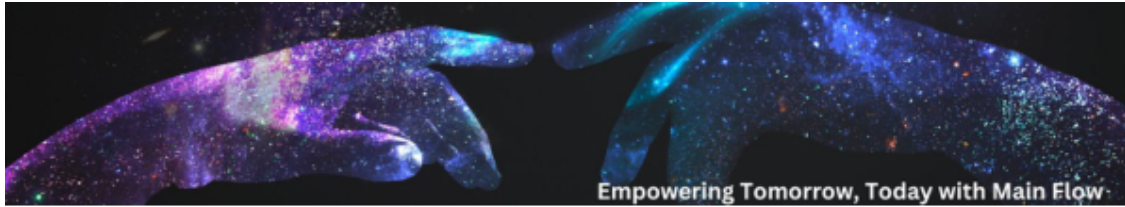


Main Flow Task_1

December 6, 2024

[1] :



PYTHON DEVELOPER

TASK-1

Basic Python Syntax Understanding

Description:

The intern will learn foundational Python concepts such as variables, data types, loops, and functions.

Responsibility:

1. Study Python syntax through tutorials and simple coding exercises.
2. Practice writing basic scripts to perform arithmetic operations, manipulate strings, and use conditional statements.
3. Gain familiarity with common data structures like lists, dictionaries, and tuples.

[]:

0.0.1 Task 1-1 Class Practice Code: Introduction to Python Basics

Focusing on foundational Python concepts. The Jupyter Notebook will cover **variables**, **data types**, and **basic operations**.

CoursePlan

1. Introduction to Python Syntax

- Print statements
- Variables and data types

2. Perform Arithmetic Operations

- Addition, subtraction, multiplication, and division
- Using variables in expressions

3. Manipulate Strings

- String concatenation and repetition
 - String methods (like `.upper()`, `.lower()`, etc.)
-

0.0.2 Practice Code

[]:

```
[2]: #### **Task 1: Print Statements and Variables**
```

```
[3]: # Print a welcome message  
print("Welcome to Python Basics!")
```

Welcome to Python Basics!

```
[4]: # Define variables  
name = "Satyanarayana"  
age = 28  
is_intern = True  
  
# Display variable values  
print("Name:", name)  
print("Age:", age)  
print("Is Intern:", is_intern)
```

Name: Satyanarayana

Age: 28

Is Intern: True

[]:

```
[5]: ##### **Task 2: Arithmetic Operations**
# Define two numbers
num1 = 10
num2 = 5

# Perform arithmetic operations
addition = num1 + num2
subtraction = num1 - num2
multiplication = num1 * num2
division = num1 / num2

# Print the results
print("Addition:", addition)
print("Subtraction:", subtraction)
print("Multiplication:", multiplication)
print("Division:", division)
```

```
Addition: 15
Subtraction: 5
Multiplication: 50
Division: 2.0
```

```
[ ]:
```

```
[6]: ##### **Task 3: String Manipulation**
# Define strings
greeting = "Hello"
name = "Mainflow"

# Concatenate strings
message = greeting + ", " + name + "!"
print(message)
```

```
Hello, Mainflow!
```

```
[ ]:
```

```
[7]: # String repetition
repeated = name * 3
print("Repeated Name:", repeated)

# Use string methods
print("Uppercase:", name.upper())
print("Lowercase:", name.lower())
```

```
Repeated Name: MainflowMainflowMainflow
Uppercase: MAINFLOW
Lowercase: mainflow
```

[]:

Exercises

1. Simple Calculator:

- Write a script to take two numbers as input and perform all arithmetic operations.

2. String Playground:

- Write a program to input a string and display:
 - Its length
 - Its first and last characters
 - The string reversed #####

[8]: #####

```
[9]: ## Exercise 1: Simple Calculator
#A script that takes two numbers as input and performs all arithmetic
#operations.
# Input two numbers from the user

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

# Perform arithmetic operations
addition = num1 + num2
subtraction = num1 - num2
multiplication = num1 * num2
division = num1 / num2 if num2 != 0 else "Division by zero not allowed"

# Display the results
print("\nResults:")
print(f"Addition: {num1} + {num2} = {addition}")
print(f"Subtraction: {num1} - {num2} = {subtraction}")
print(f"Multiplication: {num1} * {num2} = {multiplication}")
print(f"Division: {num1} / {num2} = {division}")
```

Enter the first number: 5

Enter the second number: 8

Results:

Addition: 5.0 + 8.0 = 13.0

Subtraction: 5.0 - 8.0 = -3.0

Multiplication: 5.0 * 8.0 = 40.0

Division: 5.0 / 8.0 = 0.625

[]:

```
[10]: ###Exercise 2: String Playground

# Input a string from the user
user_string = input("Enter a string: ")

# Display string details
print("\nString Analysis:")
print(f"Length of the string: {len(user_string)}")
print(f"First character: {user_string[0]}")
print(f"Last character: {user_string[-1]}")
print(f"Reversed string: {user_string[::-1]}")
```

Enter a string: Empowering Tomorrow, Today with Main Flow

String Analysis:

Length of the string: 41

First character: E

Last character: w

Reversed string: wolF niaM htiw yadoT ,worromOT gnirewopmE

[]:

[]:

[]:

0.0.3 Practice

1. Enhance the Simple Calculator:

- Add modulus (%) and exponentiation (**) operations.
- Handle cases where both numbers are zero.

