**Python – Playwright -Practice**

**Understand how to manipulate strings and print them.**

1. Create a variable named greeting and assign it the string "Welcome to Python Programming".
2. Print the greeting variable.
3. Modify the string to include your instructor name "Rahul!". For example, "Welcome to Python Programming, [Instructor Name]!" and print the modified string.

**Expected Result:**

1. Welcome to Python Programming
2. Welcome to Python Programming, Rahul!

**Program 1:**

# Create a variable named greeting

greeting = "Welcome to Python Programming"

# Print the greeting variable

print(greeting)

# Modify the string to include the instructor's name "Rahul!" and print

greeting = greeting + ", Rahul!"

print(greeting)

**Variable Assignment and Types**

1. Create three variables: age, height, and favorite\_color. Assign them values 25, 5.9, blue:
   * age: an integer (e.g., 25)
   * height: a float (e.g., 5.9)
   * favorite\_color: a string (e.g., "blue")
2. Use the print function to display each variable and its type using the type() function.

**Expected Result:**

1. Age: 25 | Type: <class 'int'>
2. Height: 5.9 | Type: <class 'float'>
3. Favorite Color: blue | Type: <class 'str'>

**Program 2:**

age = 25

height = 5.9

favorite\_color = "blue"

# Print each variable and its type

print("Age:", age, "| Type:", type(age))

print("Height:", height, "| Type:", type(height))

print("Favorite Color:", favorite\_color, "| Type:", type(favorite\_color))

**Working with Lists**

1. Create a list named fruits that contains below five different fruit names (strings).
   1. "apple", "banana", "cherry", "date", "elderberry"
2. Print the first and last elements of the list.
3. Use slicing to print the second and third fruits from the list.

**Expected Result:**

1. First fruit: apple
2. Last fruit: elderberry
3. Fruits from index 1 to 2: ['banana', 'cherry']

**Program 3:**

# Create a list named fruits

fruits = ["apple", "banana", "cherry", "date", "elderberry"]

# Print the first and last elements

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

print("Last fruit:", fruits[-1])

# Use slicing to print the second and third fruits (index 1 and 2)

print("Fruits from index 1 to 2:", fruits[1:3])

**Exploring Tuples**

1. Create a tuple named person that contains three elements: a name (string), an age (integer), and a height (float) with the below values.
2. "Rahul", 25, 5.9
3. Print the second element of the tuple.

**Expected Result:**

1. Age: 25

**Program 4:**

person = ("Rahul", 25, 5.9)

person= list(person)

print("Age:", person[1])

**Understanding Dictionaries**

1. Create a dictionary named car with the following keys: make, model, year, and color. Assign appropriate values to each key.
2. "make": "Toyota",

    "model": "Camry",

    "year": 2020,

    "color": "Blue"

1. Print the value associated with the model key.
2. Add a new key called owner and assign it the name "Rahul".
3. Print the entire dictionary.

**Expected Result:**

1. Car model: Camry
2. Updated car dictionary: {'make': 'Toyota', 'model': 'Camry', 'year': 2020, 'color': 'Blue', 'owner': 'Rahul'}

This topic is covered in

All changes saved

|

Line 1, Column 1

**Test Cases**

Failed: 0, Passed: 1 of 1 tests

**Program 5:**

car = {

"make": "Toyota",

"model": "Camry",

"year": 2020,

"color": "Blue"

}

print("Car model:", car["model"])

car.update({"owner":"Rahul"})

print("Updated car dictionary:", car)

**Practice using if-else statements.**

1. Write a program that assigns a greeting to a variable.
2. Use an if statement to check if the greeting is "Hello".
3. If it matches, print "Hello there!" and "How can I assist you today?".
4. If it does not match, print "Greetings!".
5. After the if-else block, print "Program has completed."

**Expected Output:**

* If greeting is "Hello":
  1. Hello there!
  2. How can I assist you today?
  3. Program has completed.
* If greeting is not "Hello":
  1. Greetings!
  2. Program has completed.

**Program 6:**

greeting = "Hello"

if greeting == "Hello":

print("Hello there!")

print("How can I assist you today?")

else:

print("Greetings!")

print("Program has completed.")

**Number Comparison**

1. Create a variable b and assign it a number 15.
2. Write an if statement to check if b is greater than 10.
3. If true, print "Number is greater than 10".
4. If false, print "Number is 10 or less".
5. Print "Comparison code is completed" at the end.

**Expected Output:**

* For b = 15:
  1. Number is greater than 10
  2. Comparison code is completed.
* For b = 5:
  1. Number is 10 or less
  2. Comparison code is completed.

**Program 7:**

b = 15

if b > 10 :

print("Number is greater than 10")

else:

print("Number is 10 or less")

print("Comparison code is completed.")

**Doubling Numbers with For Loop**

1. Create a list of numbers: [1, 4, 7, 10].
2. Use a for loop to iterate through the list.
3. Inside the loop, print each number multiplied by 3.

**Expected Output:**

1. 3
2. 12
3. 21
4. 30

**Program 8:**

numbers=[1, 4, 7, 10]

for num in numbers:

print(num \*3)

**Customized Greeting Based on Time of Day**

1. Create a variable user and assign the value 16
2. Use if-elif-else statements to print:
   * "Good Morning" if the hour is between 5 and 11,
   * "Good Afternoon" if the hour is between 12 and 17,
   * "Good Evening" if the hour is between 18 and 21,
   * "Good Night" for all other hours.
3. Print "Greeting code has completed."

**Expected Output:**

* For input 10:
  1. Good Morning
  2. Greeting code has completed.
* For input 15:
  1. Good Afternoon
  2. Greeting code has completed.

