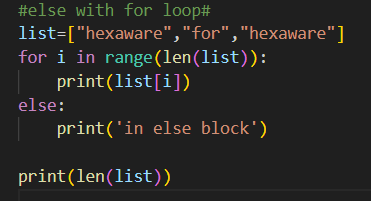
Name: K. Praveen kumar

Python Assignment-3(14-12-23)

**Using else with for loop:**

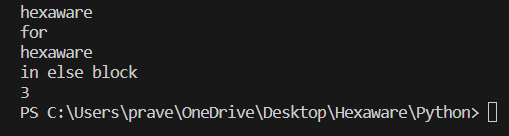
We can combine else statement with for loop like in while loop. But as there is no condition in for loop based on which the execution will terminate so the else block will be executed.

Example:



In the above example the list is created and the for loop will iterate until the condition is false and it enters into the else block. After the execution of else block the print statement which is outside the else condition will be executed.

**Output:**

****

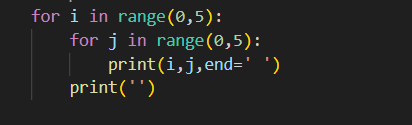
**Nested Loop:**

The loop inside loop is said as nested loop.

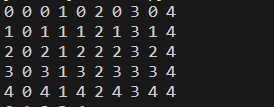
In each iteration of the outer loop, inner loop executes all its iteration. For each iteration of an outer loop the inner loop restart and completes its execution before the outer loop can continue to its next iteration.

For/while can be used in nested loops.

**Example:**

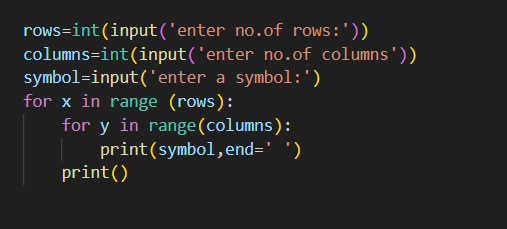


**Output:**

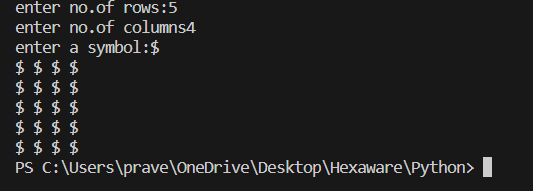


Let us consider another example and print a pattern using nested loops.

**Example:**

****

**Output:**



**Loop Control statements:**

* **Break Statement**
* **Continue Statement**
* **Pass Statement**

**1.Break Statement:**

The Break statement is used to terminate the loop or statement in which it is present.

**Syntax:**

For/while loop:

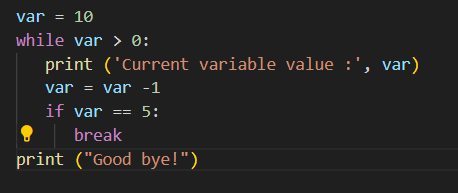
Statement(s)

If condition:

Break

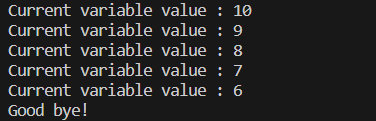
#Statement(s)

**Example:**

****

**Output:**

when the condition meets in the if block then the entire flow of execution will break and it comes out of the loop and the next statement will be printed.

****

**2.Continue Statement:**

It is also a control loop statement just like break statement which is opposite to break, instead of terminating the loop, it forces to execute the next iteration.

**Syntax:**

For/while loop:

Statement(s)

If condition:

continue

#Statement(s)

**Example:**

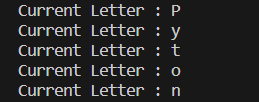
for letter in 'Python':

   if letter == 'h':

      continue

   print( 'Current Letter :', letter)

**Output:**



**3.Pass Statement:**

It is used when you don’t want any command or code to execute.

It is also used for empty control statements, Functions and Classes.

**Syntax:**

For/while loop:

Statement(s)

If condition:

pass

#Statement(s)

**Example:**

#pass

s="hexware"

for i in s:

    pass

def fun():

    pass

for i in s:

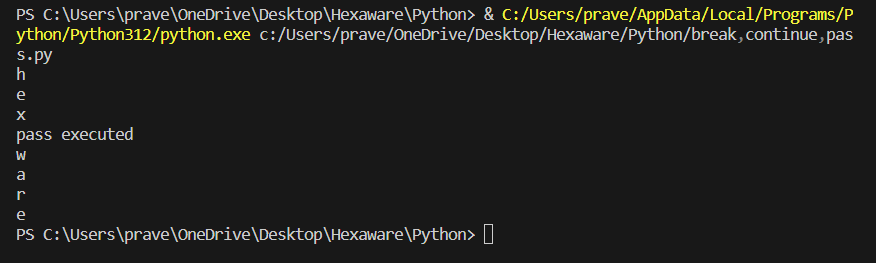
    if i=='w':

        print('pass executed')

        pass

    print(i)

**Output:**



**List slicing:**

The format for list slicing is of List Slicing is as follows:

List**[**Initial **:** End **:**IndexJump**]**

**Positive indexes:**

In the case of positive indexing, the first element will have the index number 0 and the last element has the index number as N-1.

N is the size of the list.

**Negative indexes:**

In the case of Negative indexing, the last element will have the index number -1 and the first element has the index number as -N.

N is the size of the list.

Below are some of the slicing operations performed to a list:

list=[10,20,30,40,50,60,70,80,90]

print('original list:',list)

print(list[::2])

print(list[:8:2])

print(list[2:9:3])

print(list[::-1])

print(list[:-2:-2])

print(list[:1:-1])

print(list[:2:-2])

newlist=list[2:6:1]+list[:8:2]

print("newlist:",newlist)

list[2:4]=["hex",'for','hex','!']

print("modified list:\n",list)

Output:

