Conditional Statement

Session Objectives

- ✅ Understand what control statements are.
- ✅ Understand the importance of control statements.
- ✅ Understand the types of control statements.
- ✅ Understand what conditional statements are.
- ✅ Understand what nested conditional statements are.
- ✅ Understand conditional statements with shorthand notations.

Control statements in Python are like traffic signals for your code — they decide which parts of your program should run and when.

Traffic light analogy:

Green → Go → Run code block

Red → Stop → Skip code block

Syntax : If

```
If condition:
    # code block
```

```
val = int(input("Enter the value: "))
if val > 100:
    print(f'{val} is greater than 100')
print("Outside the if conditions, Always work!")
```

```
Enter the value:  11
Outside the if conditions, Always work!
```

```
val = int(input("Enter the value: "))
if val > 100:
    print(f'{val} is greater than 100')
print("Outside the if conditions, Always work!")
```

```
Enter the value:  111
111 is greater than 100
Outside the if conditions, Always work!
```

```python
val = int(input("Enter the value: "))
if val > 100:
    print(f'{val} is greater than 100')
print("Outside the if conditions, Always work!")
```

```
Enter the value:  -1
Outside the if conditions, Always work!
```

```python
num1 = int(input("Enter the num1: "))
num2 = int(input("Enter the num2: "))
if num1 > num2:
    print(f"{num1} is greater than {num2}")
if num2 > num1:
    print(f"{num2} is greater than {num1}")
if num1 == num2:
    print(f"{num1} is equal to {num2}")
if num1 != num2:
    print(f"{num1} is not equal to {num2}")
```

```
Enter the num1:  -99
Enter the num2:  54
54 is greater than -99
-99 is not equal to 54
```

Syntax : If-else

```python
if condition:
    # Code block to execute if the condition is true
else:
    # Code block to execute if the condition is false
```

```python
_bool = True
int(_bool) # 1
```

```
1
```

```python
_bool = False
int(_bool) # 0
```

```
0
```

```python
# If you find any number either positive or negative -> True
# Except 0 : False Other examples can be -> [],(),"",{},set()
```

```python
car_list = ['Taigun','Creta','Slavia','Virtus','Thar','ScorpioN','i10','Verna','Jeep','Lord Alto']
fav_car1 = 'Virtus' # True
fav_car2 = 'Creta' # True
if fav_car1 in car_list or fav_car2 in car_list: # True
    print("We are checking Nested Conditions : ")
    if fav_car1 in car_list and fav_car2 in car_list: TT
        print(f"{fav_car1} & {fav_car2} is available in car_list")
    elif fav_car1 in car_list : TF
        print(f"{fav_car1} is available in car_list")
    elif fav_car2 in car_list : FT
        print(f"{fav_car2} is available in car_list")
    else:
        print(f"{fav_car2} is available in car_list")
else: FF
    print(f"Sorry! We will add it soon....")
```

2 conditons:

TT,TF,FT,FF

Syntax : If-elif-else

```
if condition1:
    # Block if condition1 true
elif condition2:
    # Block if condition2 true
elif condition3:
    # Block if condition3 true
...
else:
    # Block if all conditions are false
```

Syntax : Nested-if

```
if condition1:
    # Block if condition1 is true
    if condition2:
        # Block if both condition1 and condition2 are true
```

Syntax : Nested if-else

```
if condition1:
    if condition2:
        # Block if both condition1 & condition2 true
    else:
        # Block if condition1 true, condition2 false
else:
    # Block if condition1 is false

or
if condition:
    # Block if condition is true
else:
    if another_condition:
        # Block if another_condition is true
    else:
        # Block if another_condition is false
```

