Python - Module 7: LOOPS

Python

Module 7: LOOPS

Video: Introduction for loops

Hello, welcome to Python Sessions.
In the last session, we completely discussed about different
ways how to use the uh, conditional statements.
So this session onwards,
we'll enter into the looping statements.
So faster we took some basic introduction to loops.
Like loops are required.
Upload the scenario. You have,
uh, some three, three
or four elements in one list, like a hundred, 200, 300.
Take that. The list name is x.
I want to print each and every value.
So suppose without looping,
what is the process without loops?
First of all, I will print x
of zero, then 100 will be printed.
If I say print X of 200, x
of one, which is second element, 200 will be printed.
Print x of two
or minus one, then 300 be printed
here in each and everything.
The statement is print,
but the thing, the same print
statement you have given three times.
What if, if it is a hundred times, okay,

you cannot say manually print off X of zero, X of one, x
of two like this up to X of hundred.
I cannot say. So there the looping
statements will come into the picture.
So advantage of loops is
to execute a statement
or block of statements to execute
a statement
or block of statements.
Block of statement with mul multiple statements.
It relatively the number
of times
in this case you'll be using these loops.
Basically in Python you have two types of loops.
One is for loop. Second one
is wild loop.
So what is the difference, how to use them?
You will see, now
I'm going to talk about the follow loop.
This follow loop be trades through a sequence.
Here. Sequence can be at least
or dictionary or even it can be a
set simply here.
Sequence, you can general meaning you can say it
as a collection of values.
So while working with the collections,
always better practices with funds.
And second thing is while loop here.
So if the given condition is true,
then only we'll enter into the loop.
So that means before starting the loop
only it'll check the condition.
Okay? If the condition is true, then it enters the loop.
If condition falls, it'll come out of the loop.

If the given condition is true,
loop ation will be stored.
If condition falls,
loop execution will be te.
A small example we'll see with the far loop
and in the next session we see practical part of it.
After this, the first exam, I supposed think that I have
five values here in the X where A is a list,
I want to print each way running a workload like this
for a fee in X exam site.
So here actually X contains like, uh, five values.
That's why it runs for five iterations at each iteration,
value is coming into fee.
Here we can give you anywhere.
So in the ation, what is the value of V 10?
I'm simply printing that V.
So without loop, we have to give five times print statement.
Now with thought loop, we have only one time.
So that is like a for V in X print of now it'll be like this
in the first iteration, ten second iteration, 23rd,
duration 50, next iteration, 60 and that hundred.
This is all. Okay, so list has five elements.
That's why five iterations run for five iterations,
the same style
while while loop.
Let's see the syntax and uh, how to operate this.
For example, task is like this
numbers from zero to
or one to 10.
So here the initial value of I some number, I'm taking
initial value I'm taking as well
because my starting point is one here
and I'm giving a condition.
While the condition is I less the

level because up to 10 I want to print it as well.
Less than 10 is condition.
Now check in the beginning, before the loop, I value one.
What about the condition? Condition is true.
Then you can print it.
Print five. At every threshold, I'm giving one increment
to the i i plus equals two one.
So first one will preprint, two printed,
three printed every I values keep on incrementing.
And before going for the next
generation, it is checking the condition.
If that condition is true, then
only fill enter into the loop.
Like this nine reached, 10 reached.
After then 10 I value will become 11. 11.
Less than 11 condition falls.
Then if you break the loop, it won't enter.
So that final, I got this loop exhibited for 10 times.
That means 10 iterations.
In this way, based on the condition, whether loop
to be started or not, to decide that you need
to use while loop
or else like if your sequence is having fixed number
of elements or some, a number of elements you want to work
with each and every element, then you will go with for loop.
Okay? This is the clarity between when to use for loop
and when to use while loop.
Okay? In the next session we see the practical part of it
and how to use these uh, while loops, more examples.
So we'll perform. Thank you very much.
Between the next session.

