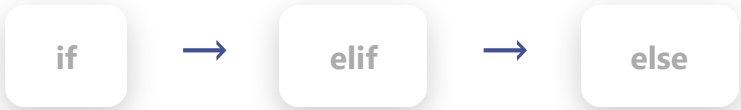
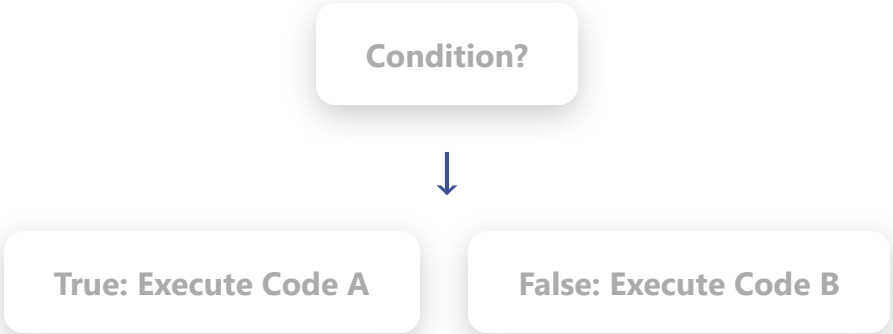
Master the Art of Decision Making in Code



What are Conditional Statements?

Conditional statements allow your program to make decisions based on certain conditions. They enable your code to execute different blocks of code depending on whether specific conditions are true or false.

Think of it like this: Just like you make decisions in real life ("If it's raining, I'll take an umbrella"), your program can make decisions too!



The if Statement

The `if` statement is the most basic conditional statement. It executes a block of code only if a specified condition is true.

```
if condition:
    # code to execute if condition is true
```

Basic Example:

```
# Check if a number is positive number = 10 if number > 0:
print("The number is positive!") print(f"The number is
{number}")
```

Output:

```
The number is positive!
The number is 10
```

Real-world Example:

```
# Check if someone can vote age = 18 if age >= 18: print("You
are eligible to vote!") print("Please register if you haven't
already.")
```

Output:

```
You are eligible to vote!
Please register if you haven't already.
```

The else Statement

The `else` statement provides an alternative path when the `if` condition is false. It's like saying "otherwise, do this instead."

```
if condition:
# code if condition is true
else:
# code if condition is false
```

Basic Example:

```
# Check if a number is even or odd number = 7 if number % 2 ==
0: print(f"{number} is even") else: print(f"{number} is odd")
```

Output:

```
7 is odd
```

Password Checker Example:

```
# Simple password checker password = "python123"
correct_password = "python123" if password == correct_password:
print("Access granted! Welcome!") else: print("Access denied!
Incorrect password.")
```

Output:

```
Access granted! Welcome!
```

The elif Statement

The `elif` (else if) statement allows you to check multiple conditions in sequence. It's perfect when you have more than two possible outcomes.

```
if condition1:
# code if condition1 is true
elif condition2:
```

```
    # code if condition2 is true
else:
    # code if all conditions are false
```

Grade Calculator Example:

```
# Grade calculator based on score score = 85 if score >= 90: grade = "A" print(f"Excellent! Your grade is {grade}") elif score >= 80: grade = "B" print(f"Good job! Your grade is {grade}") elif score >= 70: grade = "C" print(f"Average. Your grade is {grade}") elif score >= 60: grade = "D" print(f"Below average. Your grade is {grade}") else: grade = "F" print(f"Failed. Your grade is {grade}")
```

Output:

```
Good job! Your grade is B
```

Complex Examples

Weather Advisory System:

```
# Weather advisory based on temperature temperature = 35 humidity = 80 if temperature > 40: print("🔥 Extreme heat warning! Stay indoors.") elif temperature > 30: if humidity > 70: print("🌡️ Hot and humid! Stay hydrated.") else: print("🌞 Hot day! Perfect for swimming.") elif temperature > 20: print("🌤️ Pleasant weather! Great for outdoor activities.") elif temperature > 10: print("🧥 Cool weather. Consider wearing a jacket.") else: print("🥶 Cold weather! Bundle up warm.")
```

Output:

```
🌡️ Hot and humid! Stay hydrated.
```

Login System with Multiple Checks:

```
# Advanced login system username = "admin" password = "secret123" attempts = 2 if attempts >= 3: print("❌ Account locked due to too many failed attempts.") elif username == "admin" and password == "secret123": print("✅ Admin access granted! Welcome, administrator.") elif username == "user" and password == "pass123": print("✅ User access granted! Welcome, user.") elif username in ["admin", "user"]: print("❌ Incorrect password. Please try again.") else: print("❌ Username not found. Please check your credentials.")
```

Output:

```
✅ Admin access granted! Welcome, administrator.
```

Nested Conditional Statements

You can place conditional statements inside other conditional statements. This is called nesting and allows for more complex decision-making logic.