Syntax : Nested if-elif-else

```
if condition1:
    if condition2:
        # Block if both condition1 & condition2 true
    elif condition3:
        # Block if condition1 true, condition2 false, condition3 true
    else:
        # Block if condition1 true, others false
elif condition4:
    # Block if condition1 false, condition4 true
else:
    # Block if all conditions false
```

```
val = int(input("Enter the value: "))
if val > 100:
    print(f"{val} is greater than 100.")
else:
    print(f"{val} is less than or equal to 100.")
```

```
Enter the value:  11
11 is less than or equal to 100.
```

```
val = int(input("Enter the value: "))
if val > 100:
    print(f"{val} is greater than 100.")
else:
    print(f"{val} is less than or equal to 100.")
```

```
Enter the value:  111
111 is greater than 100.
```

```
val = int(input("Enter the value: "))
if val > 100:
    print(f"{val} is greater than 100.")
else:
    print(f"{val} is less than or equal to 100.")
```

```
Enter the value:  -999
-999 is less than or equal to 100.
```

```
val = int(input("Enter the value: "))
if val > 100:
    print(f"{val} is greater than 100.")
else:
    print(f"{val} is less than or equal to 100.")
```

```
Enter the value:  a
```

Fix Code

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[12], line 1
----> 1 val = int(input("Enter the value: "))
      2 if val > 100:
      3     print(f"{val} is greater than 100.")

ValueError: invalid literal for int() with base 10: 'a'
```

```python
# Checking for a multiple of 10
val = int(input("Enter the value: "))
if val % 10 == 0:
    print(f"{val} is factor of 10")
else:
    print(f"{val} is not a factor of 10")
```

```
Enter the value:  1200
1200 is factor of 10
```

```python
# Checking for a multiple of 10
val = int(input("Enter the value: "))
if val % 10 == 0:
    print(f"{val} is factor of 10")
else:
    print(f"{val} is not a factor of 10")
```

```
Enter the value:  111
111 is not a factor of 10
```

```python
# Checking the type of an input
val = "11"
print(type(val))
if val == 11: # False
    print(f"{val} is a numeric value")
else:
    print("Please fix the data type of the input, as its not a numeric type!")
```

```
<class 'str'>
Please fix the data type of the input, as its not a numeric type!
```

```python
# Checking the type of an input
val = 11.1
if type(val) is float:
    print(f"{val} is a float value")
else:
    print("Please fix the data type of the input, as its not a numeric type!")
```

```
11.1 is a float value
```

```python
# Checking the type of an input
val = "11"
if type(val) is float:
    print(f"{val} is a float value")
else:
    print("Please fix the data type of the input, as its not a float type!")
```

```
Please fix the data type of the input, as its not a float type!
```

```python
val = 10
if type(val) is int:
    print("val is an integer")
```

```
val is an integer
```

```python
val = 3.14
if isinstance(val, (int, float, complex, str)):
    print("val is a number")
else:
    print("Hello World!")
```

```
val is a number
```

```python
val = "11"
if isinstance(val, (int, float, complex)):
    print("val is a number")
else:
    print("Hello World!")
```

```
Hello World!
```

```python
# 'in'/'not in' operator -> membership operator [Returning True or False]
car_list = ['Taigun','Creta','Slavia','Virtus','Thar','ScorpioN','i10','Verna','Jeep','Lord Alto']
fav_car = 'Thar'
if fav_car in car_list:
    print(f"{fav_car} is available in car_list")
else:
    print(f"Sorry! We will add it soon....")
```

```
Thar is available in car_list
```

```python
# 'in'/'not in' operator -> membership operator [Returning True or False]
car_list = ['Taigun','Creta','Slavia','Virtus','Thar','ScorpioN','i10','Verna','Jeep','Lord Alto']
fav_car = 'GWagon'
if fav_car in car_list:
    print(f"{fav_car} is available in car_list")
else:
    print(f"Sorry! We will add it soon....")
```

```
Sorry! We will add it soon....
```

```python
# 'in'/'not in' operator -> membership operator [Returning True or False]
car_list = ['Taigun','Creta','Slavia','Virtus','Thar','ScorpioN','i10','Verna','Jeep','Lord Alto']
fav_car = 'Thar'
if fav_car in car_list:
    print(f"{fav_car} is available in car_list")
else:
    print(f"Sorry! We will add it soon....")
```