Video: For and While loop basics lab

Welcome to sessions.
In the last session we discussed about introduction
to looping statements.
As we discussed, we have two types of loops for loop
and while loop and we have got the clarity when
to use what type of the loop.
Now let's implement some practical stuff with that
PA loop format is like this
are variable in
any sequence already.
I told you sequence can be of anything.
It can be listed double dictionary set.
It can be of anything. That's why I simply say it as a
sequence and give the block of the statements.
So
this is basic format of the par.
So with this we do examples
and the Be example, I'm taking a list.
Let's say XX is having some five values like
10, 20, 30, 40, 50.
Check this line of x,
but five values that we have.
Now I want to iterate loop through all the sequence
part V in X.
So the Excels five values so
that the five iterations will happen
at each ation value is coming into V.
We are printing that.
Just look into this. So what if there is no loop?
What you have to say Print X of zero print.
Same thing like a printer X one.
In this for five statements you need to give.
So suppose what if, if you have like a hundred hundred
values, we cannot give a hundred print statements of here.

So simply loop at loop advantages to execute a statement,
a block of statements re okay, that is advantage.
Okay, and for the same sequence, one more example,
we'll see, look into the X value side.
I want to find out what the sum,
of course you have a predefined function thing
that you don't have any predefined function.
How to do, how to write the code within one.
I'm creating one initial variable.
So before looping, the total value is zero here and
after that I want to access each value for V in.
Yes, each value is coming into the V.
That value VI want to add to the total. So the total plus
Sequence.
Now this runs for five times auto.
Of the five times the total value is ready.
So 10, 20, 30, 40, 50, expected one 50 just like this.
Got it. Same kind of uh, demo we see for while. No.
For example, I want to print the values from one to 10.
So the starting value I, which is one I'm giving
while and then condition.
If this condition is true, then the block
of the statements, this is the format.
So I want from one to 10, that means
I hear the condition I less equals
to level.
After. If the given condition is true,
then only we'll get into the law.
Hmm, come on, check it please.
In the beginning, I value one,
check the condition one less can true
because of true you'll get into the low.
So in this, I'm printing
at each, I want to give an increment for that time.

I plus equals two. Just check it now.
So in the beginning I value one auto
accomplish of the fast iteration.
I begin to, again, it'll check the condition.
If condition two, then only it'll get into the loop.
In this way, 2, 3, 4, up, 2, 9, 10
for the 10 also conditions.
Two, once it entered into the loop,
after 10, 10, ation high value will become level
before going for next iteration.
Again, it is checking the condition 11 less than article two
10 condition falls.
Then it'll break the loop.
Just check where it is, not come
check all the valve one, two, and
whenever I reaches 11 condition falls,
then stop the loop.
But this is a basic difference
between follow loop and viral.
One more example we see with the wire,
I'm taking like a,
I'm asking the system for some text.
Let's say continue. We have predefined. Uh, what is there?
That's why want to continue.
This is my variable. Initially I'm giving blank back.
My condition is like this one to continue,
not equals to the word stop,
right?
I say hello and just a test.
Now I'm taking some input.
The input is into one, two,
continue input off.
Do you want to continue?
Even though user has typed upper case letters,

that is cannot into lower case
because of now just check this execution
initially that entered into the loop so that a hello printed
then it is seeking for user input.
I'm saying that yes. One more time.
It entered into the loop.
I'm saying stop now.
Next time the condition falls, the loop is break.
So conditionally want to control your loops.
Then while loop on a sequence to each
and every value you want to work, then go for Far Loop.
This is a basic difference between for loop and while Loop.
I hope all of you understood about this.
In the next session, we go
with the different scenarios based on the
for loop and while loops.
Thank you, Lammi there. Bye-bye.

Video: Finding sum and counting loops

Hello, welcome to Python sessions.
In last session we have seen some basic lab with the
for loops and wire loops.
Okay? Now with the for loop
or any Y loop, how
to perform the different aggregations fields,
the aggregations functions like a sum count.
So in this session, without using any predefined functions,
how to find the sum and count of a list.
We'll see, let
me take the rush list here.
Same previous value, 10, 20, 30, 40, 50.
Running the cell.

