

# If-Else

## Conditional Statements and Flow Control

### 1. Introduction to Conditional Statements

- **Overview of Control Flow**
  - How conditionals (if/else) affect the program's execution path
  - The importance of runtime decisions in code
- **Real-World Applications**
  - Age checks (for restricted access)
  - Access control (e.g., bouncer scenario)
  - Setting or canceling notifications based on user input

### 2. Basic If/Else Syntax

#### 1. if Statement

- Structure: `if condition:`
- Example: Checking if one variable is greater than another
- Whitespace and indentation rules in Python

#### 2. else Statement

- Structure: `else:`
- Default or "catch-all" branch when `if` condition is false

#### 3. elif Statement

- Multiple conditions in sequence
- How Python evaluates them in order

**Example:**

```
if b > a:  
    print("b is greater than a")
```

```
elif a == b:
    print("a and b are equal")
else:
    print("a is greater than b")
```

## 3. Short-Hand If and Ternary Operators

### 4. Single-Line If

- Placing the action on the same line as the condition:

```
if a > b: print("a is greater than b")
```

### 5. Ternary Operator (One-Line If/Else)

- Syntax:

```
print("a is greater") if a > b else print("b is greater")
```

- Understanding three operands (condition, true-expression, false-expression)

### 6. Chained Ternary Expressions

- Handling multiple conditions inline
- Syntax can become hard to read use carefully

## 4. Short-Circuiting and Logical Operators

### 7. Logical Operators: `and` , `or` , `not`

### 8. Short-Circuiting Behavior

- `or` : stops evaluating if the first condition is `True`
- `and` : stops evaluating if the first condition is `False`

### 9. Combining Conditions

- Example scenario with `if y > x or x > z or z > y:`
- Common pitfalls when using chained conditions

## 5. Nested If Statements

### 10. Definition: if statements within another if block

### 11. Use Cases

- More granular checks (e.g., if above 10, then check if above 20)

- Age-check scenario:

```
if oldSucker >= allowedAge:
    pass
elif youngGirl >= allowedAge:
    pass
else:
    print("Minor not allowed")
```

## 12. Readability Concerns

- When to refactor nested conditions into separate functions

## 6. The `pass` Statement

- Why `pass` ?
  - Placeholder to avoid `IndentationError` or `SyntaxError` in an empty code block
- Example Usage:

```
if some_condition:
    pass # Future logic goes here
```

## 7. Syntax and Semantics

### 13. Syntax

- Proper indentation is crucial in Python (4 spaces recommended)
- Colons ( `:` ) after `if` , `elif` , `else`

### 14. Semantics

- Understanding how the interpreter evaluates conditions
- The difference between “compiling” time vs. “runtime” checks

### 15. Runtime Decisions vs. Static Code

- Code “flow” is fixed, but the path taken depends on the evaluated conditions (user or external factors)

## 8. Practical Scenarios and Examples

### 16. Age Restriction (Bouncer)

- If age  $\geq 18$ , allow entry; else deny

#### 17. Comparisons with Variables

- a vs. b, b vs. a, multiple conditions with `and` / `or`

#### 18. User Interaction

- Accept user input (e.g., seat booking or cancellation)
- Validate or handle with if/else

#### 19. Nested if with Ternary

- Combining short-hand if with an inner check

## 9. Common Pitfalls and Best Practices

#### 20. Indentation Errors

- Always align code blocks correctly

#### 21. Excessive Nested If

- Can reduce readability consider refactoring

#### 22. Overuse of Ternary Operators

- One-liners can become confusing if chained excessively

#### 23. Short-Circuiting Surprises

- Understanding how `and` / `or` skip evaluating subsequent conditions

#### 24. Testing Edge Cases

- e.g., boundary conditions, equal or not-equal checks

## 10. Lab Exercises / Assignments

#### 25. Temperature Checker

- If temperature > 30, print "It's hot"; elif 20–30 "Warm"; else "Cold"

#### 26. Grade Calculator

- `if score >= 90: A` , `elif score >= 80: B` , etc.

#### 27. Nested Decision

- Validate user's age, then check if they have a membership before granting access

#### 28. Ternary Practice

- Single-line checks for multiple conditions (one-liner if-else)

## 29. Logical Operators

- Combine conditions: if user is "admin" OR "moderator," allow editing; else not