

Python Assignment

Arithmetic operations between 2 numbers

main.py	Run	Output
<pre>1 a = 46 2 b = 4 3 print("For a =", a, "and b =", b, "\nCalculate the following:") 4 print('1. Addition of two numbers: a + b =', a + b) 5 print('2. Subtraction of two numbers: a - b =', a - b) 6 print('3. Multiplication of two numbers: a * b =', a * b) 7 print('4. Division of two numbers: a / b =', a / b) 8 print('5. Floor division of two numbers: a // b =', a // b) 9 print('6. Remainder of two numbers: a mod b =', a % b) 10 print('7. Exponent of two numbers: a ^ b =', a ** b)</pre>		<pre>For a = 46 and b = 4 Calculate the following: 1. Addition of two numbers: a + b = 50 2. Subtraction of two numbers: a - b = 42 3. Multiplication of two numbers: a * b = 184 4. Division of two numbers: a / b = 11.5 5. Floor division of two numbers: a // b = 11 6. Remainder of two numbers: a mod b = 2 7. Exponent of two numbers: a ^ b = 4477456 === Code Execution Successful ===</pre>

Comparison operations between 2 numbers

main.py	Run	Output
<pre>1 a = 46 2 b = 4 3 print("For a =", a, "and b =", b, "\nCheck the following:") 4 print('1. Two numbers are equal or not:', a == b) 5 print('2. Two numbers are not equal or not:', a != b) 6 print('3. a is less than or equal to b:', a <= b) 7 print('4. a is greater than or equal to b:', a >= b) 8 print('5. a is greater b:', a > b) 9 print('6. a is less than b:', a < b)</pre>		<pre>For a = 46 and b = 4 Check the following: 1. Two numbers are equal or not: False 2. Two numbers are not equal or not: True 3. a is less than or equal to b: False 4. a is greater than or equal to b: True 5. a is greater b: True 6. a is less than b: False === Code Execution Successful ===</pre>

Assignment operators





main.py	Run	Output
<pre>1 a = 34 2 b = 6 3 print('a += b:', a + b) 4 print('a -= b:', a - b) 5 print('a *= b:', a * b) 6 print('a /= b:', a / b) 7 print('a %= b:', a % b) 8 print('a **= b:', a ** b) 9 print('a //= b:', a // b)</pre>		<pre>a += b: 40 a -= b: 28 a *= b: 204 a /= b: 5.666666666666667 a %= b: 4 a **= b: 1544804416 a //= b: 5 === Code Execution Successful ===</pre>

Bitwise operation between 2 numbers




main.py	Run	Output
<pre>1 a = 7 2 b = 8 3 print('a & b :', a & b) 4 print('a b :', a b) 5 print('a ^ b :', a ^ b) 6 print('~a :', ~a) 7 print('a << b :', a << b) 8 print('a >> b :', a >> b)</pre>		<pre>a & b : 0 a b : 15 a ^ b : 15 ~a : -8 a << b : 1792 a >> b : 0 === Code Execution Successful ===</pre>

Python Assignment




Logical operators

main.py	   Share Run	Output  Clear
<pre>1 a = 7 2 print("For a = 7, checking whether the following conditions are True or False:") 3 print('\a > 5 and a < 7' =>, a > 5 and a < 7) 4 print('\a > 5 or a < 7' =>, a > 5 or a < 7) 5 print('\a not (a > 5 and a < 7)' =>, not(a > 5 and a < 7))</pre>		<pre>For a = 7, checking whether the following conditions are True or False: "a > 5 and a < 7" => False "a > 5 or a < 7" => True "not (a > 5 and a < 7)" => True === Code Execution Successful ===</pre>



Membership operators

main.py	   Share Run	Output
<pre>1 myList = [12, 22, 28, 35, 42, 49, 54, 65, 92, 103, 245, 874] 2 x = 31 3 y = 28 4 print("Given List:", myList) 5 if x not in myList: 6 print("x =", x, "is NOT present in the given list.") 7 else: 8 print("x =", x, "is present in the given list.") 9 if y in myList: 10 print("y =", y, "is present in the given list.") 11 else: 12 print("y =", y, "is NOT present in the given list.")</pre>		<pre>Given List: [12, 22, 28, 35, 42, 49, 54, 65, 92, 103, 245, 874] x = 31 is NOT present in the given list. y = 28 is present in the given list. === Code Execution Successful ===</pre>