```
Thar is available in car_list
```

```python
# 'in'/'not in' operator -> membership operator [Returning True or False]
car_list = ['Taigun','Creta','Slavia','Virtus','Thar','ScorpioN','i10','Verna','Jeep','Lord Alto']
fav_car = 'GWagon'
if fav_car in car_list:
    print(f"{fav_car} is available in car_list")
else:
    print(f"Sorry! We will add it soon....")
```

```
Sorry! We will add it soon....
```

```python
car_list = ['Taigun','Creta','Slavia','Virtus','Thar','ScorpioN','i10','Verna','Jeep','Lord Alto']
fav_car1 = 'GWagon'
fav_car2 = 'Creta'
if fav_car1 in car_list or fav_car2 in car_list: # True
    print("We are checking Nested Conditions : ")
    if fav_car1 in car_list: # False
        print(f"{fav_car1} is available in car_list")
    elif fav_car2 in car_list: # True
        print(f"{fav_car2} is available in car_list")
else:
    print(f"Sorry! We will add it soon....")
```

```
We are checking Nested Conditions :
Creta is available in car_list
```

```python
car_list = ['Taigun','Creta','Slavia','Virtus','Thar','ScorpioN','i10','Verna','Jeep','Lord Alto']
fav_car1 = 'GWagon'
fav_car2 = 'Creta'
if fav_car1 in car_list or fav_car2 in car_list: # True
    print("We are checking Nested Conditions : ")
    if fav_car1 in car_list: # False
        print(f"{fav_car1} is available in car_list")
    else: # True ['Creta']
        print(f"{fav_car2} is available in car_list")
else:
    print(f"Sorry! We will add it soon....")
```

if-else

```
We are checking Nested Conditions :
Creta is available in car_list
```

```python
car_list = ['Taigun','Creta','Slavia','Virtus','Thar','ScorpioN','i10','Verna','Jeep','Lord Alto']
fav_car1 = 'Virtus' # True
fav_car2 = 'Creta' # True
if fav_car1 in car_list or fav_car2 in car_list: # True
    print("We are checking Nested Conditions : ")
    if fav_car1 in car_list and fav_car2 in car_list:
        print(f"{fav_car1} & {fav_car2} is available in car_list")
    elif fav_car1 in car_list :
        print(f"{fav_car1} is available in car_list")
    elif fav_car2 in car_list :
        print(f"{fav_car2} is available in car_list")
else:
    print(f"Sorry! We will add it soon....")
```

```
We are checking Nested Conditions :
Virtus & Creta is available in car_list
```

```python
car_list = ['Taigun','Creta','Slavia','Virtus','Thar','ScorpioN','i10','Verna','Jeep','Lord Alto']
fav_car1 = 'Virtus' # True
fav_car2 = 'Bolero' # False
if fav_car1 in car_list or fav_car2 in car_list: # True
    print("We are checking Nested Conditions : ")
    if fav_car1 in car_list and fav_car2 in car_list:
        print(f"{fav_car1} & {fav_car2} is available in car_list")
    elif fav_car1 in car_list :
        print(f"{fav_car1} is available in car_list")
    elif fav_car2 in car_list :
        print(f"{fav_car2} is available in car_list")
else:
    print(f"Sorry! We will add it soon....")
```

```
We are checking Nested Conditions :
Virtus is available in car_list
```

```python
car_list = ['Taigun','Creta','Slavia','Virtus','Thar','ScorpioN','i10','Verna','Jeep','Lord Alto']
fav_car1 = 'BE6' # False
fav_car2 = 'Verna' # True
if fav_car1 in car_list or fav_car2 in car_list: # True
    print("We are checking Nested Conditions : ")
    if fav_car1 in car_list and fav_car2 in car_list:
        print(f"{fav_car1} & {fav_car2} is available in car_list")
    elif fav_car1 in car_list :
        print(f"{fav_car1} is available in car_list")
    elif fav_car2 in car_list :
        print(f"{fav_car2} is available in car_list")
else:
    print(f"Sorry! We will add it soon....")
```

