



# SKILL TRAINING ON TECHNICAL

**NAME: NISHCHAL.R**

**FIRST SEMISTER**

**BRANCH:CSE IN CYBER  
SECURITY**

**USN:1SV24CX016**

To:



Mr. Rajesh Kannan

# PYTHON PROGRAMMING: DAY 1 TO DAY 5

## Day 1: Introduction to Python

### Topics to Cover

- What is Python?
  - Features of Python
  - Installing Python and Running Scripts
  - Basic Syntax and Data Types
  - Variables and Input/Output
- 

### 1. WHAT IS PYTHON?

PYTHON IS A HIGH-LEVEL, INTERPRETED PROGRAMMING LANGUAGE KNOWN FOR ITS SIMPLICITY AND READABILITY. IT IS WIDELY USED IN WEB DEVELOPMENT, DATA SCIENCE, AUTOMATION, AI, AND MORE.

### 2. FEATURES OF PYTHON

- EASY TO LEARN AND READ.
- OPEN-SOURCE AND FREE.
- DYNAMICALLY TYPED (NO NEED TO DECLARE DATA TYPES).
- SUPPORTS OBJECT-ORIENTED, PROCEDURAL, AND FUNCTIONAL PROGRAMMING.
- EXTENSIVE LIBRARIES FOR VARIOUS APPLICATIONS (NUMPY, PANDAS, TENSORFLOW, ETC.).

### 3. INSTALLING PYTHON AND RUNNING SCRIPTS

1. DOWNLOAD PYTHON FROM [PYTHON.ORG](https://www.python.org).
2. INSTALL PYTHON AND ENSURE PIP (PYTHON PACKAGE MANAGER) IS INCLUDED.
3. VERIFY INSTALLATION USING:
4. PYTHON -VERSION
5. RUN A PYTHON SCRIPT:
6. PYTHON SCRIPT.PY

### 4. WRITING AND RUNNING YOUR FIRST PROGRAM

```
PRINT("HELLO, WORLD!")
```

### 5. VARIABLES AND DATA TYPES

```
NAME = "ALICE" # STRING  
A = 10 # INTEGER  
B = 3.14 # FLOAT  
IS_PYTHON_FUN = TRUE # BOOLEAN
```

## 6. TAKING USER INPUT

```
NAME = INPUT("ENTER YOUR NAME: ")  
PRINT("HELLO, " + NAME)
```

---

# Day 2: Control Flow in Python

## TOPICS TO COVER

- CONDITIONAL STATEMENTS
- LOOPS
- ITERATING OVER STRINGS AND LISTS
- BREAK AND CONTINUE STATEMENTS

### 1. CONDITIONAL STATEMENTS

```
AGE = INT(INPUT("ENTER YOUR AGE: "))
```

```
IF AGE >= 18:
```

```
    PRINT("YOU ARE AN ADULT.")
```

```
ELIF AGE > 12:
```

```
    PRINT("YOU ARE A TEENAGER.")
```

```
ELSE:
```

```
    PRINT("YOU ARE A CHILD.")
```

### 2. LOOPS IN PYTHON

#### FOR LOOP

```
FOR I IN RANGE(5):
```

```
    PRINT("ITERATION", I)
```

#### WHILE LOOP

```
COUNT = 0
```

```
WHILE COUNT < 5:
```

```
    PRINT("COUNT IS", COUNT)
```

```
    COUNT += 1
```

### 3. USING BREAK AND CONTINUE

```
FOR NUM IN RANGE(1, 6):
```

```
    IF NUM == 3:
```

```
        CONTINUE # SKIPS NUMBER 3
```

```
    PRINT(NUM)
```

---

# Day 3: Functions and Modules

## Topics to Cover

- Defining Functions**
- Function Parameters and Return Values**
- Importing and Using Modules**

### 1. Defining and Calling Functions

```
def greet(name):  
    print("Hello, " + name)
```

```
greet("Alice")
```

### 2. Function with Return Value

```
def add(a, b):  
    return a + b
```

```
sum_result = add(5, 3)  
print("Sum:", sum_result)
```

### 3. Importing Modules

```
import math  
print(math.sqrt(16)) # Output: 4.0
```

---

# DAY 4: DATA STRUCTURES IN PYTHON

## TOPICS TO COVER

- LISTS AND OPERATIONS**
- DICTIONARIES**
- SETS AND SET OPERATIONS**

### 1. LISTS

```
NUMBERS = [1, 2, 3, 4, 5]
```

```
MIXED = [1, "HELLO", 3.14]
```

```
PRINT(NUMBERS[0]) # OUTPUT: 1
```

```
NUMBERS.append(6)
```

```
NUMBERS.remove(2)
```

```
PRINT(NUMBERS)
```