That's what it is. Connecting with the runtime
as a first step, I want to find out, count how many number
of elements are there,
number of values, the list.
I'm taking a counter variable with initial values.
Zero, which indicates that no values in the beginning,
before the follow, right, the follow
for we in, yes.
So part the loop, it reads for a number of times where is
and is number of elements at each iteration.
I'm adding one increment two count.
What is the meaning of the center statement?
CN NT plus equals to one means CNT equal two CN NT plus one.
So in the first duration, what's the value one?
The second duration one plus one. Two ation. Two plus one.
Three. Fourth duration three plus one. Four 50.
Duration four plus one, five
out five iterations loop will be terminated.
So finally what you'll get, the number of elements
that will keep it as a comment, print
number of elements that is C.
Okay, now C, totally count value.
Same thing. By using the same follow, I want to find out
total, I mean some
that means total of all values.
So here I'm taking on variable
total X.
So total plus equals two B.
So at each iteration, each value I'm adding to that total.
So, so please try to evaluate this, how it works
before loop total value zero in the first
iteration, what is the first value?
Next values
10, 20, 30, 40, 50.

In the first value, first iteration, we value 10 10, added
to total zero plus total to zero plus 10,
10 in the second iteration, V is 20, 20 plus 10,
30 in 30, duration in V is 30,
30 plus 30, 60.
In the next ation, 40 plus 60, oh.
And next ation in the 50 duration, 50 plus hundred or 50.
Now five times five is completed. Now total is red.
Some of X are, some is
total
one zero count product.
You can find out average the average equal to total divided
by number of elements that is sealed.
Now check the values of this average
total one 50 count 5, 1 50 divided by count.
That is that, okay?
But uh, separate one, separate far loop for for count,
separate far loop for, uh, totally hard rate.
Try to keep both of the things in one far loop
far because that other here I'm starting from the scratch
count, equal to zero, totally equal to C.
Writing a fur loop for V in X.
At each situation, I'm adding one to the count.
I'm adding value to the total, total plus equals to.
So after of this slope, both ready, total ready, average,
right now I'm computing average, average equal to total Y
count print total,
which is intuitive variable print
count, which is in c entering print
average, which is average.
Now on this or
Count ready average, right?
So in the next session we see other different aggregations,
like a maximum value, minimum value

that are without using predefined function.

Okay, see you in the next session. Bye-bye.

Video: Finding maximum and minimum with for loop

Welcome to Python sessions.

In the last session we discussed about uh,

finding some count and average with Pablo.

Okay, few more aggregations.

We'll see how to find the maximum

and minimum values with the Pablo

maximum.

Let's consider some of the values.

Just check all these values, which is the maximum,

but 300 is a max.

So both at the time of finding total

or account initial value, we have slice with a zero.

Our maximum. You should not keep that zero. Okay?

That's why any one value

among this list I will keep as initial value.

Let's say, which is max is first value. I'm keeping here.

Just check this.

The value tab, the first value kept into the max.

Now writing A for V in X,

if that V is greater than max, then I'm replacing

max as we.

I'm assigning V to B max.

After competition of this loop, your max value is ready.

Now print the max in the next Excel.

Maximum value is 300.

How this logic works, right, Einstein?

Previously I have values, I will copy the values also here.

These
are essential values
and we said max equal to X off zero.
That is first value. Kept it here.
What is that first value here? 10.
In the iteration one, check the condition.
10. Greater than 10. So condition here.
False. We
Iteration for the V value.
V value is minus check the condition
V greater than max.
That means minus 10 rather than 10.
So this is also false. Nothing will happen. Ation.
Three, look at the third value. The third value is 30.
We value 30 if
we greater than max is our condition.
We value 30. Now max value already
then condition True.
Then what will happen Max equal to
that means latest value of the max 30
iteration four V value
and the fourth iteration V value is 300.
Give 300 rather than 30 condition.
So that max become that 300.
Now iteration five and the 50
ation V value 23 condition falls
23 greater than now.
Latest max is 300. Condition falls.
Nothing will happen still. Max is 300.
So this is the last ation. Ation number six is 57.
57 is greater than max. Max is 300.
Again, condition pulse. So still the max contains three.
So after accomplished of this low for six iterations,
finally you could able to find out the max value,

