

```
In [ ]: #Agenda of the Day:
        1. Continue to Nested Loops:
        2. Break and Continue Statements
        3. Functions in Python
        Problem Solving.
```

```
In [ ]: #Nested Loops?
        What is the nested loop?
        - its refers we can put the any type of loop inside the
          any other loop. (loops in another loop.)
        #How does the nested Loops works in python?
        (i) first its encounters the outer loop, excutes the first iteration
        (ii) then program returns back to top of outer loop completing the
            second iteration and again triggering the nested loop.
```

```
In [ ]: #Why we use nested Loops in python?
        - we nested the loops in python that enables a developer/programmer
          to slove even more complex problems.
```

```
In [ ]: #Nested for Loops:
        #syntax:
        for [1st iterating variable] in [outer loop]:
            [do some statements]
            for [second iterating variable] in [nested loop]:
                [do something]
```

```
In [3]: #Example:
        for i in range(3): #outer Loop
            for j in range(i+1): #1st inner Loop
                for k in range(j+1): #2nd inner Loop
                    print(i,j,k)
```

```
0 0 0
1 0 0
1 1 0
1 1 1
2 0 0
2 1 0
2 1 1
2 2 0
2 2 1
2 2 2
```

```
In [13]: #Example: (pattern forming using nested Loops)
        for i in range(5): #outer Loop
            for j in range(i): #1st Level inner Loop
                print(i, end=" ")
            print("\n") #for printing new Line.
```

```
1
```

```
2 2
```

3 3 3

4 4 4 4

In [8]:

```
#Example:
students = ["sagar", "navven", "santhosh", "surya"]
branches = ["CSE", "ECE", "EEE", "IT"]
for student in students:
    for branch in branches:
        print(student + " studying in " + branch)
```

```
sagar studying in CSE
sagar studying in ECE
sagar studying in EEE
sagar studying in IT
navven studying in CSE
navven studying in ECE
navven studying in EEE
navven studying in IT
santhosh studying in CSE
santhosh studying in ECE
santhosh studying in EEE
santhosh studying in IT
surya studying in CSE
surya studying in ECE
surya studying in EEE
surya studying in IT
```

In []:

```
#Nested While Loops?
#Syntax:
while condition:
    #code
    while condition:
        #piece of code
```

In [17]:

```
#Example: (Print stars in pattern)
i = 0
j = 0
while i <= 5:    #outer loop runs n times
    while j < i:    #inner loop runs in i times
        print("*", end=" ")
        j += 1 #or j = j+1 #increment before inner loop condition
    j = 0    #re-initialize after leaving inner loop
    i += 1    #increment before checking outer loop condition
    print("")
```

```
*
* *
* * *
* * * *
* * * * *
```

In [1]:

```
#example:
i = 1
j = 5
while i <= 4:
    while j <= 8:
        print(i, ", ", j)
```

```
j+=1  
i+=1
```

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

Break and Continue Statements , pass statements

Break :

Its used to exit the for loop permanenatly

its used to break the for loop when we meet a specific condition.

In [23]:

```
#Example:  
li = [1,2,3,4,5,6]  
n = 3  
found = False  
for num in li:  
    if n ==num:  
        found = True  
        break  
else:  
    print("for loop is terminated")  
print(num)
```

3

Continue:

we can use continue statements inside a for loop to skip the execution of the for loop body for a specific condition.

In [41]:

```
#Example: (To skip the negative numbers iteration for sum)  
nums = [1,2,-3,4,-5,6,-8,9,-10]  
sums= 0  
for n in nums:  
    if n <0:  
        continue #skips the iterations of all negative numbers.  
    sums = sums+n  
print(sums)
```

22

```
In [ ]:
#pass statement:
#Its just like a null statement
#python intrepreater ignore the comments but pass is not ignored
# Its like No Operation.(NOP.)

for (ini,condition,incre/decre)
{
    print()
}

#Syntax:
for i in []
```

```
In [48]:
#Example:
#Li = [10,20,30,40,50,60,70,80,90,100]

for i in range(1,11):
    print(i,end=" ")
```

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

```
In [ ]:
#Functions in Python:
    What is the function?
        - Its a Group of related statements that performs a specific task.
#Why we use functions?
    - To Avoid code repition.
    - To make our complex program into smaller sub programs for easy debug.
    - write once and use any times .
    - Code resuability.
```

```
In [ ]:
#How we can create functions in python?
    in Python we use def to declare the functions.
```

```
In [39]:
import keyword
print(len(keyword.kwlist),end=" ")
```

```
35
```

```
In [ ]:
#Syntax for function declaration:
    #function declaration
def funtionname(parameters):    #formal parameters
    """doc string"""    #this function used to find out even number

    #block of statements
    return

funtionname(parameters)    #function calling #actual parameters
```