1.Difference b/w break continue and pass ?

Continue:

* Python continue statement rejects the rest of the statements in the current iteration of the loop.
* Then the control is moved back to the top of the loop.
* Continue statements can be used in both 'for' and 'while' loops.
* Syntax: continue
* Mainly use inside condition in a loop.
* Ex:

x="Python"  
for i in x:  
 if i=='n':  
 continue  
 print(i)

* P
* y
* t
* h
* o
* Pass:
* The pass statement in python is used when a statement/ condition to be present in the code but don't want any code to be executed.
* Used the statement whwn syntactically.
* The pass will be executed and prints nothing.
* It is a null operation.
* Syntax: pass.
* Pass statement will be discarded during the byte compile phase.
* Ex:

x="Python"  
for i in x:  
 if i=='n':  
 pass  
 print(i)

* P
* y
* t
* h
* o
* n

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

Delete:

* Delete is used to delete the items in the list.
* Syntax: del
* Can delete the the individual items/ whole list.
* Items are deleted using the index number.
* We can also delete the list by using slicing method.
* f=['apple', 'banana', 'mango', 'cherry', 'berry', 'grape']  
  del f[3]  
  print(f)
* ['apple', 'banana', 'mango', 'berry', 'grape']

Remove:

* Remove is the Python list method.
* If a list contains same element more than one time, then removes the first occurence.
* It returns None value.
* If there no specified value, then it throws value Error.
* Syntax: remove()

f=['apple', 'banana', 'mango', 'cherry', 'berry', 'grape', 'banana', 'mango', 'cherry', 'berry', 'grape']  
f.remove("banana")  
print(f)

* ['apple', 'mango', 'cherry', 'berry', 'grape', 'banana', 'mango', 'cherry', 'berry', 'grape']

Pop:

* Used to remive the last element from the list.
* It also accepts the optional parameter by using the index number.
* It also prints the value that it is popped out.
* syntax: pop()

f=['apple', 'banana', 'mango', 'cherry', 'berry', 'grape']  
print(f.pop())

* grape

3.D/w append and extend..?

Append:

* Append is used to add the elements to the end of the list.
* This elements can be any datatype, list, sting, integer,etc.
* Syntax: append()
* Length increses by 1.
* Accepts single item.
* Time Complexity: O(1).

f=['apple', 'banana', 'mango', 'cherry', 'berry', 'grape']  
f.append("pineapple")  
print(f)

* ['apple', 'banana', 'mango', 'cherry', 'berry', 'grape', 'pineapple']

Extend:

* Adds the multiple items from iterable to the end of the list.
* This is list/ tuple.
* Syntax: extend
* Length increases by no.of items.
* Accepts multiple items.
* Time Complexity: O(k)

f=['apple', 'banana', 'mango', 'cherry', 'berry', 'grape']  
f1=['kiwi', 'pineapple', 'avacado']  
f.extend(f1)  
print(f)

* ['apple', 'banana', 'mango', 'cherry', 'berry', 'grape', 'kiwi', 'pineapple', 'avacado']

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) ..?

u=input("Enter array")  
a=u.split()  
n=int(input("Enter index: "))  
if -len(a)<= n <0:  
 print(f"The negative element: {a[n]}")  
else:  
 print("Invalid")

Enter array 10 20 230 67 89

Enter index: -2

The negative element: 67

Enter array 23 45 67 89

Enter index: -6

Invalid

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

d=("Chandra Lekha", "Software Developer", "Devops with AWS")  
print(d\*100)

('Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS', 'Chandra Lekha', 'Software Developer', 'Devops with AWS')

6. Arithmetic Operators

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

# - Addition

n=70  
n1=23  
s=n+n1  
print(s)

93

# - Subtraction

n=70  
n1=23  
d=n-n1  
print(d)

53

# - Multiplication

n=70  
n1=23  
p=n\*n1  
print(p)

1610

# - Division

n=70  
n1=23  
x=n/n1  
print(x)

3.0434782608695654

7.Logical Operators

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

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

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

if n<18:  
 print("You are minor")  
else:  
 print("You are an adult")