Identity operators

main.py	   Share Run	Output
<pre>1 a = ["Rose", "Lotus"] 2 b = ["Rose", "Lotus"] 3 c = a 4 print("a is c =>", a is c) 5 print("a is not c =>", a is not c) 6 print("a is b =>", a is b) 7 print("a is not b =>", a is not b) 8 print("a == b =>", a == b) 9 print("a != b =>", a != b)</pre>		<pre>a is c => True a is not c => False a is b => False a is not b => True a == b => True a != b => False === Code Execution Successful ===</pre>

Reverse a string using for loop

main.py	   Share Run	Output
<pre>1 def reverse_string(string): 2 str1 = "" 3 for i in string: 4 str1 = i + str1 5 return str1 6 7 string = "JavaTpoint" 8 print("The original string is:", string) 9 print("The reverse string is:", reverse_string(string))</pre>		<pre>The original string is: JavaTpoint The reverse string is: tniopTavaJ === Code Execution Successful ===</pre>

Python Assignment

Reverse a string using while loop

main.py	Output
<pre>1 str = "JavaTpoint" 2 print("The original string is:", str) 3 4 reverse_String = "" 5 count = len(str) 6 7 while count > 0: 8 reverse_String += str[count - 1] 9 count = count - 1 10 11 print("The reversed string using a while loop is:", reverse_String)</pre>	<pre>The original string is: JavaTpoint The reversed string using a while loop is: tniopTavaJ === Code Execution Successful ===</pre>

Reverse a string using slice operator

main.py	Output
<pre>1 def reverse(s): 2 s = s[::-1] 3 return s 4 5 s = "JavaTpoint" 6 print("The original string is:", s) 7 print("The reversed string using extended slice operator is:", reverse(s))</pre>	<pre>The original string is: JavaTpoint The reversed string using extended slice operator is: tniopTavaJ === Code Execution Successful ===</pre>