2. Enhance the String Playground:

- Check if the string is a palindrome (reads the same backward as forward).
- Count the number of vowels in the string.

```
[11]: #####
#####
```

[]:

0.0.4 Task 1-2 Class Practice Code: Conditional Statements and Loops

Course Plan

1. Conditional Statements

- if, elif, and else

- Relational and logical operators
2. **Loops**
 - for loop for iterating over sequences
 - while loop for conditional iteration
 - Nested loops
 3. **Combine Concepts**
 - Mini-project using conditional statements and loops
-

[]:

```
[12]: ### **Practice Code**

#### **Task 1: Conditional Statements**
# Check if a number is positive, negative, or zero
num = int(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: 8

The number is positive.

```
[13]: # Check if a number is even or odd
if num % 2 == 0:
    print("The number is even.")
else:
    print("The number is odd.")
```

The number is even.

[]:

```
[14]: #### **Task 2: For Loop**
# Print numbers from 1 to 10
print("Numbers from 1 to 10:")
for i in range(1, 11):
    print(i)
```

Numbers from 1 to 10:

1
2
3
4

5
6
7
8
9
10

[]:

```
[15]: # Print the square of each number in a list
numbers = [2, 4, 6, 8]
print("Squares of the numbers:")
for num in numbers:
    print(num, "squared is", num ** 2)
```

Squares of the numbers:
2 squared is 4
4 squared is 16
6 squared is 36
8 squared is 64

[]:

```
[16]: #### **Task 3: While Loop**
# Countdown from 5 to 1
count = 5
print("Countdown:")
while count > 0:
    print(count)
    count -= 1
print("Blast off!")
```

Countdown:
5
4
3
2
1
Blast off!

[]:

```
[17]: ##### Sum of numbers until a negative number is entered
total = 0
while True:
    num = int(input("Enter a number (negative to stop): "))
    if num < 0:
        break
    total += num
```



```
print("Total sum:", total)
```

Enter a number (negative to stop): -7

Total sum: 0

[]:

```
[18]: ##### **Task 4: Nested Loops**
# Multiplication table
print("Multiplication Table:")
for i in range(1, 6):
    for j in range(1, 6):
        print(i * j, end="\t")
    print() # Newline after each row
```

Multiplication Table:

1	2	3	4	5
2	4	6	8	10
3	6	9	12	15
4	8	12	16	20
5	10	15	20	25

[]:

0.0.5 Exercises

1. Number Guessing Game:

- Write a program where the user guesses a randomly generated number between 1 and 20. Provide hints like “Too High” or “Too Low.”

2. FizzBuzz Problem:

- Write a program to print numbers from 1 to 50.
- For multiples of 3, print “Fizz” instead of the number.
- For multiples of 5, print “Buzz.”
- For multiples of both 3 and 5, print “FizzBuzz.”

3. Factorial Calculator:

- Write a program to calculate the factorial of a given number using a **while** loop.

```
[20]: #####
```

```
[23]: ##### Exercise 1: Number Guessing Game
#A game where the user guesses a randomly generated number between 1 and 20.

import random

# Generate a random number between 1 and 20
secret_number = random.randint(1, 20)
attempts = 0
```

```

print("I have chosen a number between 1 and 20. Can you guess it?")

while True:
    guess = int(input("Enter your guess: "))
    attempts += 1

    if guess < secret_number:
        print("Too Low!")
    elif guess > secret_number:
        print("Too High!")
    else:
        print(f"Congratulations! You guessed it in {attempts} attempts.")
        break

```

I have chosen a number between 1 and 20. Can you guess it?

Enter your guess: 17

Too High!

Enter your guess: 18

Too High!

Enter your guess: 10

Too Low!

Enter your guess: 15

Too Low!

Enter your guess: 16

Congratulations! You guessed it in 5 attempts.

[]:

```

[24]: ### Exercise 2: FizzBuzz Problem
#A program to print numbers from 1 to 50 with special rules for multiples of 3
↪and 5.
for num in range(1, 51):
    if num % 3 == 0 and num % 5 == 0:
        print("FizzBuzz")
    elif num % 3 == 0:
        print("Fizz")
    elif num % 5 == 0:
        print("Buzz")
    else:
        print(num)

```

1

2

Fizz
4
Buzz
Fizz
7
8
Fizz
Buzz
11
Fizz
13
14
FizzBuzz
16
17
Fizz
19
Buzz
Fizz
22
23
Fizz
Buzz
26
Fizz
28
29
FizzBuzz
31
32
Fizz
34
Buzz
Fizz
37
38
Fizz
Buzz
41
Fizz
43
44
FizzBuzz
46
47
Fizz
49
Buzz

[]:

```
[25]: ### Exercise 3: Factorial Calculator
#A program to calculate the factorial of a given number using a `while` loop.