```
We are checking Nested Conditions :
Verna is available in car_list
```

```python
car_list = ['Taigun','Creta','Slavia','Virtus','Thar','ScorpioN','i10','Verna','Jeep','Lord Alto']
fav_car1 = 'BE6' # False
fav_car2 = 'Harrier' # False
if fav_car1 in car_list or fav_car2 in car_list: # True
    print("We are checking Nested Conditions : ")
    if fav_car1 in car_list and fav_car2 in car_list:
        print(f"{fav_car1} & {fav_car2} is available in car_list")
    elif fav_car1 in car_list :
        print(f"{fav_car1} is available in car_list")
    elif fav_car2 in car_list :
        print(f"{fav_car2} is available in car_list")
else:
    print(f"Sorry! We will add it soon....")
```

```
Sorry! We will add it soon....
```

```python
student_details = {
    'name' : 'Prabhakar Shah',
    'age' : 28,
    'gender' : 'Male',
    'city' : 'Indore',
    'State' : 'Madhya Pradesh',
    'country' : 'India',
    'course' : 'Data Analytics',
    'skills' : ['Excel','Power BI','Sql','Python']
}
key_evaluate = 'course'
if key_evaluate in student_details:
    print("You are searching : " , student_details[key_evaluate])
else:
    print("No such key exist!")
```

```
You are searching :  Data Analytics
```

```
student_details = {
    'name' : 'Prabhakar Shah',
    'age' : 28,
    'gender' : 'Male',
    'city' : 'Indore',
    'State' : 'Madhya Pradesh',
    'country' : 'India',
    'course' : 'Data Analytics',
    'skills' : ['Excel','Power BI','Sql','Python']
}
key_evaluate = 'skills'
if key_evaluate in student_details:
    print("You are searching : " , student_details[key_evaluate])
else:
    print("No such key exist!")
```

```
You are searching :  ['Excel', 'Power BI', 'Sql', 'Python']
```

```
if {}: # False
    print("Hello World")
else:
    print("Above code never run as condition {} is False")
```

```
Above code never run as condition {} is False
```

```
student_details = {
    'name' : 'Prabhakar Shah',
    'age' : 28,
    'gender' : 'Male',
    'city' : 'Indore',
    'State' : 'Madhya Pradesh',
    'country' : 'India',
    'course' : 'Data Analytics',
    'skills' : ['Excel','Power BI','Sql','Python']
}
student_details['city'] = "Gwalior"
```

```
student_details
```

```
{'name': 'Prabhakar Shah',
 'age': 28,
 'gender': 'Male',
 'city': 'Gwalior',
 'State': 'Madhya Pradesh',
 'country': 'India',
 'course': 'Data Analytics',
 'skills': ['Excel', 'Power BI', 'Sql', 'Python']}
```

```
if set(): # False () , [] , ""
    print("Hello World")
else:
    print("Above code never run as condition set() is empty representing False")
```

```
Above code never run as condition set() is empty representing False
```

```
if 0: # False () , [] , ""
    print("Hello World")
else:
    print("Above code never run as condition 0 is  representing False")
```

```
Above code never run as condition 0 is  representing False
```

```python
if -3000: # True
    print("Any number either positive or negative except 0 is True")
else:
    print("Else won't get executed ")
```

```
Any number either positive or negative except 0 is True
```

```python
num = 900
if num == 1000:
    print("num is equal to 1000")
elif num >= 700:
    print("num is greater than 700")
elif num >=500:
    print("num is greater than 500")
else:
    print("num is less than 500")
```

