Python 101

Flow Control(Looping) and Introduction to Data Structures in Python (Numbers, Strings and Lists)

Contents

An introduction to Numbers, Strings and Iterables

Numbers - int and float eg. 1 and 1.1

Strings - Array of characters or even a single character is a string in python. There is no such thing as an independent data type called character in python. These can be represented using double quotes “Double_quote_string” or ‘Single_quote_string’.

Iterables - Iterable is anything that is a collection of a number of distinct objects over which we can iterate. Iteration means going over each value one by one. Lists are a type of Iterable in python. We can go on their each value one by one using iteration.

An introduction to Numbers, Strings and Iterables contd..

Lists- Lists as we know are iterable. They are represented using the literal [ ].

These CAN be heterogeneous but SHOULD be homogeneous. We declare a list as follows -

x=[1, 2, 3, 4, 5, 'abcd', 'efgh']

Looping in Python

Looping means doing something over and over again.

Now doing something over and over again can be of various types. Examples -

- Doing something till a condition is met. (Eating and eating till stomach is full)
- Doing something for N times.(Cramming Formulas 3 times before exam)
- Doing something to each thing in a group. (shaking hands with each of your friends)
- Doing something Forever. (Breathing.. Well.. not really but virtually forever)

Let us try doing the above 4 things in python via loops.

While loops

What it can do ?

It is meant to be Doing something till a condition is met or Doing something Forever but it can also Do something N times with a little tweak.

Syntax:

```
while BOOLEAN_EXPRESSION:

    STATEMENTS_TO_LOOP_THROUGH
```

Try out the following code on the interpreter

```
>>> alive=True
>>> while alive:
. . . .    print(“Breathing Till Death by KeyboardInterrupt ctrl+C or ^C”)
>>> stomach_full=’no’
>>> while stomach_full!=’yes’:
. . . .    print(‘Eating Food’)
. . . .    stomach_full = input(“Is stomach full? Type yes or no in lower case only.”)
Eating Food
Is stomach full? Type yes or no in lower case only.
no
Eating Food
Is stomach full? Type yes or no in lower case only.
yes
>>> cramming_done_times=1
>>> while cramming_done_times<=3:
. . . .    print(cramming_done_times, ’times Crammed Formulas’)
. . . .    cramming_done_times +=1
1 times Crammed Formulas
2 times Crammed Formulas
3 times Crammed Formulas
```

For loops

What it can do ?

It is meant to Do something N times or Do something to a collection of things.

Syntax:

```
for iterating_variable in iterable:

    STATEMENTS_TO_LOOP_THROUGH_with_iterating_variable
```

Try out the following code on the interpreter

```
>>> friends=['friend1', 'friend2', 'friend3']
>>> for friend in friends:
. . . .    print('Shaking hands with', friend)
Shaking hands with friend1
Shaking hands with friend2
Shaking hands with friend3

>>> for crammed_times in range(3):
. . . .    print(crammed_times+1, 'times formulas crammed')
1 times formulas crammed
2 times formulas crammed
3 times formulas crammed
```

But what is that range(3) ?
We will study it more in depth when we will study functions and generator functions. For now let us say that range is something that gives us a list of integers according to the arguments we give to it.
ie. range(start, end, step) by default step is 1 and start is 0.
If it is given as range(some_number) then python thinks that we need the range from 0 to some_number.
Ie. Here range(3) means a list as [0,1,2]
Similarly range(1,10) means [1,2,3,4,5,6,7,8,9]

Continue & Break

Continue means just skip this time but keep doing what you were doing.

Break means just stop whatever you are doing.

Ie. Continue skips iteration in a loop while Break stops the iteration completely.

Something to focus on is that Continue and break are almost always used inside some conditional.

Try out the following code on the interpreter

```
>>> friends=['friend1', 'friend2', 'friend3', 'friend4']
>>> for friend in friends:
. . . .    if friend=='friend2':
. . . .        continue
. . . .    print('Shaking hands with', friend)
Shaking hands with friend1
Shaking hands with friend3
Shaking hands with friend4

>>> for friend in friends:
. . . .   if friend=='friend2':
. . . .       break
. . . .   print('Shaking hands with', friend)
Shaking hands with friend1
```

Doing Nothing

In a code block if we needed to do nothing in C/C++/Java we used to put { } just nothing in between the braces.