### 2. DICTIONARIES

```
BOOK = {
```

```
    "TITLE": "PYTHON BASICS",
```

```
    "AUTHOR": "JOHN DOE",
```

```
    "YEAR": 2021
```

```
}
```

```
PRINT(BOOK["TITLE"])
```

```
BOOK["YEAR"] = 2022
```

### 3. SETS AND OPERATIONS

```
SET1 = {1, 2, 3, 4}
```

```
SET2 = {3, 4, 5, 6}
```

```
PRINT("UNION:", SET1 | SET2)
```

```
PRINT("INTERSECTION:", SET1 & SET2)
```

```
PRINT("DIFFERENCE:", SET1 - SET2)
```

# Day 5: Object-Oriented Programming (OOP)

## TOPICS TO COVER

- CLASSES AND OBJECTS
- ENCAPSULATION
- INHERITANCE
- POLYMORPHISM

### 1. DEFINING A CLASS AND CREATING AN OBJECT

- **Class:** A blueprint for creating objects.
- **Object:** An instance of a class.

#### CLASS PERSON:

```
DEF __INIT__(SELF, NAME, AGE):  
    SELF.NAME = NAME  
    SELF.AGE = AGE
```

#### DEF GREET(SELF):

```
PRINT(F"HELLO, MY NAME IS {SELF.NAME} AND I AM {SELF.AGE} YEARS  
OLD.")
```

```
PERSON1 = PERSON("ALICE", 25)
```

```
PERSON1.GREET()
```

### 2. INHERITANCE

- **ALLOWS ONE CLASS TO INHERIT ATTRIBUTES AND METHODS FROM  
ANOTHER.**

#### EXAMPLE: INHERITANCE IN PYTHON

#### CLASS ANIMAL:

```
DEF SPEAK(SELF):  
    PRINT("I AM AN ANIMAL")
```

#### CLASS DOG(ANIMAL):

```
DEF SPEAK(SELF):
```

```
PRINT("I BARK")
```

```
DOG = DOG()
```

```
DOG.SPEAK()
```

### 3. POLYMORPHISM

- ALLOWS DIFFERENT CLASSES TO DEFINE THE SAME METHOD IN DIFFERENT WAYS.

#### EXAMPLE: POLYMORPHISM IN PYTHON

CLASS SHAPE:

```
DEF AREA(SELF):
```

```
PASS
```

CLASS RECTANGLE(SHAPE):

```
DEF __INIT__(SELF, LENGTH, WIDTH):
```

```
SELF.LENGTH = LENGTH
```

```
SELF.WIDTH = WIDTH
```

```
DEF AREA(SELF):
```

```
RETURN SELF.LENGTH * SELF.WIDTH
```

CLASS CIRCLE(SHAPE):

```
DEF __INIT__(SELF, RADIUS):
```

```
SELF.RADIUS = RADIUS
```

```
DEF AREA(SELF):
```

```
RETURN 3.14 * SELF.RADIUS * SELF.RADIUS
```

```
SHAPES = [RECTANGLE(4, 5), CIRCLE(3)]
```

FOR SHAPE IN SHAPES:

```
PRINT("AREA:", SHAPE.AREA())
```



NISHCHAL DAY 1.ipynb



File Edit View Insert Runtime Tools Help

Commands

+ Code + Text

[1] print("Welcome to Python Programming!")  
0s

→ Welcome to Python Programming!

[2] num = int(input("Enter a number: "))  
print(num > 0)  
2s→ Enter a number: 12  
True[3] text = input("Enter a string: ")  
print("Length of the string:", len(text))  
2s→ Enter a string: 45  
Length of the string: 2[6] value = input("Enter something: ")  
print("The data type of the entered value is:", type(value))  
2s→ Enter something: 45  
The data type of the entered value is: <class 'str'>



NISHCHAL DAY 1.ipynb



File Edit View Insert Runtime Tools Help

Commands

+ Code

+ Text



✓ 8s [4] num1 = float(input("Enter first number: "))  
num2 = float(input("Enter second number: "))  
num3 = float(input("Enter third number: "))  
  
average = (num1 + num2 + num3) / 3  
print("The average is:", average)



✓ 2s [5] num = int(input("Enter a number: "))  
for i in range(1, 11):  
 print(num, "x", i, "=", num \* i)

✓ 2s [5] Enter a number: 12  
12 x 1 = 12  
12 x 2 = 24  
12 x 3 = 36  
12 x 4 = 48  
12 x 5 = 60  
12 x 6 = 72  
12 x 7 = 84  
12 x 8 = 96  
12 x 9 = 108  
12 x 10 = 120



NISHCHAL DAY 1.ipynb



File Edit View Insert Runtime Tools Help

Commands

+ Code

+ Text



```
→ Enter something: 45
→ The data type of the entered value is: <class 'str'>
```



```
[7] num1 = float(input("Enter first number: "))
    num2 = float(input("Enter second number: "))
    operator = input("Enter operator (+, -, *, /): ")

    if operator == "+":
        print("Result:", num1 + num2)
    elif operator == "-":
        print("Result:", num1 - num2)
    elif operator == "*":
        print("Result:", num1 * num2)
    elif operator == "/":
        if num2 != 0:
            print("Result:", num1 / num2)
        else:
            print("Error! Division by zero.")
    else:
        print("Invalid operator!")
```

```
→ Enter first number: 45
Enter second number: 54
Enter operator (+, -, *, /): 12
Invalid operator!
```



```
text = input("Enter a string: ")
print("Reversed string:", text[::-1])
```



```
→ Enter a string: 15
Reversed string: 51
```

day 2 task nishu.ipynb

```
num = int(input("Enter a number: "))
if num % 2 == 0:
    print(f"{num} is an even number.")
else:
    print(f"{num} is an odd number.")

Enter a number: 5
5 is an odd number.
```

day 2 task nishu.ipynb

```
marks = int(input("Enter your marks: "))

if marks >= 90:
    grade = "A+"
elif marks >= 80:
    grade = "A"
elif marks >= 70:
    grade = "B"
elif marks >= 60:
    grade = "C"
elif marks >= 50:
    grade = "D"
else:
    grade = "F"

print(f"Your grade is: {grade}")

Enter your marks: 95
Your grade is: A+
```

day 2 task nishu.ipynb

```
num = float(input("Enter a number: "))

if num > 0:
    print("The number is positive.")
elif num < 0:
    print("The number is negative.")
else:
    print("The number is zero.")

Enter a number: 5
The number is positive.
```

day 2 task nishu.ipynb

```
a = float(input("Enter first number: "))
b = float(input("Enter second number: "))
c = float(input("Enter third number: "))

if a >= b and a >= c:
    max_num = a
elif b >= a and b >= c:
    max_num = b
else:
    max_num = c

print(f"The maximum number is: {max_num}")
```

Enter first number: 5  
Enter second number: 6  
Enter third number: 5  
The maximum number is: 6.0

Release notes X

Please follow our [blog](#) to see more information about new features, tips and tricks, and featured notebooks such as [Analyzing a Bank Failure with Colab](#).

2025-01-13

- Released version 1.2.0 of the ([Open in Colab Chrome Extension](#)).
- Released minimizable comments with indicators next to cell.
- TPU vSe1 Runtimes are now available for selection ([tweet](#)).
- GPU prices were decreased ([tweet](#)).

Python package upgrades

- accelerate 1.1.1 -> 1.2.1
- aiohttp 3.10.10 -> 3.11.11
- altair 4.2.2 -> 5.1.0
- bigframe 1.23.0 -> 1.29.0
- cmake 3.20.5 -> 3.31.2
- covpy 1.5.3 -> 2.6.0
- fastlane-api 1.20 -> 1.43
- folium 0.18.0 -> 0.19.3
- holidays 0.60 -> 0.63
- huggingface-hub 0.26.2 -> 0.27.0
- jsonpickle 3.4.2 -> 4.0.1
- kagglehub 0.3.3 -> 0.3.6
- keras 3.4.1 -> 3.5.0
- matplotlib 3.8.0 -> 3.10.0
- openai 1.54.3 -> 1.57.4
- pymc 5.18.0 -> 5.19.1
- safetensors 0.4.5 -> 0.5.0
- scikit-image 0.24.0 -> 0.25.0
- scikit-learn 1.5.2 -> 1.6.0
- sentence-transformers 3.2.1 -> 3.3.1
- tensorflow 2.17.0 -> 2.17.1
- torch 2.5.0 -> 2.5.1

day 2 task nishu.ipynb

```
year = int(input("Enter a year: "))

if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
    print(f"{year} is a leap year.")
else:
    print(f"{year} is not a leap year.")
```

Enter a year: 2003  
2003 is not a leap year.

day 2 task nishu.ipynb

```
n = int(input("Enter a positive integer: "))

sum_n = n * (n + 1) // 2

print(f"The sum of the first {n} natural numbers is: {sum_n}")
```

Enter a positive integer: 25  
The sum of the first 25 natural numbers is: 325

Start coding or [generate](#) with AI.

