত্র একটি শর্ত চেক করতে হয়, তাহলে if ব্যবহার করুন।
* যদি দুটি শর্ত চেক করতে হয়, তাহলে if-else ব্যবহার করুন।
* যদি একাধিক শর্ত চেক করতে হয়, তাহলে if-else-if বা switch ব্যবহার করুন।
* যদি শর্টহ্যান্ডে শর্ত চেক করতে হয়, তাহলে ternary অপারেটর ব্যবহার করুন।
* যদি নেস্টেড শর্ত চেক করতে হয়, তাহলে nested if ব্যবহার করুন।
* যদি ডিবাগিং এর সময় শর্ত চেক করতে হয়, তাহলে assert ব্যবহার করুন।

এই কন্ডিশনাল স্টেটমেন্টগুলি ব্যবহার করে আপনি আপনার প্রোগ্রামের ফ্লো কন্ট্রোল করতে পারবেন। 😊