**Program 9:**

hour = 16

if 5 <= hour <= 11:

print("Good Morning")

elif 12 <= hour <= 17:

print("Good Afternoon")

elif 18 <= hour <= 21:

print("Good Evening")

else:

print("Good Night")

print("Greeting code has completed.")

**Understanding class creation in Python**

**Objective**: Create a basic calculator class to perform addition, subtraction, multiplication, and division.

**Instructions**:

1. Create a class named BasicCalculator.
2. Define a constructor that initializes two numbers. Use numbers 10 & 5
3. Implement methods for:
   * Addition
   * Subtraction
   * Multiplication
   * Division
4. Each method should return the result of the operation.
5. Create an instance of the BasicCalculator class and demonstrate the functionality of each method.

**Example Output**:

1. Addition: 10 + 5 = 15
2. Subtraction: 10 - 5 = 5
3. Multiplication: 10 \* 5 = 50
4. Division: 10 / 5 = 2.0

**Program 10:**

1. class BasicCalculator:
2. def \_\_init\_\_(self, a, b):
3. self.firstNumber = a
4. self.secondNumber = b
6. def addition(self):
7. return self.firstNumber + self.secondNumber
9. def subtraction(self):
10. return self.firstNumber - self.secondNumber
12. def multiplication(self):
13. return self.firstNumber \* self.secondNumber
15. def division(self):
16. if self.secondNumber != 0:
17. return self.firstNumber / self.secondNumber
18. else:
19. return "Division by zero is not allowed."
21. # Demonstration
22. calc = BasicCalculator(10, 5)
23. print("Addition:", calc.addition())
24. print("Subtraction:", calc.subtraction())
25. print("Multiplication:", calc.multiplication())
26. print("Division:", calc.division())

**Create a Greeting function**

**Objective:** Create a function that greets the user.

**Instructions:**

1. Write a function called GreetUser that takes a single argument username.
2. The function should print "Hello, [username]! Welcome to the Python course."
3. Call the function with username "John".

**Expected Output:**

1. Hello, John! Welcome to the Python course.

**Program 11:**

def GreetUser(username):

print("Hello, " +username + "! Welcome to the Python course.")

GreetUser("John")

**Average Calculator**

**Objective:** Calculate the average of three numbers.

**Instructions:**

1. Create a function called CalculateAverage that takes three parameters: num1, num2, and num3.
2. Use the numbers 10,20,30 as input to functions
3. The function should return the average of these three numbers.

**Expected Output:**

1. The average of 10, 20, and 30 is 20.0

**Program 12:**

# Function to calculate the average of three numbers

def CalculateAverage(num1, num2, num3):

return (num1 + num2 + num3) / 3

# Calling the function with different sets of numbers

average1 = CalculateAverage(10, 20, 30)

print("The average of 10, 20, and 30 is", average1)

**Read and Print File Contents**

**Objective:** Write a program to read the contents of a file and print it to the console.

**Instructions:**

1. Sample file ***file1.txt*** is provided with this assignment.
2. Write a Python script that opens the file and reads all its contents.
3. Print the entire content of the file.

**Program 13:**

with open('file1.txt', 'r') as file:

content = file.read()

print(content)

**file1.txt**

Hello, there!

how are you?

Hope you are enjoying this course.

These exercises are designed to solidify your learning.

happy learning!

**Count Lines in a File**

**Objective:** Count and print the number of lines in a file.

**Instructions:**

1. Use the attached text file ***"file1.txt"***
2. Write a Python script that opens the file, reads through it line by line, counts how many lines it has, and prints the total count.

**Expected result:**

1. Total number of lines: 5

**Program 14:**

with open('file1.txt', 'r') as file:

count = sum(1 for line in file)

print(f'Total number of lines: {count}')

**Shopping Cart Validation**

**Instructions:**

1. Create a variable ItemsInCart and initialize it to 0.
2. Write a function called add\_to\_cart that accepts an integer parameter items\_to\_add.
3. If items\_to\_add is less than 0, raise an exception with the message "Cannot add a negative number of items."
4. If the total count of items after addition exceeds 5, raise an exception with the message "Cart limit exceeded."
5. # Example function structure
6. def add\_to\_cart(items\_to\_add):
7. # Your code here

**# Example of using the function**

1. try:
2. add\_to\_cart(2) # Add 2 items
3. add\_to\_cart(-1) # This should raise an exception
4. except Exception as e:
5. print(e)

**Expected Result:**

1. 2 items added. Total in cart: 2
2. Cannot add a negative number of items.

**Program 15:**

1. ItemsInCart = 0
3. def add\_to\_cart(items\_to\_add):
4. global ItemsInCart
5. if items\_to\_add < 0:
6. raise Exception("Cannot add a negative number of items.")
8. if ItemsInCart + items\_to\_add > 5:
9. raise Exception("Cart limit exceeded.")
11. ItemsInCart += items\_to\_add
12. print(f"{items\_to\_add} items added. Total in cart: {ItemsInCart}")
14. # Example of using the function
15. try:
16. add\_to\_cart(2) # Add 2 items
17. add\_to\_cart(-1) # This should raise an exception
18. except Exception as e:
19. print(e)

**Handle Tuple modification exception with Try Catch**

1. Create a tuple named person that contains three elements: a name (string), an age (integer), and a height (float) with the below values.
2. ***"Rahul", 25, 5.9***
3. Print the second element of the tuple.
4. Attempt to change the name in the tuple to a different name and explain why this will or will not work.

**Program 16:**

person = ("Rahul", 25, 5.9)

# Print the second element of the tuple

print(f"Age: {person[1]}")

# Attempt to change the name in the tuple (this will raise an error)

try:

person[0] = "John" # This will not work

except Exception as e:

print(f"Error: {e} - Tuples are immutable.")