CO day 2 task nishu.ipynb ☆

File Edit View Insert Runtime Tools Help

Commands + Code + Text

```
6s  num = int(input("Enter a number: "))

{x}   for i in range(1, 11):
      print(f"{num} x {i} = {num * i}")

Enter a number: 55
55 x 1 = 55
55 x 2 = 110
55 x 3 = 165
55 x 4 = 220
55 x 5 = 275
55 x 6 = 330
55 x 7 = 385
55 x 8 = 440
55 x 9 = 495
55 x 10 = 550
```

CO day 2 task nishu.ipynb ☆ Saving...

File Edit View Insert Runtime Tools Help

Commands + Code + Text

```
18s  num = int(input("Enter a number: "))

{x}   factorial = 1
      for i in range(1, num + 1):
          factorial *= i

      print(f"The factorial of {num} is: {factorial}")

Enter a number: 6
The factorial of 6 is: 720
```

CO day 2 task nishu.ipynb ☆

File Edit View Insert Runtime Tools Help

Commands + Code + Text

```
6s  n = int(input("Enter the number of terms: "))

{x}   a, b = 0, 1
      print("Fibonacci Series:", a, b, end=" ")

      for _ in range(n - 2):
          c = a + b
          print(c, end=" ")
          a, b = b, c

Enter the number of terms: 26
Fibonacci Series: 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765
```

CO day 2 task nishu.ipynb ★ Saving...

File Edit View Insert Runtime Tools Help

Commands + Code + Text

```
[10] num = int(input("Enter a number: "))

{x} if num > 1:
    for i in range(2, int(num ** 0.5) + 1):
        if num % i == 0:
            print(f"{num} is not a prime number.")
            break
        else:
            print(f"{num} is a prime number.")
    else:
        print(f"{num} is not a prime number.")

→ Enter a number: 22
22 is not a prime number.
```

CO day 2 task nishu.ipynb ★

File Edit View Insert Runtime Tools Help

Commands + Code + Text

```
2s num = input("Enter a number: ")

{x} if num == num[::-1]:
    print(f"{num} is a palindrome.")
else:
    print(f"{num} is not a palindrome.")

→ Enter a number: 12
12 is not a palindrome.
```

CO day 2 task nishu.ipynb ★

File Edit View Insert Runtime Tools Help

Commands + Code + Text

Generate create a dataframe with 2 columns and 10 rows Close

```
[12] num = int(input("Enter a number: "))

{x} rev_num = int(str(num)[::-1])
print(f"Reversed number: {rev_num}")

→ Enter a number: 22
Reversed number: 22
```

Generate create a dataframe with 2 columns and 10 rows Close

```
7s start = int(input("Enter the start of the range: "))
end = int(input("Enter the end of the range: "))

sum_even = sum(i for i in range(start, end + 1) if i % 2 == 0)

print(f"The sum of even numbers in the range {start} to {end} is: {sum_even}")

→ Enter the start of the range: 12
Enter the end of the range: 23
The sum of even numbers in the range 12 to 23 is: 102
```

The screenshot shows a Jupyter Notebook interface with the following details:

- File Menu:** File, Edit, View, Insert, Runtime, Tools, Help.
- Toolbar:** Commands, Code, Text.
- Cell List:** [14] is selected.
- Code Cell 14:**

```
num = int(input("Enter a number: "))
digit_count = len(str(abs(num)))
print(f"The number of digits in {num} is: {digit_count}")
```

Output: Enter a number: 12  
The number of digits in 12 is: 2
- Code Cell 35:**

```
num = int(input("Enter a number: "))

# calculate sum of digits
sum_digits = sum(int(digit) for digit in str(num))

# Check if Armstrong number
num_digits = len(str(num))
armstrong_sum = sum(int(digit) ** num_digits for digit in str(num))

print(f"Sum of digits of {num} is: {sum_digits}")

if num == armstrong_sum:
    print(f"{num} is an Armstrong number.")
else:
    print(f"{num} is not an Armstrong number.")
```

Output: Enter a number: 13  
Sum of digits of 13 is: 4  
13 is not an Armstrong number.