which is 300 on the
with this logic bot, same
I want to find out minimum, once again,
print the values of x.
I have totally six values here,
as usual, same technique.
I'm keeping past value into this X, into this minimum,
which is 10 here.
This time I'm reversing the condition
for V in X.
If V less than minimum
previously V greater than max,
you said now V less than minimum.
If it is less than the current minimum value,
then minimum is equal two.
Now, after this loop, your minimum value is right
mini check the output now.
So here what is expected,
minus 10 is the minimum of all these values.
So minus 10 is the minimum.
Now for maximum separate loop,
for minimum separate loop we have, I want
to write both the things into one single.
Okay? So the technique max equal
to mini equal to X of zero
by following this technique, both contents, same values,
just check max, contents done,
and also min content stand here.
Data, same value we can assign
to multiple variables in this time, right?
Hmm. Now both are in slides max and win.
Both are in slides with the first value writing A, our loop
for V in X if V greater than max,
the max equal to,

and one more separate condition I'm writing.
If a wheel less than minimum, then minimum is equal to
finally print things, print x, print, maximum
max print, minimum
A, both right now,
check the final output.
Got it. So maximum 300, minimum value minus.
So of course we have predefined functions without predefined
functions for how we are developing your
one required functions.
Okay? So meet you in the next session.
We see more on that follow loops. Thank you.

Video: For loop on different collections

Hello, welcome to Python Sessions.
In our last two sessions,
we see the practical pr practical part of uh,
finding aggregations, like some counter average,
maximum minimum with help.
Okay, so now let's apply the far loop on different types
of the collection data types.
We have different types of data, uh, data collection types,
double
dictionary and set.
Of course there is one more kind
of things fraud set in a separate concept.
We'll separate session we'll discuss about the fraud sets.
So already have seen in the work with uh, for a like a photo
with the list objects.
So suppose if there is a double, let's say info.
I'm creating a couple.
First one is id. Second one is name.

Next one is salary, gender,
and something department number.
No, totally. This uh, info
is having five elements through
I want to access each, each element
or we in in
print that.
So there are five elements so
that five iterations will happen at each iteration.
Each element will be taken in the sequence. Okay?
So for loop can be applied not only on the list,
it can be applied on top levels.
Now same thing we see one example with the set collection.
What is set is a collection of mini items.
I'm writing a set
backup five values here.
Even though if you enter duplicate values,
it won't accept it.
Example of 53 times I'm giving,
but check, let say this is set one,
I'm printing that set one.
So P actually three times here,
but it has taken only one time.
Whenever I try to access that set elements,
I cannot access them using index numbers
only the way by performing a loop.
I can access it after this for
as in set home
Print.
But yes,
so far loop can be applied on even setups.
Okay? Of course here data is not in order.
If we want data to be in order we
to apply the sorting functionality.

Okay? Later we discuss about the sorting.
So after now we have seen Far loop on the list.
Far loop on the top and far loop on the set.
There is one more type of collection,
which is dictionary, DICT.
The full name of this DI CT is dictionary.
Dictionary is a collection of key and value beds.
Key and value beds. Bare means key.
The first one is key, second one is value.
Let's create something
and taking bank accounts.
For example, 1 0 1 is your account number
and our balance amount is 60,000.
Other customer 1 0 2 is balance. 15.
One more customer 1 0 3.
Value on lack,
1 0 4, 50,000.
So here totally bank is having four customers. See it.
Confirm that.
Now through this loop I want
to display each account holder details.
I want to print each account holder details after this.
For K in bank here, bank is dictionary.
So dictionary has four page,
but at each ation of only key part is coming into the K
just to print only.
The key only
1 0 1, 1, 0 2, 1, 0, 3, 1 0 4 account numbers as key.
That is coming into this K. Okay?
But I want to access both account number
and also it's balance amount
K in bank print
account number K,
and balance amount.