Reverse function using join

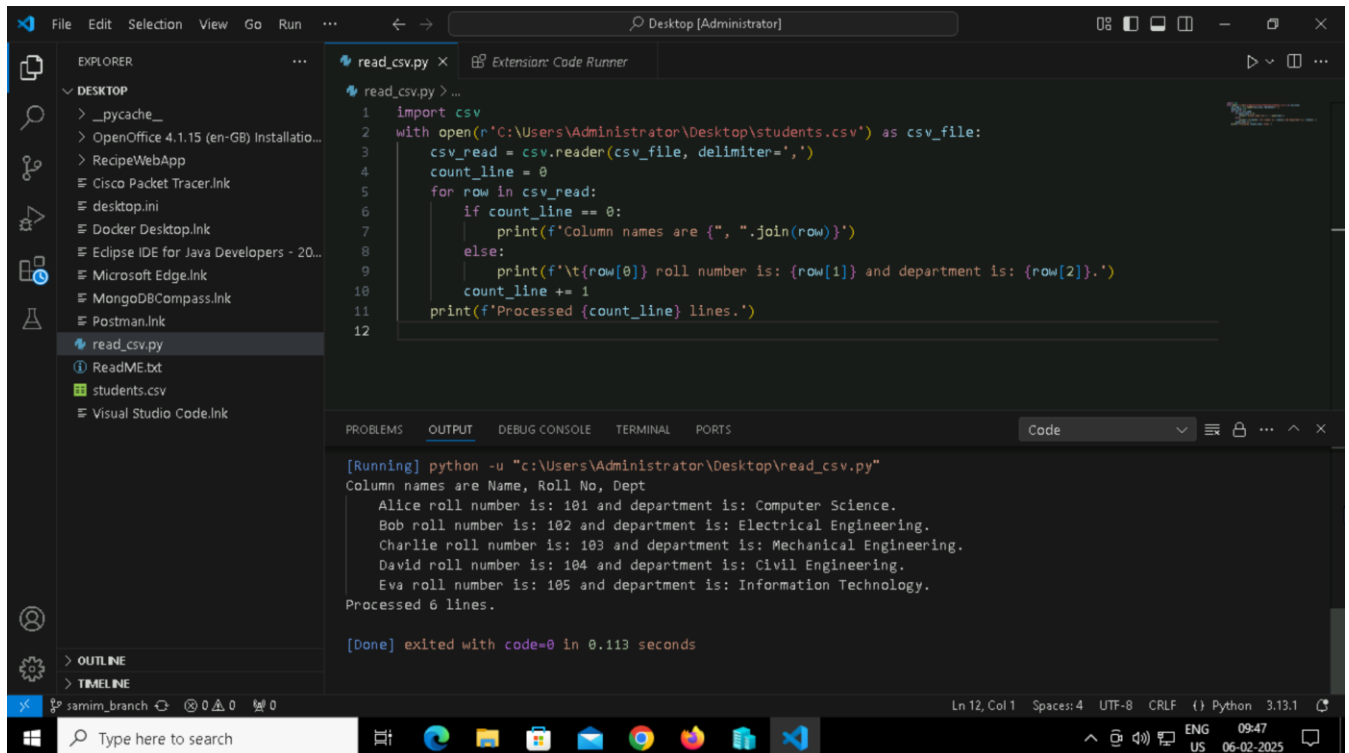
main.py	Output
<pre>1 def reverse(s): 2 string = "".join(reversed(s)) 3 return string 4 5 s = "JavaTpoint" 6 print("The original string is:", s) 7 print("The reversed string using reversed() is:", reverse(s))</pre>	<pre>The original string is: JavaTpoint The reversed string using reversed() is: tniopTavaJ === Code Execution Successful ===</pre>

Reverse using recursion

main.py	Output
<pre>1 def reverse(s): 2 string = "".join(reversed(s)) 3 return string 4 5 s = "JavaTpoint" 6 print("The original string is:", s) 7 print("The reversed string using reversed() is:", reverse(s))</pre>	<pre>The original string is: JavaTpoint The reversed string using reversed() is: tniopTavaJ === Code Execution Successful ===</pre>

Python Assignment

Read a csv file in python



The screenshot shows a Visual Studio Code editor with a Python script named `read_csv.py` open. The script reads a CSV file named `students.csv` from the desktop. The output of the script is displayed in the Output window, showing the column names and the data for each row.

```
1 import csv
2 with open(r'C:\Users\Administrator\Desktop\students.csv') as csv_file:
3     csv_read = csv.reader(csv_file, delimiter=',')
4     count_line = 0
5     for row in csv_read:
6         if count_line == 0:
7             print(f'Column names are {", ".join(row)}')
8         else:
9             print(f'\t{row[0]} roll number is: {row[1]} and department is: {row[2]}')
10            count_line += 1
11    print(f'Processed {count_line} lines.')
12
```

Output:

```
[Running] python -u "c:\Users\Administrator\Desktop\read_csv.py"
Column names are Name, Roll No, Dept
Alice roll number is: 101 and department is: Computer Science.
Bob roll number is: 102 and department is: Electrical Engineering.
Charlie roll number is: 103 and department is: Mechanical Engineering.
David roll number is: 104 and department is: Civil Engineering.
Eva roll number is: 105 and department is: Information Technology.
Processed 6 lines.

[Done] exited with code=0 in 0.113 seconds
```

Simple python program to understand if statement

main.py	Output
<pre>1 num = int(input("enter the number:")) 2 if num%2 == 0: 3 print("The Given number is an even number")</pre>	<pre>enter the number:10 The Given number is an even number === Code Execution Successful ===</pre>