# Input a number from the user
num = int(input("Enter a number to calculate its factorial: "))

# Initialize variables
factorial = 1
counter = num

while counter > 0:
    factorial *= counter
    counter -= 1

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

Enter a number to calculate its factorial: 8

The factorial of 8 is 40320

[]:

0.0.6 Practice

1. **Enhance the Number Guessing Game:**
 - Limit the number of attempts (e.g., 5).
 - Show the secret number if the user fails to guess correctly.
2. **FizzBuzz Variation:**
 - Accept the range (start and end) from the user instead of fixed numbers.
3. **Recursive Factorial:**
 - Implement the factorial calculator using a recursive function instead of a loop.

[]:

```
[ ]: #####
#####
```

0.0.7 Task 1-3 Class Practice Code: Functions and Data Structures

Class Plan

1. **Functions**
 - Defining and calling functions
 - Function parameters and return values
 - Default arguments and keyword arguments
2. **Data Structures**

- Lists: operations and methods
 - Dictionaries: creating, accessing, and modifying
 - Tuples: immutable sequences
 - Sets: unique elements and set operations
3. **Combine Concepts**
- Solve problems using functions and data structures.
-

0.0.8 Practice Code

[]:

```
[26]: ##### **Task 1: Functions**  
# Function to calculate the square of a number  
def square(num):  
    return num ** 2  
  
print("Square of 5:", square(5))
```

Square of 5: 25

[]:

```
[27]: # Function with default arguments  
def greet(name="Guest"):  
    print(f"Hello, {name}!")  
  
greet() # Default name  
greet("Mainflow") # Custom name
```

Hello, Guest!
Hello, Mainflow!

[]:

```
[28]: # Function to calculate factorial  
def factorial(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        return n * factorial(n - 1)  
  
print("Factorial of 5:", factorial(5))
```

Factorial of 5: 120

[]:

```
[29]: ##### **Task 2: Lists**
# Create and manipulate a list
fruits = ["apple", "banana", "cherry"]
print("Original list:", fruits)

# Add an element
fruits.append("orange")
print("After adding an element:", fruits)

# Remove an element
fruits.remove("banana")
print("After removing an element:", fruits)

# Access elements
print("First fruit:", fruits[0])

# Iterate over the list
for fruit in fruits:
    print("Fruit:", fruit)
```

Original list: ['apple', 'banana', 'cherry']
 After adding an element: ['apple', 'banana', 'cherry', 'orange']
 After removing an element: ['apple', 'cherry', 'orange']
 First fruit: apple
 Fruit: apple
 Fruit: cherry
 Fruit: orange

[]:

[]:

```
[30]: ##### **Task 3: Dictionaries**
# Create and access a dictionary
student = {"name": "John", "age": 21, "grade": "A"}
print("Student dictionary:", student)

# Access a value
print("Name:", student["name"])

# Add a new key-value pair
student["major"] = "Computer Science"
print("After adding a new key-value pair:", student)

# Iterate through keys and values
for key, value in student.items():
    print(f"{key}: {value}")
```

```
Student dictionary: {'name': 'John', 'age': 21, 'grade': 'A'}
Name: John
After adding a new key-value pair: {'name': 'John', 'age': 21, 'grade': 'A',
'major': 'Computer Science'}
name: John
age: 21
grade: A
major: Computer Science
```

[]:

```
[31]: ##### **Task 4: Tuples**
# Create and access a tuple
colors = ("red", "green", "blue")
print("Colors tuple:", colors)
print("First color:", colors[0])
```

```
Colors tuple: ('red', 'green', 'blue')
First color: red
```

[]:

```
[32]: # Tuples are immutable
# colors[0] = "yellow" # This will cause an error
##### **Task 5: Sets**
# Create and manipulate a set
numbers = {1, 2, 3, 4}
print("Original set:", numbers)

# Add an element
numbers.add(5)
print("After adding an element:", numbers)

# Remove an element
numbers.remove(3)
print("After removing an element:", numbers)
```

```
Original set: {1, 2, 3, 4}
After adding an element: {1, 2, 3, 4, 5}
After removing an element: {1, 2, 4, 5}
```

[]:

```
[33]: # Set operations
set1 = {1, 2, 3}
set2 = {3, 4, 5}
print("Union:", set1 | set2)
print("Intersection:", set1 & set2)
print("Difference:", set1 - set2)
```

Union: {1, 2, 3, 4, 5}
Intersection: {3}
Difference: {1, 2}

[]:

0.0.9 Practice

1. Grade Calculator:

- Write a function to calculate the grade based on a list of scores.
- Example: Input: [80, 90, 75], Output: "Average Score: 81.67, Grade: B"

2. Word Frequency Counter:

- Write a program to count the frequency of each word in a given sentence using a dictionary.

3. Unique Elements:

- Write a function that takes a list as input and returns a list of unique elements using a set.

4. Employee Management:

- Create a dictionary to store employee details (ID, Name, Salary).
- Write functions to:
 - Add a new employee
 - Update an employee's salary
 - Display all employee details

[]:

[]:

[]:

[]:

[]: