TASK 1 : PYTHON BASICS & DATA TYPES

In [2]:
```
"""Declare variables of different data types
(int, float, string, boolean) """

integer_var = 10
float_var = 25.5
string_var = "Data Analytics"
boolean_var = True
```

In [ ]:
```
""" Take user input for name, age, and marks"""

name = input("Enter Your Name")
age = input("Enter Your Age")
marks = input("Enter Your Marks")
```

In [4]:
```
"""Display the entered values using print()"""

print(f"Integer: {integer_var}, Type: {type(integer_var)}")
print(f"float: {float_var}, Type: {type(float_var)}")
print(f"String: {string_var}, Type: {type(string_var)}")
print(f"Boolean: {boolean_var}, Type: {type(boolean_var)}")
```

```
Integer: 10, Type: <class 'int'>
float: 25.5, Type: <class 'float'>
String: Data Analytics, Type: <class 'str'>
Boolean: True, Type: <class 'bool'>
```

In [5]:
```
"""Perform basic arithmetic operations"""

a = 10
b = 20
Sum = a+b
Substraction = b-a
Division = b/a
Multiplication = a*b
Floor_Division = b//a
Modulous = a%b
Exponention = a**2
print(f"Sum of {a} and {b} are",Sum)
print(f"Substraction between of {a} and {b} are",Substraction)
print(f"Division between of {b} and {a} are",Division)
print(f"Multiplication between of {a} and {b} are",Multiplication)
print(f"Floor_Division between of {b} and {a} are",Floor_Division)
print(f"Modulous between of {a} and {b} are",Modulous)
print(f"Exponention between of {a} and {b} are",Exponention)
```

```
Sum of 10 and 20 are 30
Substraction between of 10 and 20 are 10
Division between of 20 and 10 are 2.0
```

```
Multiplication between of 10 and 20 are 200
Floor_Division between of 20 and 10 are 2
Modulous between of 10 and 20 are 10
Exponention between of 10 and 20 are 100
```

TASK 2 : DATA STRUCTURES & LOOPS

In [7]:
```
""" Create a list of at least 10 numbers """

numbers_list = [15, 8, 22, 5, 30, 12, 45, 18, 9, 7]
```

In [8]:
```
"""Find:
o Sum of numbers
o Maximum and minimum value """

Sum = sum(numbers_list)
Max = max(numbers_list)
Min = min(numbers_list)
print(f"The Sum of list is",Sum)
print(f"The Max of list is",Max)
print(f"The Min of list is",Min)
```

```
The Sum of list is 171
The Max of list is 45
The Min of list is 5
```

In [9]:
```
"""Create a dictionary with student details
(name, age, course, marks) """

student_details = {
    'name': 'Anu Mole',
    'age': 24,
    'gender':'Female',
    'Department': 'Commerce',
    'marks': 56
}
```

In [10]:
```
""" Use a loop to display all dictionary values """

for key, value in student_details.items():
    print(f"{key}: {value}")
```

```
name: Anu Mole
age: 24
gender: Female
Department: Commerce
marks: 56
```

TASK 3 : FUNCTIONS & LAMBDA

In [12]:
```
"""Create a function to calculate:
o Total marks
o Average marks """
```

```python
marks = [85, 90, 78, 95]

def calculate_marks(marks_list):
    if not marks_list:
        return 0, 0.0

    total_marks = sum(marks_list)
    average_marks = total_marks / len(marks_list)

    return total_marks, average_marks

total, average = calculate_marks(marks)

print(f"Total Marks: {total}")
print(f"Average Marks: {average:.2f}")
```

```
Total Marks: 348
Average Marks: 87.00
```

In [13]:

```python
""" Create a function to check whether a number is even or odd """

def even_odd(number):
    if number % 2 == 0:
        return "Even"
    else:
        return "Odd"

test_number = int(input("Enter the number"))
print(f"The number {test_number} is: {even_odd(test_number)}")
```

```
Enter the number 99

The number 99 is: Odd
```

In [14]:

```python
""" Write a lambda function to calculate the square of a number """

square = lambda x: x * x

num_to_square = int(input("Enter the number"))
print(f"The square of {num_to_square} is: {square(num_to_square)}")
```

```
Enter the number 25

The square of 25 is: 625
```

