Day19_Conditional_Statements

June 7, 2025

Today I explored how to use **conditional statements** in Python. These are the building blocks of decision-making in programs. With conditionals, I learned how to write code that behaves differently based on certain conditions. This allows a program to **think logically**, like a human, and choose what to do based on the situation.

What are Conditional Statements?

Conditional statements are used to perform **different actions based on different conditions**. They control the **flow of execution** in a program.
In simple words, if something is true — do this, otherwise — do something else.

Why and Where Are They Used?

Conditional statements are used in **almost every program**, such as: - Checking user input - Validating data - Making decisions (e.g., if a number is even or odd) - Controlling loops - Error handling - Game logic (e.g., if player score > high score, then update)

Advantages:

- Helps in writing logical and flexible programs
- Makes code more **interactive** and **responsive**
- Essential for **real-life problem solving** (e.g., filters, conditions, eligibility checks)

Disadvantages / Caution:

- Too many separate `if` statements can **slow down performance**
- If you use only `if-if-if`, **Python checks each condition**, even if one is already true.
- Use `if-elif-else` structure when only **one** condition should match — this avoids unnecessary checks and improves speed.

1 Syntax of Conditional Statements in Python & Examples:

1.1 if

Example 1: Simple if condition with True and False

[1]:
```python
if True:  # This condition is always True, so the block runs
    print('Data Science')
```

```
Data Science
```

[2]:
```python
if False:  # This condition is always False, so nothing is printed
    print('Data Science')

print('bye for now')  # Runs regardless of the condition above
```

```
bye for now
```

[3]:
```python
if True:
    print('Data Science')
print('bye for now')
```

```
Data Science
bye for now
```

Example 2: if condition checking even or odd

[4]:
```python
x = 4
r = x % 2  # Modulus operator returns remainder

if r == 0:
    print('Even number')
```

```
Even number
```

[5]:
```python
# Another example with an odd number
x = 5
r = x % 2

if r == 0:
    print('Even number')  # This won't execute since 5 is odd
```

1.2 if-else

Example 3: Adding else block for better feedback

[6]:
```python
x = 5
r = x % 2

if r == 0:
    print('Even number')
else:
    print('Odd number')  # This will be printed
```

```
Odd number
```

[7]:
```python
# Another example
x = 8
r = x % 2
```

```
if r == 0:
    print('Even number')  # Will be printed
else:
    print('Odd number')
```

```
Even number
```

Example 4: Using multiple if blocks (not recommended)

[8]:

```
x = 13
r = x % 2

if r == 0:
    print('Even number')

if r == 1:
    print('Odd number')  # Works, but 'if-else' is better for such binary checks
```

```
Odd number
```

[9]:

```
# Using 'if' with 'not equal'
x = 19
r = x % 2

if r == 0:
    print('Even number')

if r != 0:
    print('Odd number')
```

```
Odd number
```

1.3 Nested if

Example 5: Nested if statements

[10]:

```
x = 3
r = x % 2

if r == 0:
    print('Even number')
    if x > 5:
        print('Greater number')  # This will not run
else:
    print('Odd Number')  # Will be printed
```

```
Odd Number
```

[11]:

```
# Another example
x = 8
```

```
r = x % 2

if r == 0:
    print('Even number')  # Will run
    if x > 5:
        print('Greater number')  # Will also run
```

```
Even number
Greater number
```

[12]:

```
# Example with nested if and else
x = 4
r = x % 2

if r == 0:
    print('Even number')
    if x > 5:
        print('Greater number')
    else:
        print('Smaller number')  # Will be printed
else:
    print('Odd Number')
```

```
Even number
Smaller number
```

1.4 if-elif-else

Example 6: if-elif-else chain

[13]:

```
x = 9

if x == 1:
    print('One')
elif x == 2:
    print('Two')
elif x == 3:
    print('Three')
elif x == 4:
    print('Four')
else:
    print('Number not found')  # Will be printed because x is 9
```

```
Number not found
```

2 Python Conditional Statements - Real-Life Examples

Covered:

- if
- if-else
- if-elif-else
- nested if
- logical operators
- one-line if
- if inside a loop

Conditional statements are powerful tools to write logic-driven programs that make decisions just like humans!

2.1 Simple if statement

Scenario: Check if your wallet has enough money for pani puri!

[14]:

```
money = 30  # rupees
pani_puri_price = 20

if money >= pani_puri_price:
    print("You can enjoy pani puri!  ")
```

```
You can enjoy pani puri!
```

2.2 if-else statement

Scenario: Check if it's raining – take an umbrella or not

[15]:

```
is_raining = False

if is_raining:
    print("Take your umbrella  ")
else:
    print("Enjoy the sunshine  ")
```

```
Enjoy the sunshine
```

2.3 if-elif-else statement

Scenario: Traffic light behavior

[16]:

```
traffic_light = "yellow"

if traffic_light == "green":
    print("Go  ")
elif traffic_light == "yellow":
    print("Slow down  ")
elif traffic_light == "red":
    print("Stop  ")
else:
    print("Invalid light signal  ")
```

```
Slow down
```

2.4 Nested if statement

Scenario: Online exam result checker

[17]:

```python
marks = 85

if marks >= 35:
    print("You passed the exam  ")
    if marks >= 75:
        print("Great job! You scored distinction  ")
    else:
        print("Good, but you can aim higher!  ")
else:
    print("Unfortunately, you failed the exam  ")
```

```
You passed the exam
Great job! You scored distinction
```

2.5 if-elif-else with multiple elif

Scenario: Grade assignment system

[18]:

```python
score = 92

if score >= 90:
    print("Grade: A+  ")
elif score >= 80:
    print("Grade: A")
elif score >= 70:
    print("Grade: B")
elif score >= 60:
    print("Grade: C")
elif score >= 50:
    print("Grade: D")
else:
    print("Grade: F - Needs Improvement  ")
```

```
Grade: A+
```

2.6 if with logical operators (and/or/not)

Scenario: Checking user login credentials

[19]:

```python
# AND Logic
username = "akshay"
password = "1234"

if username == "akshay" and password == "1234":
    print("Login successful  ")
else:
    print("Login failed  ")
```

```
Login successful
```

[20]:

```
# OR logic
age = 16
has_permission = True

if age >= 18 or has_permission:
    print("You are allowed to enter the event  ")
else:
    print("Access denied  ")
```

```
You are allowed to enter the event
```

[21]:

```
# NOT logic
is_banned = False

if not is_banned:
    print("You can play the game  ")
else:
    print("Your account is banned  ")
```

```
You can play the game
```

2.7 One-line if (Ternary operator)

Scenario: Assign status based on age

[22]:

```
age = 21
status = "Adult" if age >= 18 else "Minor"
print(f"Status: {status}")
```

```
Status: Adult
```

2.8 if inside a loop

Scenario: Filter even numbers from a list

[23]:

```
numbers = [1, 4, 7, 10, 15, 22]

print("Even numbers in the list:")
for num in numbers:
    if num % 2 == 0:
        print(num)
```

```
Even numbers in the list:
4
```

```
10
22
```