Loops in Python are a fundamental concept that allows you to execute a block of code repeatedly based on a condition or over a sequence. There are two main types of loops in Python: for loops and while loops.

```
# A for loop is used to iterate over a sequence (such as a list, tuple, dictionary, set, or string) or any other iterable object
# In Python, there is “for in” loop which is similar to foreach loop in other languages.
# It can be used to iterate over a range and iterators.

string1 = "Good"
for i in string1:
  print(i)

# A list is an ordered collection, and you can iterate over its elements in the order they appear.
list1 = [1,"Govind",2,2.8]
for i in list1:
  print(i)

# A tuple is similar to a list but is immutable (its elements cannot be changed). You can iterate over a tuple just like a list.
tuple1 = (1,2,2.8,"Kite")
for i in tuple1:
  print(i)

# A set is an unordered collection of unique elements. When you iterate over a set, the order of elements is not guaranteed.
set1 = {1,"Fly",2.5}
for i in set1:
  print(i)

# Dictionaries store key-value pairs. When you iterate over a dictionary, you can choose to loop over the keys, values, or both.
dict1 = {'a':1,'b':2,'c':3}
for i in dict1:
  print(i)

dict1 = {'a':1,'b':2,'c':3}
for i in dict1.keys():
  print(i)

dict1 = {'a':1,'b':2,'c':3}
for i in dict1.values():
  print(i)

dict1 = {'a':1,'b':2,'c':3}
for i,j in dict1.items():
  print(i,j)
```

```
G
o
o
d
1
Govind
2
2.8
1
2
2.8
Kite
1
2.5
Fly
a
b
c
a
b
c
1
2
3
a 1
b 2
c 3
```

```
# The range() function can be called with one, two, or three arguments:

# range(stop)
# range(start, stop)
# range(start, stop, step)
```

```
# When you pass a single argument to range(), it generates numbers starting from 0 up to (but not including) the stop value.
for i in range(5):
    print(i)

# With two arguments, range() generates numbers starting from the start value and stops just before the stop value.
for i in range(2,7):
  print(i)

# With three arguments, range() allows you to specify a step value, which determines the difference between each number in the
for i in range(0,10,2):
  print(i)

# You can use a negative step value to generate a sequence in reverse order. Start value here should be be greater than stop va
for i in range(10,0,-2):
  print(i)

# If the start value is greater than or equal to the stop value and the step is positive (or vice versa for negative step), the
```

```
0
1
2
3
4
2
3
4
5
6
0
2
4
6
8
10
8
6
4
2
```

```
# range() is often used in list comprehensions to generate sequences or to create a list from its output.

cube = [i*i*i for i in range(1,6)]
print(cube)
type(cube)

list2 = list(range(5))
print(list2)
```

```
[1, 8, 27, 64, 125]
[0, 1, 2, 3, 4]
```

```
# You can also nest for loops to iterate over multi-dimensional sequences, such as lists of lists.

matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
for row in matrix:
    for element in row:
        print(element, end=" ")
    print()
```

```
1 2 3
4 5 6
7 8 9
```

```
# else with for

for i in range(1, 4):
    print(i)
else:
    print("Loop completed")
```

```
1
2
3
Loop completed
```

```
# While Loop :- A while loop repeatedly executes a block of code as long as a given condition is true.
```

```
count = 1
while count <=3:
  print("Iteration = ",count)
  count = count+1;
```

```
Iteration =  1
Iteration =  2
Iteration =  3
```

```
# If the condition never becomes false, the while loop will continue indefinitely, creating an infinite loop.
# To avoid this, ensure that the loop has a way to terminate.
```

```
# Using else statement with While Loop in Python
# The else clause is only executed when your while condition becomes false.
# If you break out of the loop, or if an exception is raised, it won't be executed.

count = 1
while count <=3:
  print("Iteration = ",count)
  count = count+1;
else:
  print("Loop Terminated")
```

```
Iteration =  1
Iteration =  2
Iteration =  3
Loop Terminated
```

```
# while True:
#     print("This will run forever!")

#Warning: This is an infinite loop. Use Ctrl + C to stop execution.
```

```
password = ""
while password != "govind":
    password = input("Enter the password: ")
print("Access granted!")
```

```
Enter the password: govind
Access granted!
```

```
#nested for and while loops
for i in range(1,5):
  for j in range(i):
    print(i,end=" ")
  print()

print()
a=1
b=1
while(a!=10):
  while(b!=10):
    print(a*b,end=" ")
    b=b+1
  a=a+1
  b=1
  print()
  print()
```

```
1
2 2
3 3 3
4 4 4 4

1 2 3 4 5 6 7 8 9

2 4 6 8 10 12 14 16 18

3 6 9 12 15 18 21 24 27

4 8 12 16 20 24 28 32 36

5 10 15 20 25 30 35 40 45

6 12 18 24 30 36 42 48 54
```

```
7 14 21 28 35 42 49 56 63

8 16 24 32 40 48 56 64 72

9 18 27 36 45 54 63 72 81
```

```
for i in range(5):
  pass
```

```
for i in range(5):
  if i == 3:
    continue
  print(i)
```

```
0
1
2
4
```

```
for i in range(5):
  if i == 3:
    break;
  print(i)
```

```
0
1
2
```

```
for i in range(5):
  for j in range(i):
    print(i, end =" ")
  print()
```

```

1
2 2
3 3 3
4 4 4 4
```