The screenshot shows the PyCharm IDE interface with the following details:

- Top Bar:** Shows the title "Day 2 task mishra.ipynb" and standard menu options: File, Edit, View, Insert, Runtime, Tools, Help.
- Sidebar Icons:** Includes icons for file operations (New, Open, Save, Find, etc.), a search bar, and a code editor tab.
- Code Editor:** Displays Python code for generating a triangle of asterisks and another script for a break statement.
- Output Area:** Shows the execution results for each script, including the input prompt, the generated output, and the final printed value.

```
[18] rows = int(input("Enter the number of rows: "))
58
for i in range(1, rows + 1):
    print(" " * (rows - i) + "* " * i)

→ Enter the number of rows: 12
      *
      **
      ***
      ****
      *****
      ******
      *****
      ****
      ***
      **
      *

0s
for i in range(1, 11):
    if i == 5:
        print("Breaking the loop at", i)
        break
    print(i)

→ 1
2
3
4
Breaking the loop at 5
```



```
File Edit View Insert Runtime Tools Help
Q Commands + Code + Text

[2] def greet():
    print("Hello, python!")
{x}
greet()

Hello, python!

Generate a slider using jupyter widgets
[4] def sum_of_two(a, b):
    return a + b

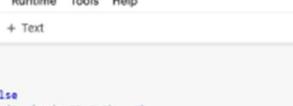
print("Sum:", sum_of_two(5, 3))

Sum: 8

[5] def factorial(n):
    return 1 if n == 0 else n * factorial(n - 1)

print("Factorial:", factorial(5))

Factorial: 120
```



```
File Edit View Insert Runtime Tools Help
Q Commands + Code + Text

[7] def is_prime(n):
    if n < 2:
        return False
    for i in range(2, int(n ** 0.5) + 1):
        if n % i == 0:
            return False
    return True

num = int(input("Enter a number: "))
print("Prime" if is_prime(num) else "Not Prime")

Enter a number: 26
Not Prime

[11] #fibonacci series generator
def fibonacci(n):
    a, b = 0, 1
    for _ in range(n):
        print(a, end=" ")
        a, b = b, a + b

num = int(input("Enter the number of terms: "))
fibonacci(num)

Enter the number of terms: 46
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946 17711 28657 46368 75025 121393 196418
```



```
[13] #Pallindrome checker
def is_palindrome(s):
    return s == s[::-1]

word = input("Enter a word: ")
print("Pallindrome" if is_palindrome(word) else "Not Pallindrome")

Enter a word: 23
Not Pallindrome
```

File of contents

Section

Commands + Code + Text

```
[8] # Sum of digits function
def sum_of_digits(n):
    return sum(int(digit) for digit in str(n))

print(sum_of_digits(1234))
```

10

Generate create a dataframe with 2 columns and 10 rows

```
[9] # Default arguments function
def greet(name="Guest"):
    return f"Hello, {name}!"

print(greet())
print(greet("Alice"))
```

Hello, Guest!  
Hello, Alice!

```
[10] # Using *args in function
def sum_all(*args):
    return sum(args)

print(sum_all(1, 2, 3, 4, 5))
```

15

Nishchal/Zombie Survival: Lab1.ipynb · Colab · Even Odd Checker Code · New tab

File of contents

Section

Commands + Code + Text

```
[11] # Using **kwargs in function
def user_info(**kwargs):
    return kwargs

print(user_info(name="Alice", age=25, city="New York"))
```

{'name': 'Alice', 'age': 25, 'city': 'New York'}

```
[12] # Using the math module
import math
print(math.sqrt(25))
print(math.factorial(5))
```

5.0  
120

```
[13] # Using the random module
import random
print(random.randint(1, 100))
print(random.choice(["Apple", "Banana", "Cherry"]))
```

37  
Apple

```
[14] # Lambda function for squaring
square = lambda x: x ** 2
print(square(6))
```

36

Nishchal/Zombie Survival: Lab | day 3 tasks nishu.ipynb - Colab | Even Odd Checker Code | New Tab

File of contents | Commands | + Code | + Text

GCD: 1

```
[ ] #LCM function
def lcm(a, b):
    return (a * b) // gcd(a, b)

def gcd(a, b):
    while b:
        a, b = b, a % b
    return a

num1 = int(input("Enter first number: "))
num2 = int(input("Enter second number: "))

print("LCM:", lcm(num1, num2))
```

Enter first number: 23  
Enter second number: 2  
LCM: 46

```
[7] # Counting vowels in a string
def count_vowels(s):
    return sum(1 for char in s.lower() if char in "aeiou")

print(count_vowels("Hello World"))
```

3

File Edit View Insert Runtime Tools Help

Table of contents | Commands | + Code | + Text

Results: Invalid Operator

```
[ ] #even or odd function
def even_or_odd(n):
    return "Even" if n % 2 == 0 else "Odd"

num = int(input("Enter a number: "))
print(even_or_odd(num))
```

