Functions: Conditional Branching

September 12, 2025

Control Statements and Logical Operators

Control statements determine the order in which your code is executed. Logical operators help you build complex conditions for these statements.

Logical Operators: Building Complex Conditions

Logical operators combine multiple conditional expressions into a single True or False result.

and: Returns True if **both** operands are True.

Truth Table:

Operand 1	Operand 2	Result
True	True	True
True	False	False
False	True	False
False	False	False

or: Returns True if **at least one** operand is True.

Truth Table:

Operand 1	Operand 2	Result
True	True	True
True	False	True
False	True	True
False	False	False

- **not:** Reverses the logical state. not True is False, and not False is True.

- **Sample Code with Logical Operators:**

```
# Example 1: `and`
is_admin = True
is_logged_in = True
if is_admin and is_logged_in:
    print("Access granted to admin panel.")

# Example 2: `or`
is_weekend = False
has_day_off = True
if is_weekend or has_day_off:
    print("Time to relax!")

# Example 3: `not`
is_valid = False
if not is_valid:
    print("Data is invalid. Please correct it.")
```

Control Statements: if, elif, else

This structure allows your program to make decisions and follow a specific "branch" of code.

- **if:** The most basic control statement. The code block is executed only if the condition is True.

```
score = 95
if score > 90:
    print("You got an A.")
```

- **if...else:** Provides a default path to take if the initial condition is False.

```
age = 17
if age >= 18:
    print("You are an adult.")
else:
    print("You are a minor.")
```

- **if...elif...else:** Used to check for another condition if the previous if or elif conditions were False. You can have multiple elif blocks.

```
grade = 85
if grade >= 90:
    print("Excellent! You got an A.")
elif grade >= 80:
    print("Great job! You got a B.")
elif grade >= 70:
    print("Good effort. You got a C.")
else:
    print("You got a D or F. Keep studying!")
```