Program to print largest of 3 numbers




main.py	Output
<pre>1 a = int (input("Enter a: ")); 2 b = int (input("Enter b: ")); 3 c = int (input("Enter c: ")); 4 if a>b and a>c: 5 print ("From the above three numbers given a is largest"); 6 if b>a and b>c: 7 print ("From the above three numbers given b is largest"); 8 if c>a and c>b: 9 print ("From the above three numbers given c is largest");</pre>	<pre>Enter a: 1 Enter b: 2 Enter c: 3 From the above three numbers given c is largest === Code Execution Successful ===</pre>

Program to check whether a person is eligible to vote or not




main.py	Output
<pre>1 age = int (input("Enter your age: ")) 2 if age>=18: 3 print("You are eligible to vote !!"); 4 else: 5 print("Sorry! you have to wait !!");</pre>	<pre>Enter your age: 20 You are eligible to vote !! === Code Execution Successful ===</pre>

Python Assignment




Program to check if the given number is even or odd

main.py	   Share	Run	Output
<pre>1 num = int(input("enter the number:")) 2 if num%2 == 0: 3 print("The Given number is an even number") 4 else: 5 print("The Given Number is an odd number")</pre>			<pre>enter the number:9 The Given Number is an odd number === Code Execution Successful ===</pre>




Python program 1 elif statement

main.py	   Share	Run	Output
<pre>1 number = int(input("Enter the number:")) 2 if number == 10: 3 print("The given number is equals to 10") 4 elif number == 50: 5 print("The given number is equal to 50") 6 elif number == 100: 7 print("The given number is equal to 100") 8 else: 9 print("The given number is not equal to 10, 50 or 100")</pre>			<pre>Enter the number:10 The given number is equals to 10 === Code Execution Successful ===</pre>

Python program 2 elif statement

main.py	   Share	Run	Output
<pre>1 marks = int(input("Enter the marks:")) 2 if marks > 85 and marks <= 100: 3 print("Congrats ! you scored grade A ...") 4 elif marks > 60 and marks <= 85: 5 print("You scored grade B + ...") 6 elif marks > 40 and marks <= 60: 7 print("You scored grade B ...") 8 elif marks > 30 and marks <= 40: 9 print("You scored grade C ...") 10 else: 11 print("Sorry you failed")</pre>			<pre>Enter the marks:60 You scored grade B ... === Code Execution Successful ===</pre>

List of squares using for loop

main.py	   Share	Run	Output	Clear
<pre>1 numbers = [4, 2, 6, 7, 3, 5, 8, 10, 6, 1, 9, 2] 2 square = 0 3 squares = [] 4 for value in numbers: 5 square = value ** 2 6 squares.append(square) 7 print("The list of squares is", squares)</pre>			<pre>The list of squares is [16, 4, 36, 49, 9, 25, 64, 100, 36, 1, 81, 4] === Code Execution Successful ===</pre>	

Python Assignment

Using loop in string manipulation

main.py	Output
<pre>1 string = "Python Loop" 2 for s in string: 3 if s == "o": 4 print("If block") 5 else: 6 print(s)</pre>	<pre>P y t h If block n L If block If block p === Code Execution Successful ===</pre>

Using else statement with for loop

main.py	Output
<pre>1 tuple_ = (3, 4, 6, 8, 9, 2, 3, 8, 9, 7) 2 for value in tuple_: 3 if value % 2 != 0: 4 print(value) 5 else: 6 print("These are the odd numbers present in the tuple")</pre>	<pre>3 These are the odd numbers present in the tuple These are the odd numbers present in the tuple These are the odd numbers present in the tuple 9 These are the odd numbers present in the tuple 3 These are the odd numbers present in the tuple 9 7 === Code Execution Successful ===</pre>

Range function

main.py	Output
<pre>1 print(range(15)) 2 print(list(range(15))) 3 print(list(range(4, 9))) 4 print(list(range(5, 25, 4)))</pre>	<pre>range(0, 15) [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14] [4, 5, 6, 7, 8] [5, 9, 13, 17, 21] === Code Execution Successful ===</pre>

