Python

Write a simple program to find minimum of three numbers using conditional statements (if elif else) in python.

```
a = 14
b = 21
c = 67

smallest = 0

if a < b and a < c :
    smallest = a
elif b < a and b < c :
    smallest = b
else c < a and c < b :
    smallest = c

print(smallest, "is the smallest of three numbers.")
```

Or

```
# Input three numbers
num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))
num3 = float(input("Enter the third number: "))

# Using if-elif-else to find the minimum
if num1 <= num2 and num1 <= num3:
    min_num = num1
elif num2 <= num1 and num2 <= num3:
    min_num = num2
else:
    min_num = num3

# Display the result
print("The minimum of the three numbers is:", min_num)
```

Discuss slicing and indexing of list in python using a small code snippet.

Indexing:

- You can access individual elements in a list using square brackets and the index of the element you want. Here's an example:

```
my_list = [10, 20, 30, 40, 50]

# Accessing elements using indexing
first_element = my_list[0]
second_element = my_list[1]

print("First element:", first_element)
print("Second element:", second_element)

#out put:-
#First element: 10
#Second element: 20
```

Slicing:

- Slicing allows you to extract a portion of the list by specifying a range of indices. The syntax is list[start:stop:step]. Here's an example:

```
my_list = [10, 20, 30, 40, 50]

# Slicing to get a subset of the list
subset = my_list[1:4]  # Elements from index 1 to 3 (4 is exclusive)

print("Subset:", subset)

#output:-Subset: [20, 30, 40]
```

What is the use of “break” and “continue” in python.

- In Python, break and continue are control flow statements used within loops to modify the loop's behavior.

- ‘break ‘ statement:
- It is used to exit a loop prematurely, before its normal completion.
- When a break statement is encountered inside a loop, the loop is terminated, and the program continues with the next statement after the loop.
- Typically, break is used when a certain condition is met, and you want to exit the loop early.

```
for num in range(1, 10):
    if num == 5:
        break
    print(num)
```

Output

```
1
2
3
4
```

- ‘continue’ statement:
- It is used to skip the rest of the code inside a loop for the current iteration and move to the next iteration.
- When a continue statement is encountered, the loop jumps to the next iteration without executing the remaining code in the loop body.

```
for num in range(1, 6):
    if num == 3:
        continue
    print(num)
```

Output:-

```
1
2
4
5
```

What is mutable and immutable datatypes in python. Discuss about list and tuple.

Mutable Data Types:

- List:
  - Lists are mutable in Python, meaning you can modify their elements after the list is created.
  - You can change, add, or remove elements in a list using various methods like indexing, slicing, append, extend, pop, etc.
- Example:

```
my_list = [1, 2, 3]
my_list[0] = 10  # Modifying the list
my_list.append(4)  # Adding an element
print(my_list)  # Output: [10, 2, 3, 4]
```

Immutable Data Types:

- Tuple:
  - Tuples are immutable, meaning once a tuple is created, you cannot change the values of its elements or add/remove elements.
  - You define a tuple using parentheses (). Elements are accessed using indexing.
- Example:

```
my_tuple = (1, 2, 3)
# The following line would result in an error:
# my_tuple[0] = 10  # TypeError: 'tuple' object does not support item assignment
```

Discuss comparison operator and assignment operators.

Comparison Operators:

Comparison operators are used to compare values and return Boolean results (True or False).

Assignment Operators:

Assignment operators are used to assign values to variables.

Discuss usage of append(), remove(), insert(), extend(), len(), sort(), reverse() functions and the concatenation operation using + operator in list.

- append() Function:
  Adds an element to the end of the list.

```
my_list = [1, 2, 3]
my_list.append(4)
# Result: [1, 2, 3, 4]
```

remove() Function:

- Removes the first occurrence of a specified element from the list.

```
my_list = [1, 2, 3, 2]
my_list.remove(2)
# Result: [1, 3, 2]
```

insert() Function:

- Inserts an element at a specific position in the list.

```
my_list = [1, 2, 3]
my_list.insert(1, 4)
# Result: [1, 4, 2, 3]
```

extend() Function:

- Extends the list by appending elements from an iterable.

```
list1 = [1, 2, 3]
list2 = [4, 5, 6]
list1.extend(list2)
# Result: [1, 2, 3, 4, 5, 6]
```

len() Function:

- Returns the number of elements in the list.

```
my_list = [1, 2, 3, 4, 5]
length = len(my_list)
# Result: 5
```

sort() Function:

- Sorts the elements of the list in ascending order (in-place).

```
my_list = [3, 1, 4, 1, 5, 9, 2]
my_list.sort()
# Result: [1, 1, 2, 3, 4, 5, 9]
```

reverse() Function:

- Reverses the elements of the list (in-place).

```
my_list = [1, 2, 3, 4]
my_list.reverse()
# Result: [4, 3, 2, 1]
```

Concatenation using + Operator:

- Concatenates two lists to create a new list.

```
list1 = [1, 2, 3]
list2 = [4, 5, 6]
concatenated_list = list1 + list2
# Result: [1, 2, 3, 4, 5, 6]
```

Discuss for and while loop in python with an example. What is the usage of range() function in a for loop?

for Loop:

The for loop is typically used for iterating over a sequence (such as a list, tuple, string, or range) or other iterable objects.

```
# Using for loop to iterate over a list
fruits = ["apple", "banana", "cherry"]
for fruit in fruits:
    print(fruit)
```

Output

```
apple
banana
cherry
```

while Loop:

The while loop is used for executing a block of code repeatedly as long as the given condition is true.

```
# Using while loop to print numbers from 1 to 5
num = 1
while num <= 5:
    print(num)
    num += 1
```

Output

```
1
2
3
4
5
```

Usage of range() Function in a for Loop:

The range() function is often used in for loops to generate a sequence of numbers.

```
# Using range() in a for loop to print numbers from 1 to 5
for num in range(1, 6):
    print(num)
```

Output

```
1
```

2
3
4
5

Programs to find min, avg of list, count occurance of letter in a string using dictonary.