```
num is greater than 700
```

```python
# Find the max marks by comparing
marks1 = int(input("Enter your marks1 : "))
marks2 = int(input("Enter your marks2 : "))
marks3 = int(input("Enter your marks3 : "))

if marks1 > marks2 and marks1 > marks3: # 'marks1' is a winner
    print(f"{marks1} is greater than {marks2} & {marks3}.")
elif marks2 > marks1 and marks2 > marks3: # 'marks2' is a winner
    print(f"{marks2} is greater than {marks1} & {marks3}.")
else: # 'marks3' is a winner
    print(f"{marks3} is greater than {marks1} & {marks2}.")
```

```
Enter your marks1 :  90
Enter your marks2 :  77
Enter your marks3 :  99
99 is greater than 90 & 77.
```

```python
# Find the min marks by comparing
marks1 = int(input("Enter your marks1 : "))
marks2 = int(input("Enter your marks2 : "))
marks3 = int(input("Enter your marks3 : "))

if marks1 < marks2 and marks1 < marks3: # 'marks1' is a winner
    print(f"{marks1} is lesser than {marks2} & {marks3}.")
elif marks2 < marks1 and marks2 < marks3: # 'marks2' is a winner
    print(f"{marks2} is lesser than {marks1} & {marks3}.")
else: # 'marks3' is a winner
    print(f"{marks3} is lesser than {marks1} & {marks2}.")
```

```
Enter your marks1 :  95
Enter your marks2 :  85
Enter your marks3 :  99
85 is lesser than 95 & 99.
```

```
employee_details = {
    'name' : 'Bhupinder Jogi',
    'age' : 28, # [<=30,>30-45,>45-60,>60]
    'gender': 'Male',
    'email' : 'bhupinder@yahoo.com',
    'experience' : 'Fresher',
    'designation' : 'Deputy Manager',
    'salary' : 900000, # [<50K, <1L, >1L]
    'state': 'Florida',
    'country' : 'USA',
    'total children' : 4, #
    'home_owner' : 'Y', #
    'marital_status' : 'Single' # ['Single', 'Married' , 'Divorce']
}
```

```
# We are here to sell a Life Insurance as per the person details
premium_amount = 0
if employee_details['home_owner'] == 'Y':
    if employee_details['marital_status'] == 'Married' and employee_details['total children'] >= 2:
        print("Married and having children greater or equal to 2")
        if employee_details['age'] >30 and employee_details['age'] <=45 and employee_details['salary'] <= 50000:
            # age [>30-45] and salary <=50k
        elif employee_details['age'] >30 and employee_details['age'] <=45 and employee_details['salary'] <= 100000:
            # age [>30-45] and salary <=1L
        elif employee_details['age'] >30 and employee_details['age'] <=45 and employee_details['salary'] > 100000:
            # age [>30-45] and salary >1L
    elif employee_details['marital_status'] == 'Married' and employee_details['total children'] == 1:
        print("Married and having children equal to 1")
    elif employee_details['marital_status'] == 'Married' and employee_details['total children'] == 0:
        print("Married but no children yet")
    elif employee_details['marital_status'] == 'Single':
        if employee_details['age'] <=30 and employee_details['salary'] <= 50000:
            # age <= 30 and salary <=50k
        elif employee_details['age'] <=30 and employee_details['salary'] <= 100000:
            # age <= 30 and salary <= 1L **[Bhupinder Jogi]**
        elif employee_details['age'] <=30 and employee_details['salary'] > 100000:
            # age <= 30 and salary > 1L
        print("Single")
    else:
        print("Divorce")

else : # home_owner == 'N'
    print("He is Living in a Rental House")
    if employee_details['marital_status'] == 'Married':
        print("Married")
    elif employee_details['marital_status'] == 'Single':
        print("Single")
    else:
        print("Divorce")
```

Shorthand Conditional Statements :

if : `if condition: statement`

if-else : `result = value1 if condition else value2`

if-elif-else : `result = (value1 if condition1 else value2) if condition2 else value3`