Lecture 6 – For loops in python

Hi guys, thanks for staying connected. I am Akshansh (+91, 8384891269, akshanshofficial@gmail.com) and we will be talking about the iteration here. We've already talked about the while loop in last lecture notes. While I was talking about the "while" loops, I told you about the for loop. Here is the time to cover this topic. Let's get started, pythonistas!

FOR LOOPS ITERATION -

Iteration, you know it. It is like searching a girlfriend in Facebook, you have to go to each account. "Accessing each item/element in a list/tuple/dictionary/set/string is called iteration."

Let's do what we are famous for- taking realistic example. Suppose you and your girlfriend is making love (ahem ahem, sorry for hurting sentiments of few/ a few people here ha ha ha). What will you do? You have to take off clothes one by one, right? Unbound her hairs, put off her shoes, socks, kiss, her top, jeans, etc. That is iteration, **for** making love with her, you have to take steps one by one. Did you notice "for" word there? Yes, that what's for loop does.

Let's create, a list of work you have to do-

>>>making_love = ["unbound hairs", "put off shoes", "socks", "kiss", "top", "jean"]

#let's access each item from this list with help of the "for loop"

```
]:  1 #making list of the work you have to do to make love with your girlfriend
    2 making_love = ["unbound hairs", "put off shoes", "socks", "top", "kiss","jeans"]
    3

]:  1 for items in making_love: #notice the identation
    2     print(items)

unbound hairs
put off shoes
socks
top
kiss
jeans
```

As you can see that for accessing each item in list, we have used for loop-

>>>for items in making_love:
#making_love is data set from where you have to pick the items
#if you want to print items just type
>>> print(items)

Hope you have got it. Let me share another example with you-

```
:  1 #creating a data_set from where I'll be picking the items
   2 data_set = (1,2,3,4,5,6)
   3

:  1 for num in data_set:
   2     print (num)

1
2
3
4
5
6
```

#data set is a tuple here, it is like a list but it can't be changed, It is immutable. Tuples are preferred over the list becuase processing of a tuple set is way faster than the list.

Range() functions-

Let's talk about the range function here. You remember the range it starts from the begining index and ends at one less.

```
>>> for num in range(6):
>>>        print(num)
```
O/P = 0,1,2,3,4,5 (see one less than 6, indexing always starts from 0)

```
>>>for num in range(0,6):
>>>        print(0,6)
```
O/P = 0,1,2,3,4,5 (one less than 6, again starts from 0)

```
>>>for num in range(1,6):
>>>       print(num)
```
O/P = 1,2,3,4,5 (starts from 1, ending at one less than 6)

```
>>>for num in range (1,7,2):
>>>      print(num)
```
O/P = 1,3,5 (starts from 1, will end at 6 (7-1), but gap between two numbers will be 2)

```
>>>for num in range (0,10,2):
>>>     print(num)
```
O/P= 0,2,4,6,8 (like previous example, gap between them is 2 here, it should end it at 9 but when there is a gap of 2, 9 doesn't come)

#Generalising range in for loops
>>> for x in range (starts, end point, gap)
#if gap value is not provided, by default it will take one

```
>>>for num in (10,0,-1):
```

hanshofficial@gmail.com

>>> print(num)
O/P = 10,9,,8,7,6,5,4,3,2,1 (descending when gap is minus 1)

#for loop in a string so that we can access each item(letter in this case) of string
>>> name= “python” #this is a string
>>> for letter in name:
>>> print(latter)
O/P = p,y,t,h,o,n

#loop inside a loop : Concept of nested loop

Loop inside a loop is called nested loop
in this concept there are two loops
outer loops take control of inner loops

suppose we have an outer loops like this
>>> for items in [“first”, “second”]:
#it means there are two items in list [“first”, “second”], that is why it will run 2 times
>>> for items in [“first”, “Second”]:
>>> for num in [1,2,3]:
>>> print(items, num)
O/P= first 1, first 2, first 3
 second 1, second 2, second 3

Explanation – You can see that from outer loop “first” engages with each item of inner loop. “first” engages with 1, result is “first1” then “first” engages with 2 and 3 (from inner loop) hence results are “first2” and “first3”. Same works for second, it engages

with each item of inner loop and second1, second2, second3 are the results.

Nested loop is bit difficult to understand, I’ll have a separate lecture notes and video tutorial for that. Thanks for your time, happy pythoning till next lecture.

Thanks!

akshanshofficial@gmail.com