

## 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

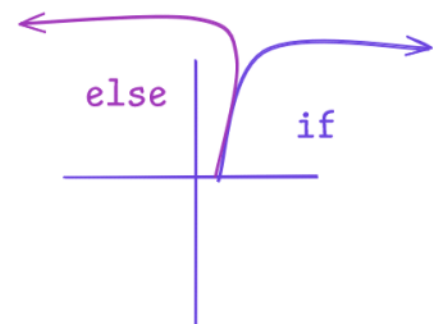
```
If condition:
    # code block
```

#### Syntax : If-else

```
if condition1:
    # Block if condition1 true
else:
    # Block if all conditions are false
```

#### 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
```

### 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`

```
# Conditional Statements :
# 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!
```

```
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!
```

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

print("Outside the if conditions, Always Work!")
# ValueError: invalid literal for int() with base 10: 'a'
```

```
num1 = int(input("Enter the num1: ")) # -99
num2 = int(input("Enter the num2: ")) # 55
if num1 > num2: # Condition [Boolean Return][True/False]
    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 {num2}")
```

```
Enter the num1: -99
Enter the num2: 55
55 is greater than -99
-99 is not equal 55
```

```
# if condition1:
    # Block if condition1 true
# else:
    # Block if condition1 is false
_bool = True
print(_bool)
print(int(_bool)) # 1
```

True

1

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

False

0

```
val = int(input("Enter the value: "))
if val > 100:
    print(f"{val} is greater than 100.")
else: # This block will run if the above the condition is false
    print(f"{val} is less than or equal to 100.")
```

Enter the value: 99

99 is less than or equal to 100.

```
val = int(input("Enter the value: "))
if val > 100:
    print(f"{val} is greater than 100.")
else: # This block will run if the above the condition is false
    print(f"{val} is less than or equal to 100.")
```

Enter the value: 100

100 is less than or equal to 100.

```
val = int(input("Enter the value: "))
if val > 100:
    print(f"{val} is greater than 100.")
else: # This block will run if the above the condition is false
    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: # This block will run if the above the condition is false
    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: # This block will run if the above the condition is false
    print(f"{val} is less than or equal to 100.")
# ValueError: invalid literal for int() with base 10: 'A10'

```

```

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

```

```

Enter the value: 1100
10 is a factor of 1100.

```

```

# Checking for a multiple of 10
val = int(input("Enter the value: "))
remainder = val % 10
if val % 10 == 0:
    print(f" 10 is a factor of {val}.")
else:
    print(f" 10 is not a factor of {val}. You left with {remainder} as remainder")

```

```

Enter the value: 777
10 is not a factor of 777. You left with 7 as remainder

```

```

# Checking the type of an input
val = "11"
print(type(val)) # 'str'
if val == 11: # False
    print(f"{val} is a numeric Value")
else:
    print("Please fix the data type of an input, as it is not a numeric type!")

```

```

<class 'str'>
Please fix the data type of an input, as it is not a numeric type!

```

```

# Checking the type of an input
val = 11.1
print(type(val)) # 'float'
if type(val) is float : # True
    print(f"{val} is a float Value")
else:
    print("Please fix the data type of an input, as it is not a numeric type!")

```

```

<class 'float'>
11.1 is a float Value

```

```

# Checking the type of an input
val = "11"
print(type(val)) # 'str'
if type(val) is float : # False
    print(f"{val} is a float Value")
else:
    print("Please fix the data type of an input, as it is not a float type!")

```

```

<class 'str'>
Please fix the data type of an input, as it is not a float type!

```

```

val = 10
if type(val) is int:
    print(f'{val} as a value is an integer.')
10 as a value is an integer.

val = 3.14
if isinstance(val , (int, float, complex, str)): # True
    print(f'{val} as a value is a number.')
else:
    print("Hello World.")
3.14 as a value is a number.

val = "11"
if isinstance(val , (int, float, complex)): # False
    print(f'{val} as a value is a number.')
else:
    print("Hello World.")
Hello World.

```

```

# 'in' / 'not in' Membership Operators <Iterables> [True/False][Boolean Returns]
car_list = ['Taigun', 'i20', 'Defender', 'Creta', 'Venue', 'Slavia', 'Harrier', 'City', 'Civic', 'Tiago', 'Verna']
fav_car = 'Thar'
if fav_car in car_list: # False
    print(f'{fav_car} is avaialble in car_list')
else:
    print(f'Sorry! We will add it soon to our collections....')
Sorry! We will add it soon to our collections....

# 'in' / 'not in' Membership Operators <Iterables> [True/False][Boolean Returns]
car_list = ['Taigun', 'i20', 'Defender', 'Creta', 'Venue', 'Slavia', 'Harrier', 'City', 'Civic', 'Tiago', 'Verna']
fav_car = 'Defender'
if fav_car in car_list: # True
    print(f'{fav_car} is avaialble in car_list')
else:
    print(f'Sorry! We will add it soon to our collections....')
Defender is avaialble in car_list

