1. Difference b/w break continue and pass?

* Break: will terminate the loop if the condition is true.
* Example: for i in range(5)

if i == 3:

break

print(i)

output:0

1

2

* Continue: will skip the current iteration if the condition is true.

Example: for i in range(5):

if i == 3:

continue

print(i)

* Pass: The pass statement is a placeholder for future code. nothing happens in the present code.

2.d/w remove , delete, pop and write an example program in python to demonstrate 3 of them.?

* Remove: It is one of the SET method.

It removes the specific element from the list.

Example: a={1,2,3,4,5}

a.remove(2)

print(a)

output: {1,3,4,5}

* Delete: Delete is represented by “del” keyword.

It is a built-in keyword in Python.

* Pop: it removes any random element from the set.

Example: a={65,78,55,98,34}

a.pop()

print(a)

output: {98,34,55,78}

3.D/w append and extend..?

* Append: add a single element to an existing list.

It adds a new element at the end of the existing list.

num1=[34,67,54,87,98]

x=num1.append(22)

print(num1)

output:[34,67,54,87,98,22]

* Extend: It adds multiple elements to an existing list.

It adds new elements at the end of the list.

list\_one=[10,40,30,35,25]

x=list\_one.append([45,67,’ python’,98,’ ruby’,100])

print(list\_one)

output: [10,40,30,35,25,45,67,’python’,89,’ruby’,100]

4.Write a python program to print the element in the array with negative indexes(ex : print the element which is present in -2 positions) ..?

* var=[1,3,7,9,4,10]

Print(var[-5])

Output: 3

5.Write a python program to print your name, designation, technology 100times?

* details=”bhavya”, “software developer”, “python”

for i in range (100) :

print(details)

output: ('bhavya', 'software developer', 'python')

('bhavya', 'software developer', 'python')

('bhavya', 'software developer', 'python')

('bhavya', 'software developer', 'python')

('bhavya', 'software developer', 'python')

('bhavya', 'software developer', 'python')

('bhavya', 'software developer', 'python')

('bhavya', 'software developer', 'python')

('bhavya', 'software developer', 'python')

('bhavya', 'software developer', 'python')

('bhavya', 'software developer', 'python')

('bhavya', 'software developer', 'python')

('bhavya', 'software developer', 'python')

('bhavya', 'software developer', 'python')

('bhavya', 'software developer', 'python')

('bhavya', 'software developer', 'python')

('bhavya', 'software developer', 'python')

('bhavya', 'software developer', 'python')

('bhavya', 'software developer', 'python') etc.

6. Arithmetic Operators

# Write a program that takes two numbers from the user and performs the following operations:

# - Addition

* Add the two operands.

Example: first\_num=60

second\_num=90

result=first\_num+second\_num

print(result)

output: 150.

# - Subtraction

* Subtracts the second operand from the first operand.
* Example: first\_num=100

second-num=40

result=first\_num-second\_num

print(result)

Output: 60

# - Multiplication

* Multiplies to operands

Example: a=67

b=56

result=a\*b

print(result)

output: 3752.

# - Division

* Divides the first operand by the second one.

Example: a=40

b=20

result=40/20

print(result)

output: 2.0

7.Logical Operators

# Write a program that asks the user for their age.

* age=” ”

print(“my age is:”,34)

output: my age is 34.

# - If the age is less than 18, print "You are a minor."

* age=16

if age<18:

print(“you are a minor”)

output: you are a minor.

# - If the age is 18 or older, print "You are an adult."

Age=25

if age==18 or age>18:

print(“you are an adult”)

output: you are an adult.

8. Comparison Operators

# Write a program that compares two strings entered by the user.

# - If the strings are equal, print "Strings are equal."

string\_one=”python”

string\_two=”python”

if string\_one==string\_two:

print(“strings are equal”)

else:

print(“strings are not equal”)

output: strings are equal.

# - If not, print "Strings are not equal."

string\_one=”python”

string\_two=”devops”

if string\_one==string\_two:

print(“strings are equal”)

else:

if string\_one!=string\_two:

print(“strings are not equal”)

output: strings are not equal.

9.While Loop

# Write a program that uses a while loop to print the numbers from 1 to 5.

* count=1

while count <=5 :

print(count)

count +=1

output: 1

2

3

4

5.

10. For Loop

# Write a program that uses a for loop to iterate over a list of fruits and print each fruit.

* fruits=[“apple”, ”banana”, ”grapes”]

for x in fruits:

print(fruits)

output: ['apple', 'banana', 'grapes']

['apple', 'banana', 'grapes']

['apple', 'banana', 'grapes']

11.Lists

# Create a list of numbers and perform the following operations:

# - Add a new number to the list.

* list\_one=[2,5,6,9,10]

second\_list=list\_one.append(11)

print(list\_one)

Output: [2,5,6,9,10,11]

# - Remove an existing number from the list.

* list\_one=[3,4,5,6]

second\_list=list\_one.remove(5)

print(list\_one)

Output: [3,4,6]

12.Dictionaries

# Create a dictionary representing a person with attributes like name, age, and city.

# - Print the person's information.

* details={“name”: ”bhavya”,” age”: 24, “city”: ” eluru”}

print(details)

output: {'name': 'bhavya', 'age': 24, 'city': 'eluru'}

# - Add a new attribute (e.g., occupation) to the dictionary.

* details={“name”: ”bhavya”,” age”: 24, “city”: ” eluru”}

details\_two=details.update({“occupation”: “IT employee”})

print(details)

output: {'name': 'bhavya', 'age': 24, 'city': 'eluru', 'occupation': 'IT employee'}

13. What is a list in Python, and how is it used in DevOps?

* It represents an ordered collection of items.
* List is used to store multiple items in a single variable.
* Lista are mutable, which means you can modify, add or remove elements after list creation.
* Elements in a list can be of different data types.
* It allows duplicate values.
* It is identified with square braces [].

