Loop Statements (for,while)

Statements	Syntax	Example	Meaning
while	while (Condition): statement(s)	count = 0 while (count < 3): count = count+1 print("Hello Bennettians") **Output:** Hello Bennettians Hello Bennettians Hello Bennettians	while loop is used for iterators
for	for iterator_var in sequence: statements(s)	l = ["bennett", "for", "bennettians"] for i in l: print(i) **Output:** bennett for bennettians	for can be used to iterate over iterators and a range.
range	for iterator_var in range(n):	for x in range(4): print(x) **output:** 0	It returns a sequence of numbers, starting from 0 by default, and increments by 1

		1 2 3	(by default), and ends at a specified number.

nested-for	for iterator_var in sequence:
	for iterator_var in sequence:
	statements(s)
	statements(s)
		for i in range(1, 5):
		for j in range(i):
		print(i, end=' ')
		print( )
		Output:
		1
		2 2
		3 3 3
		4 4 4 4
			Python programming language allows to use for loop inside another for loop.

nested-while	while expression:
	while expression:
	statement(s)
	statement(s)
		i = 1
		j = 5
		while i < 4:
		while j < 8:
		print(i, ",", j)
		j = j + 1
		i = i + 1
		Output:
		1 , 5
			Python programming language allows to use while loop inside another while loop.

		2 , 6 3 , 7	

Else in for loop	for iterator_var in sequence:
	statements(s)
	else
	statements
		for x in range(6):
		print(x)
		else:
		print("Finally finished!")
		Output:
		0
		1
		2
		3
		4
		5
		Finally finished!
			The else keyword in a for loop specifies a block of code to be executed when the loop is finished:

Else in while loop	While condition:
	statements(s)
	else
	statements
		x=0
		y=6
		while y>x:
		print(x)
		x=x+1
		else:
		print("Finally finished!")Output:
		0
		1
		2
		3
		4
		5
		Finally finished!
			The else keyword in a while loop specifies a block of code to be executed when the loop is finished:

Control Statements (Continue,Break,Pass)

Statements	Example	Meaning
continue	for char in 'Pythn': if (char == 'y'):	When the program encounters continue statement, it will skip the

	continue print("Current character: ", char) **Output:** Current character: P Current character: t Current character: h Current character: n	statements which are present after the continue statement inside the loop and proceed with the next iterations.

break	for char in 'Python':
	if (char == 'h'):
	break
	print("Current character: ", char)
	
	Output:
	Current character: P
	Current character: y
	Current character: t
		The break statement is used to terminate the loop containing it, the control of the program will come out of that loop.

pass	for char in 'Python':
	if (char == 'h'):
	pass
	print("Current character: ", char)
	
	Output:
	Current character: P
	Current character: y
	Current character: t
	Current character: h
	Current character: o
	Current character: n
		Pass statement is python is a null operation, which is used when the statement is required syntactically.

1. **Predict the output:**

```
j=1
while j<=10:
   print(j)
   j=j+1
```

Output:

1
2
3
4
5
6
7
8
9
10

2. **Predict the output:**

```
num = 10
while num > 6:
   print(num)
   num = num-1
else:
   print("loop is finished")
```

Output:

10
9
8
7
loop is finished

3. **Predict the output**

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
print(x)
```

Output:

apple
banana

cherry

4. **Predict the output:**

```
for x in "apple":
  print(x)
```

Output:

a
p
p
l
e

5. **Predict the output:**

```
batch = ["eb10", "eb12", "eb14"]
for x in batch:
   print(x)
   if x == "eb12":
      break
```

Output:

eb10
eb12

6. **Predict the output:**

```
batch = ["eb20", "eb21", "eb22"]
for x in batch:
  if x == "eb21":
    continue
  print(x)
```

Output:

eb20
eb22

7. **Predict the output:**

```
for x in range(2, 6, 2):
  print(x)
```

Output:
2

4

8. A series has been provided (1/1! + 2/2! + 3/3! + …), calculate the sum of first 5 numbers of the series using While loop.

Solution:

```
x=1
sum=0
while(x<=5):
 j=x
 fact=1
 while(j>1):
  fact=fact*j
  j=j-1;
 sum = sum + x/fact
 x=x+1
print("Summation of series is: ", sum)
```

Output: Minimum value is 2.708333333333333

9. Print the following pattern using for loop

```
                1
          2           3
    4           5           6
7           8           9           10
```

Solution:

```
num=1
for i in range(4):
 for j in range(3,i,-1):
  print(" ", end='')
 for j in range(i+1):
  print(num,end=' ')
  num=num+1
 print("")
```

10. Calculate the summation of first 10 numbers using while loop, where x is a user input and value is 2.

$$\frac{x-1}{x}+\frac{1}{2}\left(\frac{x-1}{x}\right)^2+\frac{1}{3}\left(\frac{x-1}{x}\right)^3+\frac{1}{4}\left(\frac{x-1}{x}\right)^4+\cdots$$

Solution:

```
x=1
in_var = 2
in_var=(in_var-1)/in_var
sum=0
for x in range(1,10):
   sum = sum + ((in_var)**x)/x
   x=x+1
print("Summation of ", sum)
```

11. Create a list of 10 elements and check whether a number is available in the list using for loop.

    Output:

```
list=[1,2,3,7,9,4,56,23,12,3]
#value=7
value=int(input())
flag=0
for x in range(1, 10):
  if value==list[x]:
    print("number found at", x)
    flag=1
if flag==0:
  print("number not found")
```

12. Create a list of 10 elements and find out the minimum element using while loop.

    Output:

```
list=[10,2,3,7,9,4,56,23,12,3]
min_value=list[0]
x=0
while(x<10):
 if min_value>list[x]:
   min_value = list[x]
 x=x+1
print("Minimum value is", min_value)
```

13. Enter the string of your name and print the ASCII value of it.

    Solution:

```
print("Enter Name:", end="")
name = input()
```

```
namelength = len(name)
for char in name:
    ascii = ord(char)
    print(char, "\t", ascii)
```