Enter a number: 85  
Odd

```
[ ] #find GCD using function
def gcd(a, b):
    while b:
        a, b = b, a % b
    return a

num1 = int(input("Enter first number: "))
num2 = int(input("Enter second number: "))

print("GCD:", gcd(num1, num2))
```

Enter first number: 21  
Enter second number: 64  
GCD: 1

day4.ipynb Unsaved changes since 2:11PM

File Edit View Insert Runtime Tools Help

Commands + Code + Text

```
[1] #list creation
nishchal = ("Nishchal", 20, 30, 40, 50)
print(nishchal[0])

{x} ➜ Nishchal

[2] #list indexing
temp_list = list(nishchal)
temp_list.remove("Nishchal")
nishchal = tuple(temp_list)
print(nishchal)

➜ (20, 30, 40, 50)

[3] #List Append and Insert:
temp_list = list(nishchal)
temp_list.append(60)
nishchal = tuple(temp_list)
print(nishchal)

➜ (20, 30, 40, 50, 60)

[4] #list removal
temp_list = list(nishchal)
temp_list.pop(1)
nishchal = tuple(temp_list)
print(nishchal)

➜ (20, 40, 50, 60)

[5] #list sorting
temp_list = list(nishchal)
temp_list.sort()
nishchal = tuple(temp_list)
print(nishchal)

➜ (20, 40, 50, 60)
```

colab.research.google.com/gist/2006nishu/3a3f81f0898d6b1c02dbc74d1f4537f4/data-structures.ipynb#scrollTo=MA7BaR565CZA

day4.ipynb Unsaved changes since 2:11PM

File Edit View Insert Runtime Tools Help

Commands + Code + Text

```
[6] #List Append and Insert:
temp_list = list(nishchal)
temp_list.reverse()
nishchal = tuple(temp_list)
print(nishchal)

{x} ➜ (60, 50, 40, 20)

[7] #List Removal
temp_list = list(nishchal)
temp_list.insert(2, 15)
nishchal = tuple(temp_list)
print(nishchal)

➜ (60, 50, 15, 40, 20)

[8] #List Sorting
temp_list = list(nishchal)
temp_list.sort()
nishchal = tuple(temp_list)
print(nishchal)

➜ (15, 20, 40, 50, 60)

[9] #List Reversal:
temp_list = list(nishchal)
temp_list.reverse()
nishchal = tuple(temp_list)
print(nishchal)

➜ (60, 50, 40, 20, 15)

[10] #Sum of List Elements:
temp_list = list(nishchal)
total = sum(temp_list)
print(total)

<> ➜ 185
```