Python program to iterate over a sequence using with the help of indexing

main.py	Output
<pre>1 tuple_ = ("Python", "Loops", "Sequence", "Condition", "Range") 2 for iterator in range(len(tuple_)): 3 print(tuple_[iterator].upper())</pre>	<pre>PYTHON LOOPS SEQUENCE CONDITION RANGE === Code Execution Successful ===</pre>

Python Assignment

While loop

main.py		Output
<pre>1 counter = 0 2 while counter < 10: 3 counter = counter + 3 4 print("Python Loops")</pre>	<div>Run</div>	<div>Python Loops Python Loops Python Loops Python Loops</div> <div>=== Code Execution Successful ===</div>

Else statement inside while loop

main.py		Output
<pre>1 counter = 0 2 while counter < 10: 3 counter = counter + 3 4 print("Python Loops") 5 else: 6 print("Code block inside the else statement")</pre>	<div>Run</div>	<div>Python Loops Python Loops Python Loops Python Loops Code block inside the else statement</div> <div>=== Code Execution Successful ===</div>

Single statement while block

main.py		Output
<pre>1 counter = 0 2 while counter < 3: 3 print("Python Loops") 4 counter += 1</pre>	<div>Run</div>	<div>Python Loops Python Loops Python Loops</div> <div>=== Code Execution Successful ===</div>

Continue statement

main.py		Output
<pre>1 for string in "Python Loops": 2 if string == "o" or string == "p" or string == "t": 3 continue 4 print('Current Letter:', string)</pre>	<div>Run</div>	<div>Current Letter: P Current Letter: y Current Letter: h Current Letter: n Current Letter: Current Letter: L Current Letter: s</div> <div>=== Code Execution Successful ===</div>

Python Assignment

Break statement

main.py	Output
<pre>1 for string in "Python Loops": 2 if string == 'l': 3 break 4 print('Current Letter: ', string)</pre>	Current Letter: P Current Letter: y Current Letter: t Current Letter: h Current Letter: o Current Letter: n Current Letter: === Code Execution Successful ===

Pass statement

main.py	Output
<pre>1 for string in "Python Loops": 2 pass 3 print('Last Letter:', string)</pre>	Last Letter: s === Code Execution Successful ===

Code to find the sum of squares of each element of the list using for loop

main.py	Output
<pre>1 numbers = [3, 5, 23, 6, 5, 1, 2, 9, 8] 2 sum_ = 0 3 for num in numbers: 4 sum_ = sum_ + num ** 2 5 print("The sum of squares is: ", sum_)</pre>	The sum of squares is: 774 === Code Execution Successful ===

Code to find the sum of squares of each element of the list using for loop

main.py	Output
<pre>1 numbers = [3, 5, 23, 6, 5, 1, 2, 9, 8] 2 sum_ = 0 3 for num in range(len(numbers)): 4 sum_ = sum_ + numbers[num] ** 2 5 print("The sum of squares is: ", sum_)</pre>	The sum of squares is: 774 === Code Execution Successful ===

code to print marks of a student from the record

main.py	Output
<pre>1 student_name_1 = 'Itika' 2 student_name_2 = 'Parker' 3 records = {'Itika': 90, 'Arshia': 92, 'Peter': 46} 4 def marks(student_name): 5 for a_student in records: 6 if a_student == student_name: 7 return records[a_student] 8 return f'There is no student of name {student_name} in the record' 9 print(f'Marks of {student_name_1} are: ", marks(student_name_1)) 10 print(f'Marks of {student_name_2} are: ", marks(student_name_2))</pre>	Marks of Itika are: 90 Marks of Parker are: There is no student of name Parker in the record === Code Execution Successful ===