Enter your age54

You are an adult

Enter your age12

You are minor

8.Comparison Operators

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

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

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

s="Python"  
s1="Language"  
if s == s1:  
 print("Strings are equal")  
else:  
 print("String are not equal")

9.While Loop

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

i=1  
while i<6:  
 print(i)  
 i+=1

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.

f=['apple', 'banana', 'mango', 'cherry', 'berry', 'grape']  
for i in f:  
 print(i)

apple

banana

mango

cherry

berry

grape

11.Lists

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

# - Add a new number to the list.

l=["apple", "banana", " grapes", "cherry"]  
l.append("kiwi")  
print(l)

['apple', 'banana', ' grapes', 'cherry', 'kiwi']

l=["apple", "banana", " grapes", "cherry"]  
l.insert(3, "pineapple")  
print(l)

['apple', 'banana', ' grapes', 'pineapple', 'cherry']

# - Remove an existing number from the list.

l=["apple", "banana", "grapes", "cherry"]  
l.remove('banana')  
print(l)

['apple', 'grapes', 'cherry']

12.Dictionaries

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

# - Print the person's information.

d={"Name": "lekha", "Age": "23", "City": "Ponnur"}  
print(d)

{'Name': 'lekha', 'Age': '23', 'City': 'Ponnur'}

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

d={"Name": "lekha", "Age": "23", "City": "Ponnur"}  
nk="Occupation"  
nv='Software Developer'  
#The \*\* operator is used to unpack the existing dictionary into the new one.  
u={\*\*d, nk: nv}  
print(u)

{'Name': 'lekha', 'Age': '23', 'City': 'Ponnur', 'Occupation': 'Software Developer'}

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

List:

* List is an Ordered collection of items stored in a single variable.
* List have a defined ordered, order cannot be changed until explicitly changed.
* The items can be accessed by the index number and starts at 0.
* List can be mutable.
* List allows duplicates.
* List can store different data types.

How is it used in Devops:

* List can store configuration parameters, such as server IP's, environment variables, or application settings.
* This allows the consistency accross the multiple environments.
* List can be used to iterate over the servers for the deployment.This is useful for automating the tasks accross different environments.
* Lists can be useful to collect and store metrics or logs accross different platforms, makes it easier to analyze the performance.
* In CI/Cd pipelines, lists are used to track the statuses which makesv the tracking for updates are easy.
* Lists has various built-in functions.
* Lists can be used to store the multiple envirnments/servicdes.

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

l=["apple", "banana", " grapes", "cherry"]  
print(l)

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

dt=['Docker', 'Jenkins', 'Terraform', 'Kubernetes', 'Ansible']  
print(dt)

['Docker', 'Jenkins', 'Terraform', 'Kubernetes', 'Ansible']

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:

* List is an Ordered collection of items stored in a single variable.
* List have a defined ordered, order cannot be changed until explicitly changed.
* The items can be accessed by the index number and starts at 0.
* List can be mutable.
* List allows duplicates.
* List can store different data types.

TUPLE:

* It is an Oredered collection of items used to store items in a single vriable.
* Tuplle is imutable.
* U can access the items in the tuplle.
* Items in the Tuple cannot be manipulated.
* Tuple values are indexed and starts from '0'.
* Can be defined in'()'.

Both plays crucial role in Devops, We use Devops during fixed data collections that should be remain constant.

We use Devops, during the mutable collections, where data can be changed.

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

dt=['Docker', 'Jenkins', 'Terraform', 'Kubernetes', 'Ansible']  
print(dt)  
print(dt[0])  
print(dt[-1])  
print(dt[2::5])

['Docker', 'Jenkins', 'Terraform', 'Kubernetes', 'Ansible']

Docker

Ansible

['Terraform']

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

l=["apple", "banana", " grapes", "cherry"]  
l.append('berry')  
print(l)

['apple', 'banana', ' grapes', 'cherry', 'berry']

dt=['Docker', 'Jenkins', 'Terraform', 'Kubernetes', 'Ansible']  
dt.append('Helm')  
print(dt)

['Docker', 'Jenkins', 'Terraform', 'Kubernetes', 'Ansible', 'Helm']

