# LOOPS

for-loop

for-loop

i =0  
while (i <= 5): #condition  
 print ("Hello world")  
 i = i+1

Hello world  
Hello world  
Hello world  
Hello world  
Hello world  
Hello world

i =0  
while (i <4): #condition  
 print ("Hello world")  
 i = i+1

Hello world  
Hello world  
Hello world  
Hello world

i=6  
while (i >=1):  
 print ("Hello Geek")  
 i=i-1  
print("i am out")

Hello Geek  
Hello Geek  
Hello Geek  
Hello Geek  
Hello Geek  
Hello Geek  
i am out

3 in [0,1,2,3,4]

True

5 in [0,1,2,3,4]

False

for x in "java":  
 print (x)

j  
a  
v  
a

s="python"  
for x in s:  
 print (x)

p  
y  
t  
h  
o  
n

for x in 10:  
 print(x)

---------------------------------------------------------------------------  
  
TypeError Traceback (most recent call last)  
  
<ipython-input-46-8a1f11565974> in <module>  
----> 1 for x in 10:  
 2 print(x)  
  
  
TypeError: 'int' object is not iterable

for x in '10':  
 print(x)

1  
0

for i in "java":  
 print (i +" python")

j python  
a python  
v python  
a python

words = ['Linux', 'window', 'defenestrate']  
for y in words:  
 print (y)

Linux  
window  
defenestrate

words = ['Linux', 'window', 'defenestrate']  
for y in words:  
 print (y+" Python")

Linux Python  
window Python  
defenestrate Python

for x in ['Linux', 'window', 'defenestrate']:  
 print( x,len(x))

Linux 5  
window 6  
defenestrate 12

#Python 2.7   
print(range(10))  
# [0,1,2,3,4,5,6,7,8,9]  
  
# for i in xrange(10):  
# print i  
  
# Python 3  
# No xrange  
# and Xrange of python 2.7 is conerted to range  
# print (range(20))  
  
# print range(2,20,3)-->2,5,8,11,14,17  
# print range(-10, -100, -30) ----> [-10,-40,-70]

for i in range(10):  
 print (i)

0  
1  
2  
3  
4  
5  
6  
7  
8  
9

for i in range(2,10):  
 print (i)

2  
3  
4  
5  
6  
7  
8  
9

for i in range(5):  
 print ("python")

python  
python  
python  
python  
python

for i in range(2,10):#[2,3,4,5,6,7,8,9]  
 print (i)  
print("Here")

2  
3  
4  
5  
6  
7  
8  
9  
Here

a = ['Mary', 'had', 'xyz', 'little', 'lamb']  
for i in [0,1,2,3,4]:  
 print (i, a[i])

0 Mary  
1 had  
2 xyz  
3 little  
4 lamb

a = ['Mary', 'had', 'xyz', 'little', 'lamb']  
for i in range(5):# for i in [0,1,2,3,4,]  
 print (i, a[i])

0 Mary  
1 had  
2 xyz  
3 little  
4 lamb

a = ['Marry', 'had', 'xyz', 'little', 'lamb']  
for i in range(len(a)):  
 print (i, a[i],len(a[i]))

0 Marry 5  
1 had 3  
2 xyz 3  
3 little 6  
4 lamb 4

n=0  
for x in range(2, 5):#[2,3,4]  
 print( "value of n" , n)  
 print( "value of x" , x)  
 print("\*"\*15)

value of n 0  
value of x 2  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
value of n 0  
value of x 3  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
value of n 0  
value of x 4  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

n=1  
for x in range(2, 5):#[2,3,4]  
 print( "value of n" , n)  
 print( "value of x" , x)  
 print("\*"\*15)

value of n 1  
value of x 2  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
value of n 1  
value of x 3  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
value of n 1  
value of x 4  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

n=2  
for x in range(2, 5):#[2,3]  
 print( "value of n" , n)  
 print( "value of x" , x)  
 print("\*"\*15)