Python Assignment

Nested loops

main.py	Output
<pre>1 import random 2 numbers = [] 3 for val in range(0, 11): 4 numbers.append(random.randint(0, 11)) 5 for num in range(0, 11): 6 for i in numbers: 7 if num == i: 8 print(num, end=" ")</pre>	<pre>0 2 4 4 5 8 8 8 9 9 === Code Execution Successful ===</pre>

while loops in Python for printing numbers from 1 to 10

main.py	Output
<pre>1 i = 1 2 while i <= 10: 3 print(i, end=' ') 4 i += 1</pre>	<pre>1 2 3 4 5 6 7 8 9 10 === Code Execution Successful ===</pre>

while loops in Python for Printing those numbers divisible by either 5 or 7 within 1 to 50

main.py	Output
<pre>1 i = 1 2 while i < 51: 3 if i % 5 == 0 or i % 7 == 0: 4 print(i, end=' ') 5 i += 1</pre>	<pre>5 7 10 14 15 20 21 25 28 30 35 40 42 45 49 50 === Code Execution Successful ===</pre>

the sum of squares of the first 15 natural numbers using a while loop.

main.py	Output
<pre>1 num = 15 2 summation = 0 3 c = 1 4 while c <= num: 5 summation = c**2 + summation 6 c = c + 1 7 print("The sum of squares is", summation)</pre>	<pre>The sum of squares is 1240 === Code Execution Successful ===</pre>

Python Assignment

while loops in Python for a number is Prime number or not.

main.py	Output
<pre>1 num = [34, 12, 54, 23, 75, 34, 11] 2 def prime_number(number): 3 condition = 0 4 iteration = 2 5 while iteration <= number / 2: 6 if number % iteration == 0: 7 condition = 1 8 break 9 iteration = iteration + 1 10 if condition == 0: 11 print(f"{number} is a PRIME number") 12 else: 13 print(f"{number} is not a PRIME number") 14 for i in num: 15 prime_number(i)</pre>	<pre>34 is not a PRIME number 12 is not a PRIME number 54 is not a PRIME number 23 is a PRIME number 75 is not a PRIME number 34 is not a PRIME number 11 is a PRIME number === Code Execution Successful ===</pre>

while loops in Python for a number is Armstrong number or not.

```
main.py
1 n = int(input())
2 n1 = str(n)
3 l = len(n1)
4 temp = n
5 s = 0
6 while n != 0:
7     r = n % 10
8     s = s + (r ** l)
9     n = n // 10
10 if s == temp:
11     print("It is an Armstrong number")
12 else:
13     print("It is not an Armstrong number")
```

```
153
It is an Armstrong number

...Program finished with exit code 0
Press ENTER to exit console.
```

Python Assignment

while loop for printing the multiplication table of a given number.

main.py	Output
<pre>1 num = 21 2 counter = 1 3 print("The Multiplication Table of: ", num) 4 while counter <= 10: 5 ans = num * counter 6 print(num, 'x', counter, '=', ans) 7 counter += 1</pre>	<pre>The Multiplication Table of: 21 21 x 1 = 21 21 x 2 = 42 21 x 3 = 63 21 x 4 = 84 21 x 5 = 105 21 x 6 = 126 21 x 7 = 147 21 x 8 = 168 21 x 9 = 189 21 x 10 = 210 === Code Execution Successful ===</pre>

while loops in Python for square every number of a list

main.py	Output
<pre>1 list_ = [3, 5, 1, 4, 6] 2 squares = [] 3 while list_: 4 squares.append((list_.pop())**2) 5 print(squares)</pre>	<pre>[36, 16, 1, 25, 9] === Code Execution Successful ===</pre>

while loops in Python for determine odd and even number from every number of a list

main.py	Output
<pre>1 list_ = [3, 4, 8, 10, 34, 45, 67, 80] 2 index = 0 3 while index < len(list_): 4 element = list_[index] 5 if element % 2 == 0: 6 print('It is an even number') 7 else: 8 print('It is an odd number') 9 index += 1</pre>	<pre>It is an odd number It is an even number It is an even number It is an even number It is an even number It is an odd number It is an odd number It is an even number === Code Execution Successful ===</pre>

while loops in Python for determine the number letters of every word from the given list.

main.py	Output
<pre>1 List_ = ['Priya', 'Neha', 'Cow', 'To'] 2 index = 0 3 while index < len(List_): 4 element = List_[index] 5 print(len(element)) 6 index += 1</pre>	<pre>5 4 3 2 === Code Execution Successful ===</pre>