```

```

# Logical Operators [or , and , not] # True/False
# Nested Conditions
car_list = ['Taigun', 'i20', 'Defender', 'Creta', 'Venue', 'Slavia', 'Harrier', 'City', 'Civic', 'Tiago', 'Verna']
fav_car1 = 'Creta'
fav_car2 = 'GWagon'
if fav_car1 in car_list or fav_car2 in car_list: # True
    print("We are still Processing to evaluate if both car exist or not")
    if fav_car1 in car_list : # True
        print(f'{fav_car1} is available in stock.")
    if fav_car2 in car_list : # False
        print(f'{fav_car2} is available in stock.")
else:
    print(f'Sorry! We will add it soon to our collections....')
We are still Processing to evaluate if both car exist or not
Creta is available in stock.

```



```
# Logical Operators [or , and , not] # True/False
# Nested Conditions
car_list = ['Taigun','i20','Defender','Creta','Venue','Slavia','Harrier','City','Civic','Tiago','Verna']
fav_car1 = 'GWagon'
fav_car2 = 'Creta'
if fav_car1 in car_list or fav_car2 in car_list: # True
    print("We are still Processing to evaluate if both car exist or not")
    if fav_car1 in car_list : # False
        print(f"{fav_car1} is available in stock.")
    else: # True
        print(f"{fav_car2} is available in stock.")
else:
    print(f'Sorry! We will add it soon to our collections....')
```

We are still Processing to evaluate if both car exist or not  
Creta is available in stock.

```
# Logical Operators [or , and , not] # True/False
# Nested Conditions
car_list = ['Taigun','i20','Defender','Creta','Venue','Slavia','Harrier','City','Civic','Tiago','Verna']
fav_car1 = 'Defender'
fav_car2 = 'Creta'
if fav_car1 in car_list or fav_car2 in car_list: # True
    print("We are still Processing to evaluate if both car exist or not")
    if fav_car1 in car_list and fav_car2 in car_list: # True
        print(f'{fav_car1} & {fav_car2} are available in stock')
    elif fav_car1 in car_list : # True
        print(f"{fav_car1} is available in stock.")
    elif fav_car2 in car_list : # True
        print(f"{fav_car2} is available in stock.")
else:
    print(f'Sorry! We will add it soon to our collections....')
```

We are still Processing to evaluate if both car exist or not  
Defender & Creta are available in stock

```
# Logical Operators [or , and , not] # True/False
# Nested Conditions
car_list = ['Taigun','i20','Defender','Creta','Venue','Slavia','Harrier','City','Civic','Tiago','Verna']
fav_car1 = 'Defender' # True
fav_car2 = 'Creta' # True
if fav_car1 in car_list or fav_car2 in car_list: # True
    print("We are still Processing to evaluate if both car exist or not")
    if fav_car1 in car_list and fav_car2 in car_list: # True
        print(f'{fav_car1} & {fav_car2} are available in stock')
    if fav_car1 in car_list : # True
        print(f"{fav_car1} is available in stock.")
    if fav_car2 in car_list : # True
        print(f"{fav_car2} is available in stock.")
else:
    print(f'Sorry! We will add it soon to our collections....')
```

We are still Processing to evaluate if both car exist or not  
Defender & Creta are available in stock  
Defender is available in stock.  
Creta is available in stock.

```
# Logical Operators [or , and , not] # True/False
# Nested Conditions
car_list = ['Taigun','i20','Defender','Creta','Venue','Slavia','Harrier','City','Civic','Tiago','Verna']
fav_car1 = 'Defender' # True
fav_car2 = 'Bolero' # False
if fav_car1 in car_list or fav_car2 in car_list: # True
    print("We are still Processing to evaluate if both car exist or not")
    if fav_car1 in car_list and fav_car2 in car_list: # False
        print(f'{fav_car1} & {fav_car2} are available in stock')
    elif fav_car1 in car_list : # True
        print(f"{fav_car1} is available in stock.")
    elif fav_car2 in car_list : # False
        print(f"{fav_car2} is available in stock.")
else:
    print(f'Sorry! We will add it soon to our collections....')
```

We are still Processing to evaluate if both car exist or not  
Defender is available in stock.

```
# Logical Operators [or , and , not] # True/False
# Nested Conditions
car_list = ['Taigun','i20','Defender','Creta','Venue','Slavia','Harrier','City','Civic','Tiago','Verna']
fav_car1 = 'Land Rover' # False
fav_car2 = 'Verna' # True
if fav_car1 in car_list or fav_car2 in car_list: # True
    print("We are still Processing to evaluate if both car exist or not")
    if fav_car1 in car_list and fav_car2 in car_list: # False
        print(f'{fav_car1} & {fav_car2} are available in stock')
    elif fav_car1 in car_list : # False
        print(f"{fav_car1} is available in stock.")
    elif fav_car2 in car_list : # True
        print(f"{fav_car2} is available in stock.")
else:
    print(f'Sorry! We will add it soon to our collections....')
```