So bank of that K.
If I pass key into the dictionary, I will get value.
Now check what happens. You'll get a four account details
like 1 0 1 account number.
The total, like
The total balance amount is available with each account.
That nothing we are coming.
Now I want to find out
total amount in the bank.
I want to find out total amount in the bank.
What is the total amount in the bank with the follow.
I want to find out this. Hmm, as usual, our technique
equal two
sum in.
She zero writing a loop for gain bank
here instead printing the Strat will calculate total plus
equals and the ation K value is 1 0 1.
I want to get this 60,000.
That's 60,000 I want to add to this total.
That's why total plus equal to bank of
that not total.
After compression of this loop, the total amount in the bank
that is ready, the total amount in the bank is
to like 60%.
Got it. So in this way you can apply
for loop on different collection data types, like list,
double set dictionary.
Right? Okay. We see more operations in the next class.
Thank you. Let's make that.

Video: Break, Continue and Pass

Welcome to pattern sessions.

In the last session, we have seen how
to apply the follow loop on
different types of the collections.
Okay, so now
there are some controlling statements.
Those are break,
continue and pause.
So what is break is
when break is applied
And that loop will be terminated.
A second example for this
or I in arrange range of tab
I before printed,
if I equals two six,
I'm applying break tab.
We now actually this loop supposed
to be executed for 10 times.
A range of 10 means zero to nine.
Zero to nine means 10 times.
But whenever you applied some break statement
because of some condition through
then remaining iterations will not be continued.
That means entire loop will be breaked out.
So just check it because of this what output you'll get.
I got up to 0 2 5.
So this is what break is
for example, these are our marks.
So if minimum two subjects failed,
like the,
this is what the requirement think
that past market is at 35.
So for subject pass, second,
subject pass third, subject fail.
Fourth, subject failed already two reached the fail.

Number of subjects two reached then no need
to continue the remaining.
Okay, so I want to break that through.
So here, let's say you have C
and T where have C and Ts per failed.
That is zero. Writing a loop for Yamin box
check at yam greater than 35,
greater than Oracle store 35 is true.
I'm giving one increment for the C
when now this is reached two
and I don't want to continue.
Okay, so finally
if you have C and T print, FC
and check what happens.
Loop only four times.
Okay, for more clarity,
we said greater than Oracle 35 than FCT is less than Oracle.
Recall less than 35. We need to give it less than 35.
And that is a fail. The fail count is plus
if a CD value is A two, once it reach two, then break the
that.
Yeah, check this.
Hundred, 220, 23.
Fourth God fail, then no God. Finally you are printing.
Printing the minimum criteria matter. That is true. Got it.
So it won't go for the ones that condition.
Two, once you break it, it won't go to the next level.
That is what the break and
second thing is continue.
Let's check with the respond. Next statement is continue.
Once continue is applied
current ations and next step will not be.
I mean next steps means next statements
will not be executed.

Let's look into this
or I in range of tab.
If I equal to five,
simply saying that continue.
That means only current iteration, current duration.
Next statements will not be executed, then goes
to next iteration.
The next iteration will be continue.
Here I'm printing.
So the output here, column missed
except number five remaining.
All of these because once the condition met
to the file, okay, continue.
So that next spot will not be executed.
It is going to the next generation.
So this is with the cutting.
Same thing pass.
If pass is applied,
nothing is done.
So generally we at what scenarios we'll use.
This is observe this loop or I a range of,
but if I equal to three,
I don't want to do anything, then I'm keeping pulse.
Yes. If condition falls,
then normally I wanted to do something.
This. If I equal to three,
if the condition is true, nothing to be done.
If condition falls, that means INZI use not three,
then only, only remain takes to be the spot to be executed.
This is what my expectation, except three.
So here again, the column missed
zero.
1, 2, 3, 0, 1, 2, 4. The three is not printed here.
This is about pass.

Okay, so this, all the things will be used uh,
regularly in our applications.
Looping statements are very important. Okay?
So all dimensions of the thought loop
and while loop we have seen, okay?
See you in the next session with a different topic.
Thank you very much.