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

l=["apple", "banana", " grapes", "cherry"]  
l.remove('cherry')  
print(l)

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

dt=['Docker', 'Jenkins', 'Terraform', 'Kubernetes', 'Ansible']  
dt.remove('Terraform')  
print(dt)

['Docker', 'Jenkins', 'Kubernetes', 'Ansible']

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

n=[1,23,45,67,89]  
t=sum(n)  
print(t)

225

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

l=["apple", "banana", " grapes", "cherry", "banana", " grapes", "cherry"]  
r=[]  
for i in l:  
 if i not in r:  
 r.append(i)  
print(r)

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

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

s={"apple", "banana", "grapes", "cherry"}  
s1={'kiwi', 'avacado', 'papaya', 'pineapple', 'apple'}  
r=s.union(s1)  
print(r)

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

s=("apple", "banana", "grapes", "cherry")  
s1=('kiwi', 'avacado', 'papaya', 'pineapple', 'apple')  
if s==s1:  
 print("Tuples are identical")  
else:  
 print("Tuples are not identical")

Tuples are not identical

s=("apple", "banana", "grapes", "cherry")  
s1=("apple", "banana", "grapes", "cherry")  
if s==s1:  
 print("Tuples are identical")  
else:  
 print("Tuples are not identical")

Tuples are identical

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

u= input("Enter string: ")  
cc={}  
for char in u:  
 if char in cc:  
 cc[char]+=1  
 else:  
 cc[char]=1  
print("Character Occurence of each character:")  
for char,count in cc.items():  
 print(f"'{char}': {count}")

'c': 1

'h': 2

'a': 3

'n': 1

'd': 1

'r': 1

'l': 1

'e': 1

'k': 1

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

s='chandra lekha'  
print(s[::-1])

ahkel ardnahc

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

l=["apple", "banana", " grapes", "cherry", "banana", " grapes", "cherry"]  
l1=["apple", "banana", "grapes", "cherry"]  
l2=l and l1  
print(l2)

["apple", "banana", "grapes", "cherry"]

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

l="KANAK"  
r=reversed(l)  
if list(l) == list(r):  
 print("TRUE")  
else:  
 print("FALSE")

TRUE

l="PYTHON"  
r=reversed(l)  
if list(l) == list(r):  
 print("TRUE")  
else:  
 print("FALSE")

FALSE

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

l="python is a high Level language"  
print(l.title())

Python Is A High Level Language

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

l=["Kiwi","Apple", "Banana", "Pineapple", "Grapes","Cherry"]  
l.sort()  
print(l)

['Apple', 'Banana', 'Cherry', 'Grapes', 'Kiwi', 'Pineapple']

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

l="Python is a High Level Language"  
c=0  
for i in range(len(l)):  
 if(  
 (l[i] == 'a')  
 or (l[i] == 'e')  
 or (l[i] == 'i')  
 or (l[i] == 'o')  
 or (l[i] == 'u')  
 or (l[i] == 'A')  
 or (l[i] == 'E')  
 or (l[i] == 'I')  
 or (l[i] == 'O')  
 or (l[i] == 'U')  
 ):  
 c += 1  
print(c)

10

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

s1="eat"  
s2="tea"  
a=[s1[i] for i in range(0,len(s1))]  
a.sort()  
b=[s2[i] for i in range(0,len(s2))]  
b.sort()  
if(not(a != b)):  
 print("True")  
else:  
 print("False")

TRUE

s1="lekha"  
s2="tea"  
a=[s1[i] for i in range(0,len(s1))]  
a.sort()  
b=[s2[i] for i in range(0,len(s2))]  
b.sort()  
if(not(a != b)):  
 print("True")  
else:  
 print("False")

False

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

d=[1,3,2,4,5,6,7,8,9,0]  
n=[]  
for i in d:  
 if i%2==0:  
 n.append(i)  
print(n)

[2, 4, 6, 8, 0]

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

l="List allows Duplicate values"  
print(l.upper())

LIST ALLOWS DUPLICATE VALUES

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

k=[-10,-20,56,78,90,34,56,78,99]  
x=min(k)  
y=max(k)  
print(x)  
print(y)

-20

99