

CONTROL STATEMENTS → CODE FLOW

✦ Definition

Control Statements are used to **control the flow of execution** of a program. They decide **which statement will run, how many times it will run, and when execution should stop or skip.**

👉 Without control statements, a program runs **line by line only** (top to bottom).

◆ Types of Control Statements

1. **Conditional Statements** (Decision Making)
 2. **Looping Statements** (Repetition)
 3. **Jumping Statements** (Flow Control)
-

1 CONDITIONAL STATEMENTS

✦ Definition

Conditional statements are used to **check conditions** and execute different blocks of code based on **True or False** results.

◆ 1. IF Statement

✦ Definition

The if statement executes a block of code **only if the condition is True.**

◆ Syntax

if condition:

 statement

◆ Example

```
age = 18
```

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

```
    print("Eligible for voting")
```

✦ If condition is False → nothing executes.

◆ 2. IF ELSE Statement

✦ Definition

The if else statement executes **one block if condition is True** and **another block if condition is False**.

◆ Syntax

if condition:

 statement

else:

 statement

◆ Example

```
num = 5
```

```
if num % 2 == 0:
```

```
    print("Even number")
```

```
else:
```

```
    print("Odd number")
```

◆ 3. ELIF (Else If) Statement

✦ Definition

elif is used to **check multiple conditions** one by one.

◆ Syntax

if condition1:

 statement

elif condition2:

 statement

else:

 statement

◆ Example

```
marks = 75
```

```
if marks >= 90:
```

```
    print("Grade A")
```

```
elif marks >= 60:
```

```
    print("Grade B")
else:
    print("Fail")
```

✦ Python checks conditions **top to bottom**.

◆ 4. NESTED IF ELSE

✦ Definition

An if else inside another if else is called **Nested If Else**.

◆ Syntax

if condition:

```
    if condition:
        statement
```

```
    else:
        statement
```

else:

```
    statement
```

◆ Example

```
num = 10
```

```
if num > 0:
```

```
    if num % 2 == 0:
        print("Positive Even")
```

```
    else:
        print("Positive Odd")
```

```
else:
```

```
    print("Negative number")
```

◆ 5. SHORTHAND IF ELSE (Ternary Operator)

✦ Definition

A **single-line if else** used for short conditions.

◆ Syntax

statement_if_true if condition else statement_if_false

◆ **Example**

```
a = 10
```

```
b = 20
```

```
print("A is greater") if a > b else print("B is greater")
```

◆ **6. MATCH CASE (Python 3.10+)**

✦ **Definition**

match case is used to **compare one value with many cases**, similar to switch-case.

◆ **Syntax**

match value:

```
case x:
```

```
    statement
```

```
case y:
```

```
    statement
```

```
case _:
```

```
    statement
```

◆ **Example**

```
day = 3
```

match day:

```
case 1:
```

```
    print("Monday")
```

```
case 2:
```

```
    print("Tuesday")
```

```
case 3:
```

```
    print("Wednesday")
```

```
case _:
```

```
    print("Invalid day")
```

2 LOOPING STATEMENTS

✦ Definition

Loops are used to **repeat a block of code multiple times**.

◆ 1. FOR Loop

✦ Definition

Used when **number of iterations is known**.

◆ Syntax

for variable in sequence:

```
statement
```

◆ Example

for i in range(1, 6):

```
print(i)
```

◆ 2. WHILE Loop

✦ Definition

Used when **condition-based repetition** is required.

◆ Syntax

while condition:

```
statement
```

◆ Example

```
i = 1
```

```
while i <= 5:
```

```
print(i)
```

```
i += 1
```

3 JUMPING STATEMENTS

✦ Definition

Jumping statements **change the normal flow** of loops.

◆ 1. BREAK

✦ Definition

Stops the loop **completely**.

◆ **Example**

```
for i in range(1, 10):  
    if i == 5:  
        break  
    print(i)
```

◆ **2. CONTINUE**

✦ **Definition**

Skips the **current iteration** and moves to next.

◆ **Example**

```
for i in range(1, 6):  
    if i == 3:  
        continue  
    print(i)
```

◆ **3. PASS**

✦ **Definition**

Used as a **placeholder** when a statement is required syntactically but no action is needed.

◆ **Example**

```
for i in range(5):  
    if i == 2:  
        pass  
    print(i)
```

🔥 **TOP 10 LOGICAL PROGRAMS USING IF ELSE**

1 **Check Even or Odd**

```
num = 7  
if num % 2 == 0:  
    print("Even")
```

else:

```
print("Odd")
```

2 Check Positive, Negative or Zero

```
n = -5
```

```
if n > 0:
```

```
    print("Positive")
```

```
elif n < 0:
```

```
    print("Negative")
```

```
else:
```

```
    print("Zero")
```

3 Largest of Two Numbers

```
a = 10
```

```
b = 20
```

```
if a > b:
```

```
    print("A is largest")
```

```
else:
```

```
    print("B is largest")
```

4 Largest of Three Numbers

```
a, b, c = 10, 25, 15
```

```
if a > b and a > c:
```

```
    print("A largest")
```

```
elif b > c:
```

```
    print("B largest")
```

```
else:
```

```
    print("C largest")
```

5 Leap Year Check

```
year = 2024
```

```
if year % 4 == 0 and year % 100 != 0 or year % 400 == 0:
```

```
    print("Leap Year")
else:
    print("Not Leap Year")
```

6 Voting Eligibility

```
age = 17
if age >= 18:
    print("Can vote")
else:
    print("Cannot vote")
```

7 Grade System

```
marks = 85
if marks >= 90:
    print("A")
elif marks >= 75:
    print("B")
elif marks >= 50:
    print("C")
else:
    print("Fail")
```

8 Check Divisible by 5 and 11

```
num = 55
if num % 5 == 0 and num % 11 == 0:
    print("Divisible by 5 and 11")
else:
    print("Not divisible")
```

9 Simple Calculator

```
a = 10
b = 5
```

```
op = "+"
```

```
if op == "+":
```

```
    print(a + b)
```

```
elif op == "-":
```

```
    print(a - b)
```

```
elif op == "*":
```

```
    print(a * b)
```

```
elif op == "/":
```

```
    print(a / b)
```

```
else:
```

```
    print("Invalid operator")
```

10 Check Character Type

```
ch = 'A'
```

```
if ch.isalpha():
```

```
    print("Alphabet")
```

```
elif ch.isdigit():
```

```
    print("Digit")
```

```
else:
```

```
    print("Special Character")
```