Python Assignment

while loops in Python for multiple condition.

main.py	Output
<pre>1 num1 = 17 2 num2 = -12 3 while num1 > 5 and num2 < -5: 4 num1 -= 2 5 num2 += 3 6 print((num1, num2))</pre>	<pre>(15, -9) (13, -6) (11, -3) === Code Execution Successful ===</pre>

break statement with for loop

main.py	Output
<pre>1 my_list = [1, 2, 3, 4] 2 count = 1 3 for item in my_list: 4 if item == 4: 5 print("Item matched") 6 count += 1 7 break 8 print("Found at location", count)</pre>	<pre>Item matched Found at location 2 === Code Execution Successful ===</pre>

Breaking out of a loop early

main.py	Output
<pre>1 my_str = "python" 2 for char in my_str: 3 if char == 'o': 4 break 5 print(char)</pre>	<pre>p y t h === Code Execution Successful ===</pre>

break statement with while loop

main.py	Output
<pre>1 i = 0 2 while 1: 3 print(i, " ", end="") 4 i = i + 1 5 if i == 10: 6 break 7 print("came out of while loop")</pre>	<pre>0 1 2 3 4 5 6 7 8 9 came out of while loop === Code Execution Successful ===</pre>

Python Assignment

break statement with nested loops

main.py	Output
1 n = 2	2 X 1 = 2
2 while True:	
3 i = 1	2 X 2 = 4
4 while i <= 10:	
5 print("%d X %d = %d\n" % (n, i, n * i))	2 X 3 = 6
6 i += 1	
7 choice = int(input("Do you want to continue printing the table? Press 0 for no: "))	2 X 4 = 8
8 if choice == 0:	2 X 5 = 10
9 print("Exiting the program...")	
10 break	2 X 6 = 12
11 n += 1	2 X 7 = 14
12 print("Program finished successfully.")	2 X 8 = 16
	2 X 9 = 18
	2 X 10 = 20

gleads.g.doubleclick.net/pcs/click?xai=AKAOjstvIWIXGTvoXB9h88...

Python Continue Statements in while Loop

main.py	Output
1 string = "JavaTpoint"	
2 iterator = 0	
3 while iterator < len(string):	
4 if string[iterator] == 'a':	
5 continue	
6 print(string[iterator])	
7 iterator += 1	

Python Continue statement in list comprehension

main.py	Output
1 numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]	
2 sq_num = [num ** 2 for num in numbers if num % 2 == 0]	[4, 16, 36, 64, 100]
3 print(sq_num)	=== Code Execution Successful ===

Creating String in Python

main.py	Output
1 str1 = 'Hello Python'	Hello Python
2 print(str1)	Hello Python
3 str2 = "Hello Python"	
4 print(str2)	Triple quotes are generally used for representing multiline strings or docstrings
5 str3 = '''Triple quotes are generally used for representing multiline strings or docstrings'''	
6	
7	
8 print(str3)	=== Code Execution Successful ===

Python Assignment

Strings indexing and splitting

Example 1

main.py	Output
<pre>1 str = "HELLO" 2 print(str[0]) 3 print(str[1]) 4 print(str[2]) 5 print(str[3]) 6 print(str[4]) 7 print(str[6])</pre>	<pre>H E L L O ERROR! Traceback (most recent call last): File "<main.py>", line 7, in <module> IndexError: string index out of range === Code Exited With Errors ===</pre>

Example 2

main.py	Output
<pre>1 str = "JAVATPOINT" 2 print(str[0:]) 3 print(str[1:5]) 4 print(str[2:4]) 5 print(str[:3]) 6 print(str[4:7])</pre>	<pre>JAVATPOINT AVAT VA JAV TPO === Code Execution Successful ===</pre>

