Conditional Statements in Python

Conditional statements allow Python to make decisions based on conditions.

1. Two cases (if else)

An **if–else statement** allows a program to make decisions by testing a condition. If the condition is true, one block of code runs; otherwise, another block runs. The condition can be **simple**, such as a single comparison, or **compound**, combining multiple conditions using logical operators like **AND** and **OR**, which helps the program handle more complex situations clearly and efficiently.

1. 1. Simple Condition

A simple condition checks (evaluates) one condition only.

Syntax

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

Example 1

Write a Python program that reads a student's grade and then displays whether the student has passed or failed.

```
Grade = float(input('enter the grade of student'))
if Grade >= 10:
   print('the student is: ')
   print('Passed')
else:
   print('the student is: ')
   print('failed')
```

Example 2

Write a Python program that checks if a number is even or odd.

```
Num = int(input("enter an integer"))
if Num % 2 == 0:
   print(Num, "is even")
```

```
else:
    print(Num, "is odd")
```

Example 3

Write a Python program that asks the user to enter two words. If the second word is not equal to the first one, print "Words are different", otherwise "Same words".

```
Word1 = input("enter the first word\n")
Word2 = input("enter the second word\n")
if Word1 != Word2:
    print("the two words are different")
else:
    print("the two words are the same")
```

Example 4

Write a Python program that calculates the absolute value of an integer.

```
Num = float(input("enter a number"))
if Num >= 0:
    num_abs = Num
else:
    num_abs = -Num
print(num_abs)
print(abs(Num))
```

1. 2. Compound Conditions

Sometimes we need to check more than one condition at the same time. To do this, we use logical operators.

Operator	Syntax	Use
and	c1 and c2	when all conditions must be true
or	c1 or c2	at least one condition must be true

Example 5

Write a Python program that reads a day then displays whether it is a weekend or a weekday.

```
day = input("enter the day name ")
if day == "saturday" or day == "friday":
    print("is a weekend ")
```

```
else:
    print("is a weekday")
```

Example 6

Write a Python program that asks the user to read a number and checks if it is between 1 and 100.

```
Num = float(input('enter a number'))
if Num >= 1 and Num <= 100:
    print("is in the range 1-100")
else:
    print("is out of the range")
```

Example 7

Write a Python program that checks if a word starts with 'a' or ends with 'z'.

```
Word = input("Enter a word")
if Word.startswith("a") or Word.endswith("z"):
    print("condition true")
else:
    print("condition false")
```

2. Multiple Cases (if, elif, else)

Multiple cases occur when a program must choose between **more than two cases**. The program checks conditions one by one and runs the code for the first condition that is true. **In Python, this is done using if, elif, and else.**

Syntax

```
if condition1:
    block1
elif condition2:
    block2
elif condition3:
    block3
...
else:
    blockN
```

Example 8

Write a Python program that reads a number and then displays whether it is positive, negative, or zero.

```
num = float(input("enter a float number: "))
if num > 0:
    print(num, ": is positive ")
elif num < 0:
    print(num, ": is negative")
else:
    print(num, ": is zero")
```

Example 9: Compare Two Numbers

Write a Python program that reads two numbers and prints which one is greater or if they are equal.

```
Num1 = float(input("enter your first number "))
Num2 = float(input("enter your second number "))
if Num1 > Num2:
    print(Num1, "is greater than ", Num2)
elif Num1 < Num2:
    print(Num2, "is greater than ", Num1)
else:
    print(Num1, "is equal to ", Num2)
```

Example 10

Write a Python program that asks the user to enter their score and prints the corresponding grade.

```
Score = float(input("Enter your score"))
if Score >= 18:
    print("Excellent")
elif Score >= 16:
    print("very good")
elif Score >= 14:
    print("good")
elif Score >= 12:
    print("fairly good")
```

```
elif Score >= 10:
    print("Acceptable")
else:
    print("Fail")
```

Example 11

Write a Python program that asks the user to enter a number from 1 to 7 and prints the corresponding day of the week:

```
day_number = int(input("Enter a number (1-7): "))

if day_number == 1:
    print("Sunday")
elif day_number == 2:
    print("Monday")
elif day_number == 3:
    print("Tuesday")
elif day_number == 4:
    print("Wednesday")
elif day_number == 5:
    print("Thursday")
elif day_number == 6:
    print("Friday")
elif day_number == 7:
    print("Saturday")
else:
    print("Invalid input")
```

Example 12

Write a Python program that asks the user for their age and prints their age group:

```
Age = int(input("enter your age: "))

if Age < 13:
    print("Child")
elif Age <= 18:
    print("Teenager")
elif Age < 60:
    print("Adult")
else:
    print("Senior")
```

3. Nested if

A **nested if** is an **if statement placed inside another if statement**. It allows a program to make decisions based on **multiple levels of conditions**. First, the outer if is evaluated; if it is true, the inner if is checked. This structure helps handle more complex situations where **one decision depends on another**.

Syntax

```
if condition1:
    if condition2:
        code1  # condition1 and condition2 are true
    else:
        code2  # condition1 is true, condition2 is false
else:
    code3 # condition1 is false, whether condition2 is true or not
```

Example 13

Write a Python program that finds the maximum among three numbers.

```
Num1 = float(input("Enter your first number "))
Num2 = float(input("Enter your second number "))
Num3 = float(input("Enter your third number "))

if Num1 > Num2:
    if Num1 > Num3:
        print(Num1, ": is the largest")
    else:
        print(Num3, ": is the largest")
else:
    if Num2 > Num3:
        print(Num2, ": is the largest ")
    else:
        print(Num3, ": is the largest")
```

Example 14

Write a Python program that reads the user's age and gender. If the age is 18 or older, print 'Adult'. If the gender is 'female', print 'Adult Woman'. Otherwise, print 'Adult Man'. If the age is under 18, print 'Minor'.

```
Age = int(input("enter your age: "))
Gender = input("enter your gender: ")
if Age >= 18:
    if Gender == "female":
        print("Adult female")
    else:
        print("Adult male")
else:
    print("Minor")
```

Optimized solution

```
Age = int(input("enter the age\n"))
Gender = input("enter the Gender ")
if Age >= 18:
    if Gender.lower() == "male":
        print("Adult male")
    elif Gender.lower() == "female":
        print("Adult female")
    else:
        print("Adult")
        print("but error of gender")
elif Gender.lower() == "male":
    print("minor male")
elif Gender.lower() == "female":
    print("minor female")
else:
    print("Minor but ")
    print("error of gender")
```

Example 15

Write a program that checks if a word contains the letter 'm', 'd', both, or neither.

```
Word = input("enter a word: ")
if "d" in Word:
    if "m" in Word:
        print(Word + ": contains both d and m")
    else:
        print(Word, ": contains d only")
elif "m" in Word:
    print(Word, ": contains m only")
else:
    print(Word, ": contains neither d nor m")
```