value of n 2  
value of x 2  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
value of n 2  
value of x 3  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
value of n 2  
value of x 4  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

for n in range(2, 5):#n=3  
 for x in range(2, 5):  
 print( "value of n" , n)  
 print( "value of x" , x)  
 print("\*"\*15)

value of n 2  
value of x 2  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
value of n 2  
value of x 3  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
value of n 2  
value of x 4  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
value of n 3  
value of x 2  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
value of n 3  
value of x 3  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
value of n 3  
value of x 4  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
value of n 4  
value of x 2  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
value of n 4  
value of x 3  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
value of n 4  
value of x 4  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

words=['Mary', 'had', 'xyz', 'little', 'lamb']  
for y in words:  
 for z in y:  
 print (z)

M  
a  
r  
y  
h  
a  
d  
x  
y  
z  
l  
i  
t  
t  
l  
e  
l  
a  
m  
b

words=['Mary', 'had', 'xyz', 'little', 'lamb']  
for z in words:  
 print (z)

Mary  
had  
xyz  
little  
lamb

### break statement

When a break statement executes inside a loop, control flow “breaks” out of the loop immediately: The loop conditional will not be evaluated after the break statement is executed. Note that break statements are only allowed inside loops, syntactically. A break statement inside a function cannot be used to terminate loops that called that function.

for i in (0, 1, 2, 3, 4,5):  
 print(i)  
 if i == 3:  
 break  
print ("outside loop")

0  
1  
2  
3  
outside loop

for i in (0, 1, 2, 3, 4):  
 print(i)  
 break  
print ("outside loop")

0  
outside loop

for i in (0, 1, 2, 3, 4):  
 if i == 2:  
 break  
 print(i)  
print ("outside loop")

0  
1  
outside loop

for letter in 'geeksforgeeks':  
 if letter == 'k':  
 break  
 print ('Current Letter :', letter)  
print ("outside loop")  
print ('Current Letter :', letter)

Current Letter : g  
Current Letter : e  
Current Letter : e  
outside loop  
Current Letter : k

for letter in 'geeksforgeeks':  
 if letter == 'g':  
 break   
 print ('Current Letter :', letter)  
print ("outside loop")

outside loop

### continue statement

A continue statement will skip to the next iteration of the loop bypassing the rest of the current block but continuing the loop. As with break, continue can only appear inside loops:

for i in (0, 1, 2, 3, 4, 5):  
 if i == 2 or i == 4:  
 continue  
 print(i)  
 print("Python")  
 print("Learning")  
print ("outside loop")

0  
Python  
Learning  
1  
Python  
Learning  
3  
Python  
Learning  
5  
Python  
Learning  
outside loop

for letter in 'geeksf':  
 print ('Before continue :', letter)  
 if letter == 'k' or letter == 's':  
 continue  
 print ('Current Letter :', letter)  
  
print ("outside loop")

Before continue : g  
Current Letter : g  
Before continue : e  
Current Letter : e  
Before continue : e  
Current Letter : e  
Before continue : k  
Before continue : s  
Before continue : f  
Current Letter : f  
outside loop

for letter in 'geeksforgeeks':  
# print("hello")  
 pass  
print (letter)

hello  
hello  
hello  
hello  
hello  
hello  
hello  
hello  
hello  
hello  
hello  
hello  
hello  
s

#### The Pass Statement

pass is a null statement for when a statement is required by Python syntax (such as within the body of a for or while loop), but no action is required or desired by the programmer. This can be useful as a placeholder for code that is yet to be written.

print (x)

The for and while compound statements (loops) can optionally have an else clause (in practice, this usage is fairly rare). The else clause only executes after a for loop terminates by iterating to completion, or after a while loop terminates by its conditional expression becoming false.