day4.ipynb Unsaved changes since 2:11PM

File Edit View Insert Runtime Tools Help

Commands + Code + Text

Show command palette (Ctrl+Shift+P)

```
[14] #Maximum and Minimum in a List
    max_value = max(temp_list)
    min_value = min(temp_list)
    print(max_value)
    print(min_value)

→ 60
15

[15] #Count Occurrences
    count = temp_list.count(20)
    print(count)

→ 1

[16] #Merging Lists:
    list1 = [1, 2, 3]
    list2 = [4, 5, 6]
    merged_list = list1 + list2
    print(merged_list)

→ [1, 2, 3, 4, 5, 6]

[17] #Tuple Creation and Access
    nishchal_tuple = ("Nishchal", 20, 30)
    print(nishchal_tuple)

→ ('Nishchal', 20, 30)

[18] #Tuple Unpacking
    name = nishchal_tuple[0]
    age = nishchal_tuple[1]
    print(name)
    print(age)

→ Nishchal
20
```

```
Nishchal
20

[19] #Convert Tuple to List and Vice Versa
    temp_list = list(nishchal_tuple)
    temp_list.append(40)
    nishchal_tuple = tuple(temp_list)
    print(nishchal_tuple)

→ ('Nishchal', 20, 30, 40)

[20] #Tuple Concatenation
    tuple1 = (1, 2, 3)
    tuple2 = (4, 5, 6)
    concatenated_tuple = tuple1 + tuple2
    print(concatenated_tuple)

→ (1, 2, 3, 4, 5, 6)

[ ] ; "Nishchal"

[22] #Dictionary Creation
    nishchal_dict = {"name": "Nishchal", "age": 25}
    print(nishchal_dict)

→ {'name': 'Nishchal', 'age': 25}

[24] #Accessing Dictionary Values
    name = nishchal_dict["name"]
    age = nishchal_dict["age"]
    print(name)
    print(age)

→ Nishchal
25
```

```
2006nishu/Nishchal x | Even Odd Checker Code x | day4.ipynb - Colab x | Task4 x |  
colab.research.google.com/gist/2006nishu/3a3f81f0898d6b1c02dbc74d1f4537f4/data-structures.ipynb#scrollTo=MA7BaR565CZA  
25  
▶ #Adding and Updating Dictionary Elements  
nishchal_dict["city"] = "New York"  
nishchal_dict["age"] = 26  
print(nishchal_dict)  
  
→ {'name': 'Nishchal', 'age': 26, 'city': 'New York'}  
  
[26] #Removing Elements from a Dictionary  
del nishchal_dict["city"]  
print(nishchal_dict)  
  
→ {'name': 'Nishchal', 'age': 26}  
  
[29] #Iterating Through a Dictionary  
for key, value in nishchal_dict.items():  
    print(key, value)  
  
→ name Nishchal  
age 26  
  
[30] #Merging Dictionaries  
dict1 = {"a": 1, "b": 2}  
dict2 = {"b": 3, "c": 4}  
merged_dict = {**dict1, **dict2}  
print(merged_dict)  
  
→ {'a': 1, 'b': 3, 'c': 4}  
  
[31] #Check Key in Dictionary  
if "name" in nishchal_dict:  
    print("Key exists")  
  
→ Key exists
```

```
Key exists  
  
[33] #Set Creation and Operations  
nishchal_set = {1,3,6,6, 20, 30}  
print(nishchal_set)  
  
→ {1, 3, 20, 6, 30}  
  
[34] #Add and Remove Elements from Set:  
nishchal_set.add(40)  
nishchal_set.remove(1)  
print(nishchal_set)  
  
→ {3, 20, 6, 40, 30}  
  
[35] #Check Element in Set  
if 20 in nishchal_set:  
    print("Element exists")  
  
→ Element exists  
  
[36] #Find Common Elements in Two Sets:  
set1 = {1, 2, 3}  
set2 = {3, 4, 5}  
common_elements = set1.intersection(set2)  
print(common_elements)  
  
→ {3}  
  
[38] #Convert List to Set and Remove Duplicates:  
my_list = [1, 2, 3, 3, 4, 5]  
unique_set = set(my_list)  
print(unique_set)  
  
→ {1, 2, 3, 4, 5}
```

The screenshot shows a Google Colab notebook titled "day4.ipynb - Colab". The notebook contains several code cells with their execution results:

- [39] #.List Comprehension:

```
squares = [x**2 for x in range(1, 6)]  
print(squares)
```

Output: [1, 4, 9, 16, 25]
- [40] #.Dictionary Comprehension:

```
square_dict = {x: x**2 for x in range(1, 6)}  
print(square_dict)
```

Output: {1: 1, 2: 4, 3: 9, 4: 16, 5: 25}
- [41] #.Set Comprehension:

```
even_set = {x for x in range(1, 11) if x % 2 == 0}  
print(even_set)
```

Output: {2, 4, 6, 8, 10}
- [42] #Nested Data Structures:

```
nested_list = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]  
print(nested_list)
```

Output: [[1, 2, 3], [4, 5, 6], [7, 8, 9]]



Nishchalday 5.ipynb



File Edit View Insert Runtime Tools Help

Commands

+ Code

+ Text



```
class Car:  
    def set_details(self, brand, model):  
        self.brand = brand # Attribute  
        self.model = model # Attribute  
  
    def display_info(self):  
        print(f"Car: {self.brand} {self.model}")  
  
my_car = Car()  
my_car.set_details("Tesla", "Model S")  
my_car.display_info()
```

→ Car: Tesla Model S

```
[ ] #area of circle  
class circle:  
    def __init__(self, radius):  
        self.radius = radius  
    def area(self):  
        return 3.14 * self.radius * self.radius
```



Nishchalday 5.ipynb



File Edit View Insert Runtime Tools Help

Commands

+ Code

+ Text



[ ]



[ ] import math



```
class Circle:  
    def __init__(self, radius):  
        self.radius = radius
```



```
    def calculate_area(self):  
        return math.pi * self.radius ** 2
```



```
    def display_area(self):  
        print(f"The area of the circle with radius {self.radius} is {self.calculate_area():.2f}")
```

```
circle1 = Circle(5)  
circle1.display_area()
```

→ The area of the circle with radius 5 is 78.54

```
[ ] import math

def calculate_area(radius):
    return math.pi * radius ** 2

def display_area(radius):
    area = calculate_area(radius)
    print(f"The area of the circle with radius {radius} is {area:.2f}")

display_area(5)
```