Reverse

main.py	Output
<pre>1 str = 'JAVATPOINT' 2 print(str[-1]) 3 print(str[-3]) 4 print(str[-2:]) 5 print(str[-4:-1]) 6 print(str[-7:-2]) 7 print(str[::-1]) 8 print(str[-12]) 9</pre>	<pre>T I NT OIN ATPOI TNIPTAVAJ ERROR! Traceback (most recent call last): File "<main.py>", line 8, in <module> IndexError: string index out of range === Code Exited With Errors ===</pre>

Reassigning Strings

main.py	Output
<pre>1 str = "HELLO" 2 print(str) 3 str = "hello" 4 print(str)</pre>	<pre>HELLO hello === Code Execution Successful ===</pre>

Deleting the String

Python Assignment

Deleting a string

main.py	Output
<pre>1 str1 = "JAVATPOINT" 2 del str1 3 print(str1)</pre>	<pre>ERROR! Traceback (most recent call last): File "<main.py>", line 3, in <module> NameError: name 'str1' is not defined. Did you mean: 'str'? === Code Exited With Errors ===</pre>

use of Python operators

main.py	Output
<pre>1 str = "Hello" 2 str1 = " world" 3 print(str*3) 4 print(str+str1) 5 print(str[4]) 6 print(str[2:4]) 7 print('w' in str) 8 print('wo' not in str1) 9 print(r'C://python37') 10 print("The string str : %s"%(str))</pre>	<pre>HelloHelloHello Hello world o ll False False C://python37 The string str : Hello === Code Execution Successful ===</pre>

Escape Sequence

main.py	Output
<pre>1 print(''''They said, "What's there?''') 2 print('They said, "What\'s going on?') 3 print("They said, \"What's going on?\"")</pre>	<pre>'''They said, "What's there?" They said, "What's going on?" They said, "What's going on?" === Code Execution Successful ===</pre>

The format() method





main.py	Output
<pre>1 print("{} and {} both are the best friend".format("Devansh", "Abhishek")) 2 print("{1} and {0} best players".format("Virat", "Rohit")) 3 print("{a},{b},{c}".format(a = "James", b = "Peter", c = "Ricky"))</pre>	<pre>Devansh and Abhishek both are the best friend Rohit and Virat best players James,Peter,Ricky === Code Execution Successful ===</pre>

Python String Formatting Using % Operator





main.py	Output
<pre>1 Integer = 10 2 Float = 1.290 3 String = "Devansh" 4 print("Hi I am Integer ... My value is %d\nHi I am float ... My value is %f\nHi I am string ... My value is %s" % (Integer, Float, String))</pre>	<pre>Hi I am Integer ... My value is 10 Hi I am float ... My value is 1.290000 Hi I am string ... My value is Devansh === Code Execution Successful ===</pre>

Python code to show the difference between creating a list and a tuple





Python Assignment

main.py		Output
<pre>1 list_ = [4, 5, 7, 1, 7] 2 tuple_ = (4, 1, 8, 3, 9) 3 print("List is: ", list_) 4 print("Tuple is: ", tuple_)</pre>	   Share 	List is: [4, 5, 7, 1, 7] Tuple is: (4, 1, 8, 3, 9) === Code Execution Successful ===

Updating the element of list and tuple at a particular index

main.py		Output
<pre>1 list_ = ["Python", "Lists", "Tuples", "Differences"] 2 tuple_ = ("Python", "Lists", "Tuples", "Differences") 3 list_[3] = "Mutable" 4 print(list_) 5 - try: 6 tuple_[3] = "Immutable" 7 print(tuple_) 8 - except TypeError: 9 print("Tuples cannot be modified because they are immutable")</pre>	   Share 	['Python', 'Lists', 'Tuples', 'Mutable'] Tuples cannot be modified because they are immutable === Code Execution Successful ===

Code to show the difference in the size of a list and a tuple

main.py		Output
<pre>1 list_ = ["Python", "Lists", "Tuples", "Differences"] 2 tuple_ = ("Python", "Lists", "Tuples", "Differences") 3 print("Size of tuple: ", tuple_.__sizeof__()) 4 print("Size of list: ", list_.__sizeof__())</pre>	   Share 	Size of tuple: 56 Size of list: 72 === Code Execution Successful ===