In [15]:

```python
""" Function that returns only even numbers from a list"""

def only_even_numbers(numbers):
    even_numbers = []
    for n in numbers:
        if n % 2 == 0:
            even_numbers.append(n)
    return even_numbers

List = [10, 20,25,55,36,84,94]
```

```
print("Given List:", List)
print("Even Numbers:", only_even_numbers(List))
```

```
Given List: [10, 20, 25, 55, 36, 84, 94]
Even Numbers: [10, 20, 36, 84, 94]
```

TASK 4 : FILE HANDLING

In [17]:
```
""" Create a text file named sample.txt  """

file_name = "Data Analytics.txt"
```

In [18]:
```
""" Write at least 5 lines of text into the file """

try:
    with open(file_name, 'w') as file:
        file.write("Line 1: This is the first line of text.\n")
        file.write("Line 2: Data Analytics is exciting.\n")
        file.write("Line 3: Python is a key tool in DA.\n")
        file.write("Line 4: This is the fourth line.\n")
        file.write("Line 5: The file is created and written to.\n")
except Exception as e:
    print(f"An error occurred during writing: {e}")
```

In [19]:
```
""" Read and display the file content """

try:
    with open(file_name, 'r') as file:
        content = file.read()
        print(content)
except FileNotFoundError:
    print(f"Error: The file '{file_name}' was not found.")
except Exception as e:
    print(f"An error occurred during reading: {e}")
```

```
Line 1: This is the first line of text.
Line 2: Data Analytics is exciting.
Line 3: Python is a key tool in DA.
Line 4: This is the fourth line.
Line 5: The file is created and written to.
```

In [20]:
```
""" Append new content to the same file """

try:
    with open(file_name, 'a') as file:
        file.write("Line 6: Appending a new line now.\n")
        file.write("Line 7: This is the final appended line.\n")
    print(f"Successfully appended new content to '{file_name}'.")
except Exception as e:
    print(f"An error occurred during appending: {e}")
```

```
Successfully appended new content to 'Data Analytics.txt'.
```

In [21]:

```python
""" Count the number of lines in the file """

line_count = 0
try:
    with open(file_name, 'r') as file:
        for line in file:
            line_count += 1
    print(f"Total number of lines in '{file_name}': {line_count}")
except FileNotFoundError:
    print(f"Error: The file '{file_name}' was not found for counting.")
except Exception as e:
    print(f"An error occurred during line counting: {e}")
```

```
Total number of lines in 'Data Analytics.txt': 7
```

TASK 5 : CSV / JSON & REGEX

In [23]:

```python
""" Create a CSV file and write sample student data """

import csv
import json
import re

csv_file_name = "students.csv"
student_data = [
    ['Name', 'Age', 'Course', 'City'],
    ['Kiran', 25, 'ADDA', 'ALUVA'],
    ['Arun', 42, 'TALLY', 'ANGAMALY'],
    ['Abhijith', 28, 'PYTHON', 'KOCHI']
]

try:
    with open(csv_file_name, 'w', newline='') as file:
        writer = csv.writer(file)
        writer.writerows(student_data)
    print(f"Successfully created and wrote data to '{csv_file_name}'.")
except Exception as e:
    print(f"An error occurred during CSV writing: {e}")
```

```
Successfully created and wrote data to 'students.csv'.
```

In [24]:

```python
""" Read the CSV file and print the contents """

try:
    with open(csv_file_name, 'r') as file:
        reader = csv.reader(file)
        for row in reader:
            print(', '.join(row))
except FileNotFoundError:
    print(f"Error: The file '{csv_file_name}' was not found.")
```

```
except Exception as e:
    print(f"An error occurred during CSV reading: {e}")
```

```
Name, Age, Course, City
Kiran, 25, ADDA, ALUVA
Arun, 42, TALLY, ANGAMALY
Abhijith, 28, PYTHON, KOCHI
```

In [25]:
```
""" Create a JSON file with at least 3 student records """

json_file_name = "student_records.json"
json_data = {
    "students": [
        {"id": 101, "name": "Alex", "major": "Computer Science"},
        {"id": 102, "name": "Ben", "major": "Statistics"},
        {"id": 103, "name": "Chris", "major": "Mathematics"}
    ]
}

try:
    with open(json_file_name, 'w') as file:
        json.dump(json_data, file, indent=4)
    print(f"\nSuccessfully created and wrote data to '{json_file_name}'.")
except Exception as e:
    print(f"An error occurred during JSON writing: {e}")
```

```

Successfully created and wrote data to 'student_records.json'.
```