Use of list in Devops:

* Configuration management: it is used to store server configurations, environment variables or deployment settings.
* Deployment: it manages software deployments to different servers or environments
* Inventory management: it store inventory data, including names, IP address and status.
* Task automation: Devops scripts often involve iterating through lists to perform tasks on multiple servers or resources.

14.How do you create a list in Python, and can you provide an example related to DevOps?

* Create a list:

list1=[“apple”,”grapes”,”jackfruit”,”pineapple”]

print[list]

output: ['apple', 'grapes', 'jackfruit', 'pineapple']

* In devops we will use list to define the instance type, list of S3 buckets etc.

15. What is the difference between a list and a tuple in Python, and when would you choose one over the other in a DevOps context?

|  |  |
| --- | --- |
| List | Tuple |
| List is a collection of data that is ordered, indexed, and changeable. | Tuple is a collection of data ordered and unchangeable. |
| Lists are mutable, that is can be modified once created. | It is immutable that is cannot be modified once created. |
| Created using [] braces. | Created using () round braces |
| list has some methods. | Tuple has some operations. |
| In dictionary, we can’t use lists as keys. | In dictionary, we can create keys using tuples. |
| It takes more memory. | It takes less memory |
| Example: list\_one[1,2,3] | Example: my\_tuple(1,2,3,4) |

* For the server configuration, target servers and deployment nodes we can use the list.
* For admin related information,deployment steps and configuration servers stores we will use tuples.

16.How can you access elements in a list, and provide a DevOps-related example?

* Indexing is the simplest and most direct way to access specific items in a list.
* Example: list\_one=[1,4,6,3,7,9]

print(list\_one[2])

print(list\_one[5])

output: 6

9

17. How do you add an element to the end of a list in Python? Provide a DevOps example.

* My\_list=[1,2,3,4]

My\_list.append(5)

18. How can you remove an element from a list in Python, and can you provide a DevOps use case?

* My\_list=[1,2,3,4]

My\_list.remove(2)

19.Write a Python program that takes a list of numbers and prints the sum of all the elements.

* list1=[20,58,90,87,100,5679,357,86533]

print(sum(list1))

Output: 92924

20.Develop a Python program that removes duplicates from a given list and prints the unique elements.

* list1=[1,1,2,3,3,4,4,5,6,7,8]

unique\_list=list(set(list1))

print(unique\_list)

output: [1,2,3,4,5,6,7]

21.Create a Python program that takes two sets as input and prints the union of these sets (all unique elements from both sets).

* set\_one={30,50,60,80,70}

set\_two={20,30,60,50,90}

print(set\_one.union(set\_two))

output: {70, 80, 50, 20, 90, 60, 30}

22.Write a Python function that checks if two given tuples are identical.

* t1=(1,2,5,7)

t2=(3,4,6,7)

result= t1== t2

print(“are the tuples identical?”)

print(result)

output: are the tuples identical?

False

23.Implement a Python program that reads a string and counts the occurrences of each character.

* sample\_text=input(‘enter a string:’)

for i in sample\_text:

print(I,’=’,sample\_text.count(i),’times’)

output: enter a string:"hello world"

" = 2 times

h = 1 times

e = 1 times

l = 3 times

l = 3 times

o = 2 times

= 1 times

w = 1 times

o = 2 times

r = 1 times

l = 3 times

d = 1 times

" = 2 times

24.Develop a Python program that reverses a given string using slicing.

* string\_one=[“sql”,”go”,”cloud]

string\_one.reverse()

print=(string\_one)

output: ['cloud', 'go', 'sql']

25.Write a Python program to find the common elements between two lists.

* list1=[1,2,3,4,5,6]

list2=[3,5,7,9]

common\_elements=list(set(list1).intersection(list2))

output: [3,5]

Create a Python function that takes a string as input and checks if it is a palindrome.

* string1=input(“enter a string:”)

if string1 == string1[::-1]:

print(“string is a palindrome”)

else:

print(“string is not a palindrome”)

Output: enter a string:malayalam

string is a palindrome.

26.Implement a Python program that converts a given string to title case (capitalize the first letter of each word).

* a=”devops”

print(a.capitalize())

output: Devops

27.Write a Python program that reads a list of strings and sorts them in alphabetical order.

* list\_one=[“python”,”java”,”devops”,”ruby”,”php”]

list\_two=list\_one.sort()

print(list\_one)

output: ['devops', 'java', 'php', 'python', 'ruby']

28.Develop a Python program that reads a string and counts the number of vowels (a, e, i, o, u) in it.

* string1=”python is a high level programming language”

vowels=”aeiou”

count=sum(string1.count(vowel) for vowel in vowels)

print(count)

output: 13

29.Create a Python program that checks if a given string is an anagram of another string.

* string1=input(“enter a first string:”)

string2=input(“enter a second string:”)

if sorted(string1) == sorted(string2):

print(“strings are anagram”)

else:

print(“strings aren’t anagram”)

output: enter a first string: elbow

enter a second string: below

strings are anagram.

30.Write a Python function that takes a list of numbers and returns a new list with only the even numbers.

* a=[1,2,4,5,6,8,10]

for val in a:

if val % 2==0:

print(val)

output: 2

4

6

8

10

31. Develop a Python program that takes a string and converts it to uppercase.

* string\_one=”devops”

print(string\_one.upper())

Output: DEVOPS

32. Implement a Python program that reads a list of integers and prints the maximum and minimum values.

* a=[2,5,6,9,7,1]

Print(“maxmum number is:” , max(a))

Print(“minimum number is:” , min(a))

Output: maximum number is: 9

minimum number is: 1