Black Origin

Loops in Python

Loops are used for sequential traversal. For traversing list, string, tuples etc.

for Loops

```
for el in list:
        #some work
```

```
list= [1,2,3]
for el in list:
    print(el)
```

Black Origin

Loops in Python

for Loop with else
for el in list:
#some work
else:
#work when loop ends

```
list= [1,2,3]
for el in list:
    print(el)
else:
    print("END")
```

Black Origin

Let's Practice

using for

Print the elements of the following list using a loop:

[1, 4, 9, 16, 25, 36, 49, 64, 81,100]

Black Origin

Let's Practice

using for

Search for a number x in this tuple using loop:

(3, 6, 9, 12, 15, 18, 21, 24, 27, 30)

Black Origin

range( )

Range functions returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and stops before a specified number.

range(start?, stop, step?)

```
for el in range(5):
    print(el)
for el in range(1,5):
    print(el)
for el in range(1,5,2):
    print(el)
```

Black Origin

Let’s Practice

Write a program using for and range() to print all odd numbers between 1 and 50.
(Hint: Start from 1, step by 2)

Black Origin

Let’s Practice

Write a program to print numbers starting from 10 up to 100, increasing by 10 each time.

Black Origin

Let’s Practice

Write a program using for and range() to print numbers from 50 down to 1, decreasing by 5 each time.

Black Origin

Nested For loop

A nested loop is a loop inside a loop.
The "inner loop" will be executed one time for each iteration of the "outer loop":

pass Statement

pass is a null statement that does nothing. It is used as a placeholder for future code.

```
for el in range(10):
    pass
```

Black Origin

Let’s Practice

Write a program using a while loop to find the sum of the first n natural numbers, where n is input by the user.

Black Origin

Let's Practice

Write a program using a for loop to calculate the factorial of a given number n.

Functions in Python

Block of statements that perform a specific task.

```
def func_name( parameter1, parameter2..) :
        #some work
        return val

func_name( arg1, arg2 ..) #function call
```

← Function Definition

```
def sum(a,b):  1 usage
    s = a+b
    return s
print(sum( a: 2,  b: 3))
```