In [26]:
```
"""  Use Regular Expressions to:
o Extract all numbers from a string
o Remove special characters from a text
o Validate a phone number """

sample_string = "The price is 1,250,00 rs and the quantity is 15 units."
special_char_text = "Hello! My name is A@#n$%^&*u."
phone_number1 = "123-456-7890"
phone_number2 = "(999) 888-7777"
invalid_phone = "123456789012"

# a. Extract all numbers from a string
numbers = re.findall(r'\d+', sample_string)
print(f"Original String: {sample_string}")
print(f"Extracted Numbers: {numbers}")

# b. Remove special characters from a text
# Keeps letters, numbers, and spaces
cleaned_text = re.sub(r'[^A-Za-z0-9\s]', '', special_char_text)
print(f"Original Text: {special_char_text}")
print(f"Cleaned Text: {cleaned_text}")

# c. Validate a phone number (simple pattern: XXX-XXX-XXXX or (XXX) XXX-XXX
phone_pattern = r'^\(?\d{3}\)?[-.\s]?\d{3}[-.\s]?\d{4}$'
```

```
def validate_phone(number):
    if re.fullmatch(phone_pattern, number):
        return "Valid"
    else:
        return "Invalid"

print(f"\nValidation for '{phone_number1}': {validate_phone(phone_number1)}
print(f"Validation for '{phone_number2}': {validate_phone(phone_number2)}")
print(f"Validation for '{invalid_phone}': {validate_phone(invalid_phone)}")
```

```
Original String: The price is 1,250,00 rs and the quantity is 15 units.
Extracted Numbers: ['1', '250', '00', '15']
Original Text: Hello! My name is A@#n$%^&*u.
Cleaned Text: Hello My name is Anu

Validation for '123-456-7890': Valid
Validation for '(999) 888-7777': Valid
Validation for '123456789012': Invalid
```

TASK 6 : MINI PYTHON PROGRAM

In [28]:

```
""" Write a mini program that:
• Takes user input
• Stores data in a file
• Reads the file content
• Cleans the data using regex
• Displays the final cleaned output """

import os
import re

FILE_NAME = 'user_data.txt'

def get_user_input():
    lines = []
    while True:
        line = input()
        if line == "":  # Blank line indicates end of input
            break
        lines.append(line)
    return '\n'.join(lines)

def store_data_to_file(data):
    with open(FILE_NAME, 'w') as file:
        file.write(data)
    print(f"Data successfully stored in '{FILE_NAME}'.")

def read_data_from_file():
    if not os.path.exists(FILE_NAME):
        print("Error: File not found!")
        return None
    with open(FILE_NAME, 'r') as file:
        return file.read()
```

```python
def clean_data_with_regex(raw_data):
    if not raw_data:
        return None

    # Step 1: Remove all characters except letters, numbers, and spaces
    cleaned = re.sub(r'[^a-zA-Z0-9\s]', '', raw_data)

    # Step 2: Replace multiple spaces/newlines with a single space
    cleaned = re.sub(r'\s+', ' ', cleaned)

    # Step 3: Strip leading/trailing whitespace
    cleaned = cleaned.strip()

    return cleaned

def display_cleaned_output(cleaned_data):
    if cleaned_data:
        print("\n--- Final Cleaned Output ---")
        print(cleaned_data)
    else:
        print("\nNo data to display after cleaning.")

def main():
    print("=== Mini Data Processing Program ===\n")

    # 1. Take user input
    raw_input = get_user_input()

    if not raw_input.strip():
        print("\nNo input provided. Exiting.")
        return

    # 2. Store to file
    store_data_to_file(raw_input)

    # 3. Read from file
    file_content = read_data_from_file()

    # 4. Clean using regex
    cleaned_output = clean_data_with_regex(file_content)

    # 5. Display cleaned result
    display_cleaned_output(cleaned_output)

    # Optional: Clean up the file
    try:
        if os.path.exists(FILE_NAME):
            os.remove(FILE_NAME)
            print(f"\nCleanup: Temporary file '{FILE_NAME}' removed.")
    except Exception as e:
        print(f"\nCould not remove file: {e}")
```

```python
# Run the program
if __name__ == "__main__":
    main()
```

```
=== Mini Data Processing Program ===


 Hello!!! This is my @data# with $special% characters... And   multiple    spaces!!


Data successfully stored in 'user_data.txt'.

--- Final Cleaned Output ---
Hello This is my data with special characters And multiple spaces

Cleanup: Temporary file 'user_data.txt' removed.
```

In [ ]:

```
```