Student Performance Analyzer:

```
# Analyze student performance with multiple criteria math_score = 85 english_score = 78 attendance = 95 if attendance >= 90: print("✅ Good attendance!") if math_score >= 80 and english_score >= 80: print("🏆 Excellent performance in both subjects!") print("🎓 Eligible for honor roll.") elif math_score >= 80 or english_score >= 80: print("👍 Good performance in at least one subject.") if math_score > english_score: print("💪 Stronger in Mathematics.") else: print("📚 Stronger in English.") else: print("⚠️ Needs improvement in both subjects.") else: print("❌ Poor attendance. Focus on attending classes regularly.")
```

Output:

```
✅ Good attendance!
👍 Good performance in at least one subject.
💪 Stronger in Mathematics.
```

Using Logical Operators

Logical operators help you combine multiple conditions in a single if statement:

and - Both conditions must be true

or - At least one condition must be true

not - Reverses the condition

Examples with Logical Operators:

```
# Online shopping discount calculator age = 25 is_student = True
order_amount = 150 is_first_time = False # Using 'and' operator
if age < 30 and is_student: print("🎓 Student discount: 20%
off!") discount = 0.20 # Using 'or' operator elif order_amount >
100 or is_first_time: print("🛍️ Special discount: 10% off!")
discount = 0.10 # Using 'not' operator elif not is_first_time
and age >= 65: print("👴 Senior citizen discount: 15% off!")
discount = 0.15 else: print("💰 Regular price applies.")
discount = 0.00 final_amount = order_amount * (1 - discount)
print(f"Final amount: ${final_amount:.2f}")
```

Output:

```
🎓 Student discount: 20% off!
Final amount: $120.00
```

Interactive Demo

Try this interactive age classifier! Enter an age and see how conditional statements work:

Age Classifier

Enter your age

Classify Age

```
# The code behind the demo def classify_age(age): if age < 0: return "Invalid age!" elif age < 13: return "👶 Child" elif age < 20: return "👦 Teenager" elif age < 60: return "👨 Adult" elif age < 120: return "👴 Senior" else: return "🎂 Wow, that's really old!"
```

Best Practices

🏆 Tips for Writing Great Conditional Statements

- ✓ **Use clear, descriptive conditions:** Make your conditions easy to understand
- ✓ **Keep it simple:** Avoid overly complex nested conditions when possible
- ✓ **Use logical operators wisely:** Combine conditions efficiently with and, or, not
- ✓ **Consider the order:** Put most likely conditions first for better performance
- ✓ **Handle edge cases:** Think about unusual inputs and handle them gracefully

- ✓ **Use consistent indentation:** Python requires proper indentation (4 spaces is standard)
- ✓ **Add comments:** Explain complex conditions for future reference
- ✓ **Test thoroughly:** Test all possible paths through your conditional statements

Common Mistakes to Avoid:

```
# ❌ Wrong: Using = instead of == for comparison # if age = 18:
# This assigns, doesn't compare! # ✅ Correct: Use == for
comparison if age == 18: print("Just turned 18!") # ❌ Wrong:
Forgetting the colon # if age >= 18 # print("Adult") # ✅
Correct: Always include the colon if age >= 18: print("Adult")
```

Summary

Conditional statements are fundamental building blocks that make your programs intelligent and responsive. Here's what we've learned:

if Statement
Executes code when condition is true

else Statement
Provides alternative when if is false

elif Statement
Handles multiple conditions efficiently

Key Takeaways:

- ✓ Conditional statements control program flow based on conditions
- ✓ Use if for single conditions, if-else for two outcomes, if-elif-else for multiple outcomes
- ✓ Logical operators (and, or, not) help combine conditions
- ✓ Proper indentation is crucial in Python
- ✓ Always test your conditions thoroughly
- ✓ Keep conditions simple and readable

🎉 Congratulations! You now understand conditional statements in Python!