



# DAY 04: CONTROL STRUCTURES

TOPICS – IF-ELSE, FOR/WHILE  
LOOPS, BREAK/CONTINUE

#IDC30DAYSCHALLENGE

By Anmol Gambhir.

# 1. IF-ELSE STATEMENTS IN PYTHON

**if**, **elif**, and **else** are used to control the flow of a program based on conditions. It allows the program to make decisions — do one thing if a condition is true, and another if it's false.

## **Syntax:**

**if** condition:

    # Code block if condition is True

**elif** another\_condition:

    # Code block if the second condition is True

**else:**

    # Code block if all conditions are False

## **Example:**

```
age = 20
```

```
if age >= 18:
```

```
    print("You can vote.")
```

```
else:
```

```
    print("You cannot vote yet.")
```

## **Explanation:**

age >= 18 is the condition.

If it's true, it prints “You can vote.”

If false, it prints “You cannot vote yet.”

# 2. For Loop in Python

A **for** loop is used to repeat a block of code for each item in a sequence (like a list, string, or range of numbers).

It's used when you know in advance how many times you want to loop.

## **Syntax:**

```
for variable in sequence:
```

```
    # code block
```

## **Example:**

```
for i in range(1, 6):
```

```
    print(i)
```

## **Output:**

1

2

3

4

5

# 3. While Loop in Python

A **while** loop keeps executing the code as long as a condition is true. It's used when you don't know how many times to loop, but you want to continue until a condition changes.

## **Syntax:**

```
while condition:  
    # code block
```

## **Example:**

```
i = 1  
while i <= 5:  
    print(i)  
    i += 1
```

## **Output:**

```
1  
2  
3  
4  
5
```



## 4. Break & Continue Statement in Python

The **break** statement is used to exit a loop immediately, even if the condition hasn't become false yet.

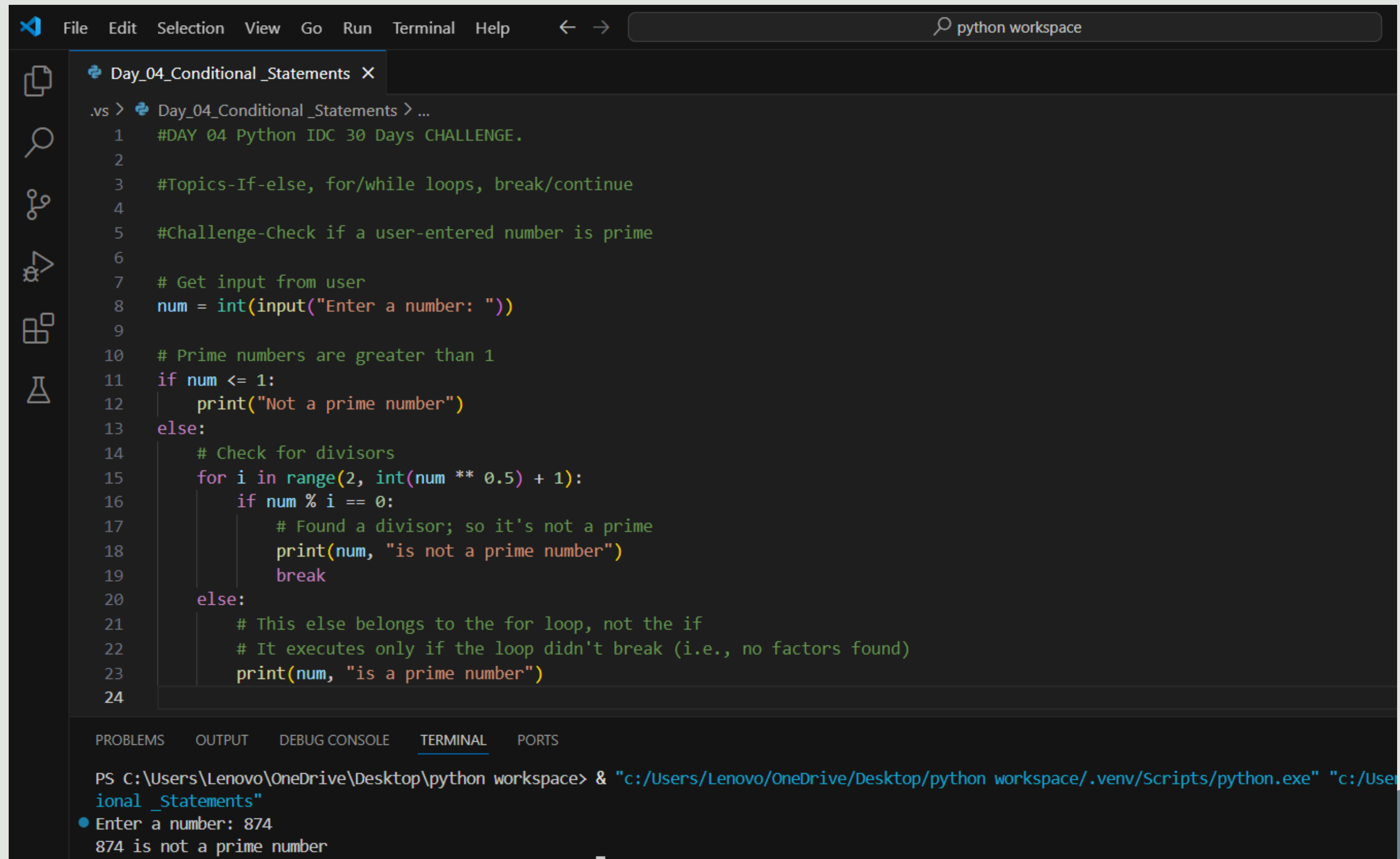
It's useful when a certain condition is met, and there's no need to continue the loop.

The **continue** statement is used to skip the current iteration and go to the next one.

It doesn't stop the whole loop — it just skips that one turn.



# Challenge - Check if a user-entered number is prime



```
File Edit Selection View Go Run Terminal Help python workspace
Day_04_Conditional_Statements X
.vs > Day_04_Conditional_Statements > ...
1 #DAY 04 Python IDC 30 Days CHALLENGE.
2
3 #Topics-If-else, for/while loops, break/continue
4
5 #Challenge-Check if a user-entered number is prime
6
7 # Get input from user
8 num = int(input("Enter a number: "))
9
10 # Prime numbers are greater than 1
11 if num <= 1:
12     print("Not a prime number")
13 else:
14     # Check for divisors
15     for i in range(2, int(num ** 0.5) + 1):
16         if num % i == 0:
17             # Found a divisor; so it's not a prime
18             print(num, "is not a prime number")
19             break
20     else:
21         # This else belongs to the for loop, not the if
22         # It executes only if the loop didn't break (i.e., no factors found)
23         print(num, "is a prime number")
24

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Lenovo\OneDrive\Desktop\python workspace> & "c:/Users/Lenovo/OneDrive/Desktop/python workspace/.env/Scripts/python.exe" "c:/Users/Lenovo/OneDrive/Desktop/python workspace/Day_04_Conditional_Statements"
● Enter a number: 874
874 is not a prime number
```