for i in range(3):  
 print("enter password")  
else:  
 print('contact to admin')

enter password  
enter password  
enter password  
contact to admin

The else clause does not execute if the loop terminates some other way (through a break statement or by raising an exception):

for i in range(2):  
 print(i)  
 if i == 1:  
 break  
 else:  
 print("inside for")  
else:  
 print('done')  
print("i am after else")

0  
inside for  
1  
i am after else

for i in range(2):  
 print(i)  
 if i == -1:  
 break  
else:  
 print('done')  
 print("i am after else")  
print('outside')

0  
1  
done  
i am after else  
outside

for i in range(2):  
 print(i)  
else:  
 print('done')  
 print("i am after for")  
print('outside')

0  
1  
done  
i am after for  
outside

for x in range(10,0,-1):  
 print (x, 'little monkeys jumping on the bed, 1 fell off and bumped his head, momma called the doctor and the doctor said, no more monkeys jumping on the bed')

from random import randint  
x = randint(1,6)  
print("dice roll:")  
print(x)

from random import randint  
roll=randint(1, 6)  
print(roll)  
if roll < 3 :  
 print("You won")  
 print(roll)  
else:  
 print("You lose")

from turtle import \*  
from freegames import line  
  
def grid():  
 "Draw tic-tac-toe grid."  
 line(-67, 200, -67, -200)  
 line(67, 200, 67, -200)  
 line(-200, -67, 200, -67)  
 line(-200, 67, 200, 67)  
  
def drawx(x, y):  
 "Draw X player."  
 line(x, y, x + 133, y + 133)  
 line(x, y + 133, x + 133, y)  
  
def drawo(x, y):  
 "Draw O player."  
 up()  
 goto(x + 67, y + 5)  
 down()  
 circle(62)  
  
def floor(value):  
 "Round value down to grid with square size 133."  
 return ((value + 200) // 133) \* 133 - 200  
  
state = {'player': 0}  
players = [drawx, drawo]  
  
def tap(x, y):  
 "Draw X or O in tapped square."  
 x = floor(x)  
 y = floor(y)  
 player = state['player']  
 draw = players[player]  
 draw(x, y)  
 update()  
 state['player'] = not player  
  
setup(420, 420, 370, 0)  
hideturtle()  
tracer(False)  
grid()  
update()  
onscreenclick(tap)  
done()

If you want to loop though both the elements of a list and have an index for the elements as well, you can use Python’s enumerate function:

## Assignment

1. Write a program which will find all such numbers which are divisible by 7 but are not a multiple of 5, between 2000 and 3200 (both included).
2. The numbers obtained should be printed in a comma-separated sequence on a single line. Hints: Consider use range(#begin, #end) method

l=[]  
for i in range(2000, 3201):  
 if (i%7==0) and (i%5!=0):  
 l.append(str(i))  
  
print ','.join(l)

for i in range(0, 5):  
 # inner loop to handle number of columns  
 # values changing acc. to outer loop  
 for j in range(0, i+1):  
 # printing stars  
 print("\* ",end="")  
 # ending line after each row  
 print("\r")

# i = 1  
# while i != 6:  
# print("\*"\*i)  
# i = i + 1  
   
i = 1  
x = int(input("enter how many starts you want "))  
for i in range(1,x+1):  
 print(" "\*(x-i) + "\*"\*i)

i = 1  
x = int(input("enter how many starts you want "))  
for i in range(1,x+1):  
 print("\*"\*i)

i = 1  
x = int(input("enter how many starts you want "))\*2  
for i in range(i,int(x/2) + 1):  
 print(" "\*x + "\* "\*i)  
x = x -1

x = int(input("Enter a number "))  
i = 1  
r = 1  
q = str()  
for i in range(1,x+1):  
while r <= i:  
r = str(r)  
q = str(q) + r  
r = int(r) + 1  
print(q)  
q = ""  
r = int(1)  
i = i + 1

Program to calculate the factorial  
5  
fact(5)= 5\*4\*3\*2\*1  
  
4