# Day 14: Conditional Statements in Python & Java

## 1) Why Do We Need Conditional Statements?

Conditional statements allow programs to make decisions based on conditions. They control the flow of execution by checking whether certain conditions are true or false. Without conditional statements, programs would execute sequentially without any logic or branching.

Example Scenarios:  
• Deciding whether a user is logged in before allowing access to a page.  
• Validating payment success before confirming an order.  
• Handling errors gracefully instead of crashing the application.

## 2) Where Are They Used in the Industry?

a) Logical Bugs: Example - Shopping cart issue on Amazon where wrong pricing was applied due to missing conditional checks.  
b) Error Handling: Example - CrowdStrike outage issue where missing or incorrect condition handling caused a major disruption. (Blue screen of death)  
In real-world applications, conditional statements are everywhere — from validating inputs to controlling critical operations in large-scale systems.

## 3) Types of Conditional Statements

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Description | Python Example | Java Example |
| if | Executes code if condition is true. | if x > 0:  print('Positive') | if(x > 0){  System.out.println('Positive'); } |
| if-else | Executes one block if condition is true, otherwise another. | if x > 0:  print('Positive') else:  print('Non-positive') | if(x > 0){  System.out.println('Positive'); }else{  System.out.println('Non-positive'); } |
| if-elif-else | Checks multiple conditions. | if x > 0:  print('Positive') elif x == 0:  print('Zero') else:  print('Negative') | if(x > 0){  System.out.println('Positive'); }else if(x == 0){  System.out.println('Zero'); }else{  System.out.println('Negative'); } |
| Nested if | if inside another if. | if x > 0:  if x < 10:  print('Single-digit positive') | if(x > 0){  if(x < 10){  System.out.println('Single-digit positive');  } } |
| Conditional (ternary) | Shorthand for simple if-else. | msg = 'Positive' if x > 0 else 'Non-positive' | String msg = (x > 0) ? 'Positive' : 'Non-positive'; |
| Switch  (Match in Python) | Used for multiple fixed options (Java only). | day = 3  match day:  case 1:  print("Monday")  case 2:  print("Tuesday")  case 3:  print("Wednesday")  case \_:  print("Other day") | switch(day){  case 1: System.out.println('Mon'); break;  case 2: System.out.println('Tue'); break;  default: System.out.println('Other'); } |

## 4) How to Write Clear and Clean Conditional Statements?

• Use meaningful variable names.  
• Avoid deep nesting – use guard clauses or early returns.  
• Break complex conditions into smaller functions.  
• Use switch-case when dealing with multiple discrete options (in Java).  
• Comment where the logic is not obvious.