We are still Processing to evaluate if both car exist or not  
Verna is available in stock.

```
# Logical Operators [or , and , not] # True/False
# Nested Conditions
car_list = ['Taigun','i20','Defender','Creta','Venue','Slavia','Harrier','City','Civic','Tiago','Verna']
fav_car1 = 'Land Rover' # False
fav_car2 = 'ScorpioN' # False
if fav_car1 in car_list or fav_car2 in car_list: # False
    print("We are still Processing to evaluate if both car exist or not")
    if fav_car1 in car_list and fav_car2 in car_list:
        print(f'{fav_car1} & {fav_car2} are available in stock')
    elif fav_car1 in car_list :
        print(f"{fav_car1} is available in stock.")
    elif fav_car2 in car_list :
        print(f"{fav_car2} is available in stock.")
else:
    print(f'Sorry! We will add it soon to our collections....') # True
```

Sorry! We will add it soon to our collections....



```
# Dictionary
student_details = {
    'name' : 'Shubham Singh',
    'age' : 28,
    'gender' : 'Male',
    'city' : 'pune',
    'state' : 'Maharashtra',
    'country' : 'India',
    'course' : 'Data Analytics',
    'skills' : ['Excel', 'PowerBI', 'SQL', 'Python']
}
key_evaluate = input("Enter the key to obtain the result : ")
if key_evaluate in student_details.keys():
    print("You are searching : " , student_details[key_evaluate])
else:
    print("No Such Key Exists....")

Enter the key to obtain the result : course
You are searching : Data Analytics
```

```
student_details.keys()

dict_keys(['name', 'age', 'gender', 'city', 'state', 'country', 'course', 'skills'])

key_evaluate = input("Enter the key to obtain the result : ")
if key_evaluate in student_details.keys():
    print("You are searching : " , student_details[key_evaluate])
else:
    print("No Such Key Exists....")

Enter the key to obtain the result : skills
You are searching : ['Excel', 'PowerBI', 'SQL', 'Python']

key_evaluate = input("Enter the key to obtain the result : ")
if key_evaluate in student_details.keys():
    print("You are searching : " , student_details[key_evaluate])
else:
    print("No Such Key Exists....")

Enter the key to obtain the result : email_address
No Such Key Exists....
```

```
if {}: # False [], {}, set(), (), 0
    print("Hello World")
else:
    print("Above Code would never run as {} is equal to 0 {False}")

Above Code would never run as {} is equal to 0 {False}

if set(): # False [], {}, set(), (), 0
    print("Hello World")
else:
    print("Above Code would never run as empty set() is equal to 0 {False}")

Above Code would never run as empty set() is equal to 0 {False}

if 0: # False [], {}, set(), (), 0
    print("Hello World")
else:
    print("Above Code would never run as 0 is equal to {False}")

Above Code would never run as 0 is equal to {False}
```

```

if -999: # 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

```

if-elif-else ladder
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

```

```

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

```

num = 300
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 less than 500.

```

num = 1001
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

```

num = 500
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 500

```
# 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: 75
Enter your marks3: 82
90 is greater than 75 & 82.
```

```
# 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: 77
Enter your marks2: 88
Enter your marks3: 99
99 is greater than 77 & 88.
```

```
# 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 == marks3:
    print('All marks are Equal')
elif 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: 99
Enter your marks2: 99
Enter your marks3: 99
All marks are Equal
```



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

Enter your marks1: 88
Enter your marks2: 77
Enter your marks3: 99
77 is smaller than 88 & 99.
```

```
# Short-Hand Conditional Statement
# if condition: statement
marks = int(input("Enter a valid marks")) # 0 - 100 [Valid Marks]
if marks >= 90 : print("Excellent")

Enter a valid marks 77

# if condition: statement
marks = int(input("Enter a valid marks")) # 0 - 100 [Valid Marks]
if marks >= 90 : print("Excellent")

Enter a valid marks 97
Excellent

# if-else -> result = value1 if condition else value2
marks = int(input("Enter a valid marks")) # 0 - 100 [Valid Marks]
print("Fail") if marks <= 33 else print("Passed")

Enter a valid marks 29
Fail
```

```
# if-else -> result = value1 if condition else value2
marks = int(input("Enter a valid marks")) # 0 - 100 [Valid Marks]
print("Fail") if marks <= 33 else print("Passed")

Enter a valid marks 57
Passed

# if-else -> result = value1 if condition else value2
marks = int(input("Enter a valid marks")) # 0 - 100 [Valid Marks]
result = 'Not Passed' if marks <= 33 else "Passed"
print(result)

Enter a valid marks 77
Passed

# if-else -> result = value1 if condition else value2
car_list = ['Taigun', 'Creta', 'Slavia', 'Venue', 'Harrier', 'Safari', 'Thar', 'Scorpio', 'Virtus']
fav_car = 'Thar'
result = "Available in Stock" if fav_car in car_list else "Not Available"
print(result)

Available in Stock
```