→ The area of the circle with radius 5 is 78.54

```
[ ] #encapsulation
class nishchal:
    def __init__(self):
        self.public="hello welcome to vik31"
        self.__private="the creator of the VIK31 is nishchal"
    def nishu_private(self):
        print(self.__private)
Nishchal1=nishchal()
print(Nishchal1.public)
Nishchal1.nishu_private()
```

→ hello welcome to vik31  
the creator of the VIK31 is nishchal

```
[ ] #multiple inheritance
class a:
    def features1(self):
        print("This is feature 1")
class b:
    def features2(self):
        print("This is feature 2")
class c(a,b): # Changed 'A' to 'a'
    def features5(self):
        print("This is feature 5")
        pass
features1 = a() # Changed 'features()' to 'a()'
features1.features1()
```

→ This is feature 1

## features1.features1()

→ This is feature 1

```
[ ] #class bank account
class bank:
    def __init__(self,balance=0):
        self.balance=balance
    def invest(self,amount):
        self.balance+=amount
        print(f"your balance is {self.balance}")
    def withdraw(self,amount):
        if amount<=self.balance:
            self.balance-=amount
            print(f"your balance is {self.balance}")
        else:
            print("insufficient balance")
obj1=bank()
obj1.invest(1000)
obj1.withdraw(500)
```

→ your balance is 1000  
your balance is 500

```
[ ] from os import name
class vehicle:
    def __init__(self,name,max_speed,avearage):
        self.name=name
        self.max_speed=max_speed
        self.average=avearage
class vehicle1:

    def display_info(self):
        print(f"Vehicle Name: {self.name}")
        print(f"Maximum Speed: {self.max_speed}")
        print(f"Average: {self.average}")
from os import name
class vehicle:
    def __init__(self,name,max_speed,avearage):
        self.name=name
        self.max_speed=max_speed
        self.average=avearage
class vehicle1:

    def display_info(self):
        print(f"Vehicle Name: {self.name}")
        print(f"Maximum Speed: {self.max_speed}")
        print(f"Average: {self.average}")
class c(vehicle,vehicle1):
    def __init__(self,name,max_speed,avearage):
        super().__init__(name,max_speed,avearage)

obj1=c("car",200,10)
obj1.display_info()
obj1.display_info()
```

→ Vehicle Name: car  
Maximum Speed: 200  
Average: 10  
Vehicle Name: car  
Maximum Speed: 200  
Average: 10

```
[ ] #getters and setters
class MyClass:
    def __init__(self):
        self._value=ValueError
    def get_value(self):
        return self._value
    def set_value(self,value): # Fixed: Changed 'set_values' to 'set_value'
        self._value=value
obj1=MyClass()
obj1.set_value(10) # Fixed: Changed 'set_values' to 'set_value'
print(obj1.get_value())
```

→ 10

```
▶ # Parent class
class Animal:
    def __init__(self, name):
        self.name = name

    def make_sound(self):
        print("Some generic animal sound")

class Dog(Animal):
    def make_sound(self):
        print(f"{self.name} barks: Woof! Woof!")

    def fetch(self):
        print(f"{self.name} is fetching the ball!")

generic_animal = Animal("Some Animal")
dog = Dog("Buddy")

generic_animal.make_sound()
dog.make_sound()
dog.fetch() # Calls Dog-specific method
```

→ Some generic animal sound  
 Buddy barks: Woof! Woof!  
 Buddy is fetching the ball!

---

 Nishchalday 5.ipynb    
 File Edit View Insert Runtime Tools Help

Commands | + Code + Text

☰	Double-click (or enter) to edit
☒	Double-click (or enter) to edit
<>	[ ] # Parent class
{x}	class Vehicle: def start(self): print("The vehicle is starting...")
☞	class Car(Vehicle): def start(self): print("The car engine is starting with a roar!")
☰	v = Vehicle() c = Car()  v.start() c.start()

→ The vehicle is starting...  
 The car engine is starting with a roar!

```
[ ] #demonstrate polymorphism
class shape:
    def area(self):
        pass
class circle(shape):
    def __init__(self, radius):
        self.radius = radius
    def area(self):
        return 3.14 * self.radius * self.radius
class circle(shape):
    def __init__(self, radius):
        self.radius = radius
    def area(self):
        return 3.14 * self.radius * self.radius
circle1 = circle(5)
print(circle1.area())

obj1 = circle(5)
print(obj1.area())
```

→ 78.5  
78.5