But python will throw an error if you will not give anything in a code block.

So when we need to give Nothing we give **pass**

Try out the following code on the interpreter

```
>>>giving_exam=True
>>> while giving_exam:
. . . .    confident = bool(input('Are you confident? Reply with something for yes else just press enter for no'))
. . . .    if not confident:
. . . .        pass
. . . .    else:
. . . .        print('Solving Questions')
yes
Solving Questions
asdas
Solving Questions



yes
Solving Questions
```

Loop Else

Unlike most of the languages where **else** has no other work than working as a negation of if statements in python else can also be used with loops.

In loop else, else executes if there was no break in the loop and the loop has finished looping.

We can use such constructs in order to detect whether our loop executed completely or not.

Lets try building a simple guess game where player has 5 chances to guess the number we will feed in our system.

Try out the following code on the interpreter

```
>>> lottery_number=42
>>> for guess_counter in range(5):
. . . .     guessed_number = int(input('Guess the prefeeded number to win absolutely nothing.'))
. . . .     if guessed_number==lottery_number:
. . . .          print('You win !')
. . . .          break
. . . .else:
. . . .    print('You didnt win')
```

Enumeration in For Loop

Enumeration in english means the action of mentioning things one by one. Programmatically also it has nearly same meaning. It means assigning a unit-step count to a list of things so that they can be mentioned using their count if needed. Its just like the index we used in arrays in C.

As we saw that for loop is quite unconventional in Python. We do not have access to the index of an element in any iterable while looping through the iterable’s contents. In this case we can use enumerate.

Try out the following code on the interpreter

```
>>> x = [‘a’, ’b’, ’c’, ’d’]
>>> for index, element in enumerate(x):
. . . .     print(element, ‘is at index’, index)
a is at index 0
b is at index 1
c is at index 2
d is at index 3
```

Nesting of Loops

Just like we nested the Conditionals in the same manner we can nest the loops as well. Ie. Loop inside a Loop.

Such as following -

```
for i in range(3):

    for j in range(3):

        print(i, j)
```

A small Introduction to Data structures in Python

Data structure is a data organization and storage format that enables efficient access and modification.

Primarily Python has 5 data structures that we need to discuss.

List, Tuple, Dictionary, Set. All of the data structures as it might seem obvious by now are accessed by objects in python.

A small Introduction to Data structures in Python

Data Structure vs Data Type

Data structure is a data organization and storage format that enables efficient access and modification. They are composed of organizations of data types.

A data type is a potentially infinite class of objects that all share the same property. Like ints, floats, strings etc. They are irreducible and do not have multiple implementations.

eg. integer is a data type but list is a data structure. Lists do not hold some property of their own value wise. They just represent a bunch of floats, strings, or ints in a different organization.

Data Structure’s Immutability

A Data Structure organized in such a manner that it’s value does not change is called immutable data structure. In broader sense immutable data structure are analogous to persisting data structures.

In python though, two data structures called tuple and frozensets exists which negates every alteration to them completely. They are strictly immutable.

In a more general case a Data structure could be called immutable if even we could make separate copies of it for each change. Instead of disallowing changes completely.

An approach of implementing immutability is called “path tracing”.

Interview Questions

Why do we use Pass ?

Pass is used in order to do nothing in a code block.

It is equivalent code block as of C’s { }.

What is loop else? Give an example?

Else in python has more significance than just a counterpart of if statements it can be used with loops also. It is a great tool to check whether a loop has completed the execution completely or it has “**broken**” out of the loop.

An example can be a simple for else.

Can we implement forever-loop using for?

Yes. But it won't be as efficient as while implementation.

```
x=[1]

for i in x:

    x.append(i)
```

append adds an element in a list. We will study it thoroughly in Data Structures Part-1.

What do you understand by Iterables?

Iterables are a heterogeneous or homogeneous collection of various objects from which each element can be accessed one by one by iterating over it.

Difference between continue and break.

Continue is used for skipping statements between iteration and break is used for stopping iteration completely.

What is enumeration used for in python?

Enumeration is way to get easier indexing in iterables.

What is difference between Data Structure and Data type?

Data structures are a way of organizing the data types.

Data types are quantities that represent a potentially infinite set of values having similar characteristics.

What are immutable data structures? Name a few?

Immutable data structure are those whose value doesnt change. In python frozenset and tuple are immutable.