Scientific Computing with Python Lab

6th Session(Feb 27th)

Kaheon Kim : https://github.com/kaheonkim/Scientific-computing-with-python-lab-material

Today

we are going to talk about

- #list_basics
- #list_for_loop
- #list_updating
- #making_function_with_the_list

What is list?

basic things

Lists are used to store multiple items in a single variable

- ex) A = [1, 2, 7, 8]
- ex) A = [1, 2, ‘a’, 1.5, 7, [2,4,’c’]]

The index starts with 0 to its (length -1)

The last index would be found by plugging in -1

Extracting the value from the list A[starting_index:ending_index+1]

Basic operations of list

characteristics of a list

- length of the list : number of elements in the list A

  len(A)

- maximum/minimum : find the largest/smallest value in the list A

  max(A)/min(A)

- total sum : sum of the elements in the list A

  sum(A)

Exercise

Basic operations of list

Manipulating the list

- add element : adding element a to the list A

  A.append(a)

- remove : remove the value a in the list A

  A.remove(a)

- pop : remove the value at location i in the list A

  A.pop(i)

- reverse: reverse the elements of the list in place

  A.reverse()

- Sort the list A with numbers in increasing order

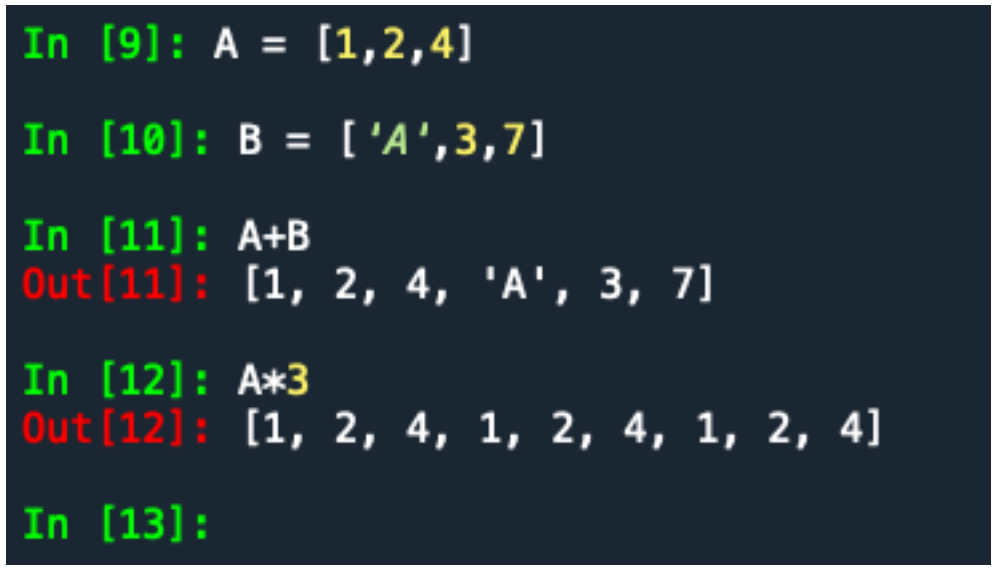
  sorted(A)

Basic operations of list

Arithmetics between lists

+ (Concatenating) : adding to list is concatenating two lists

* (Repeating) : multiplying a positive integer n to the list A means repeating the value of list A n times

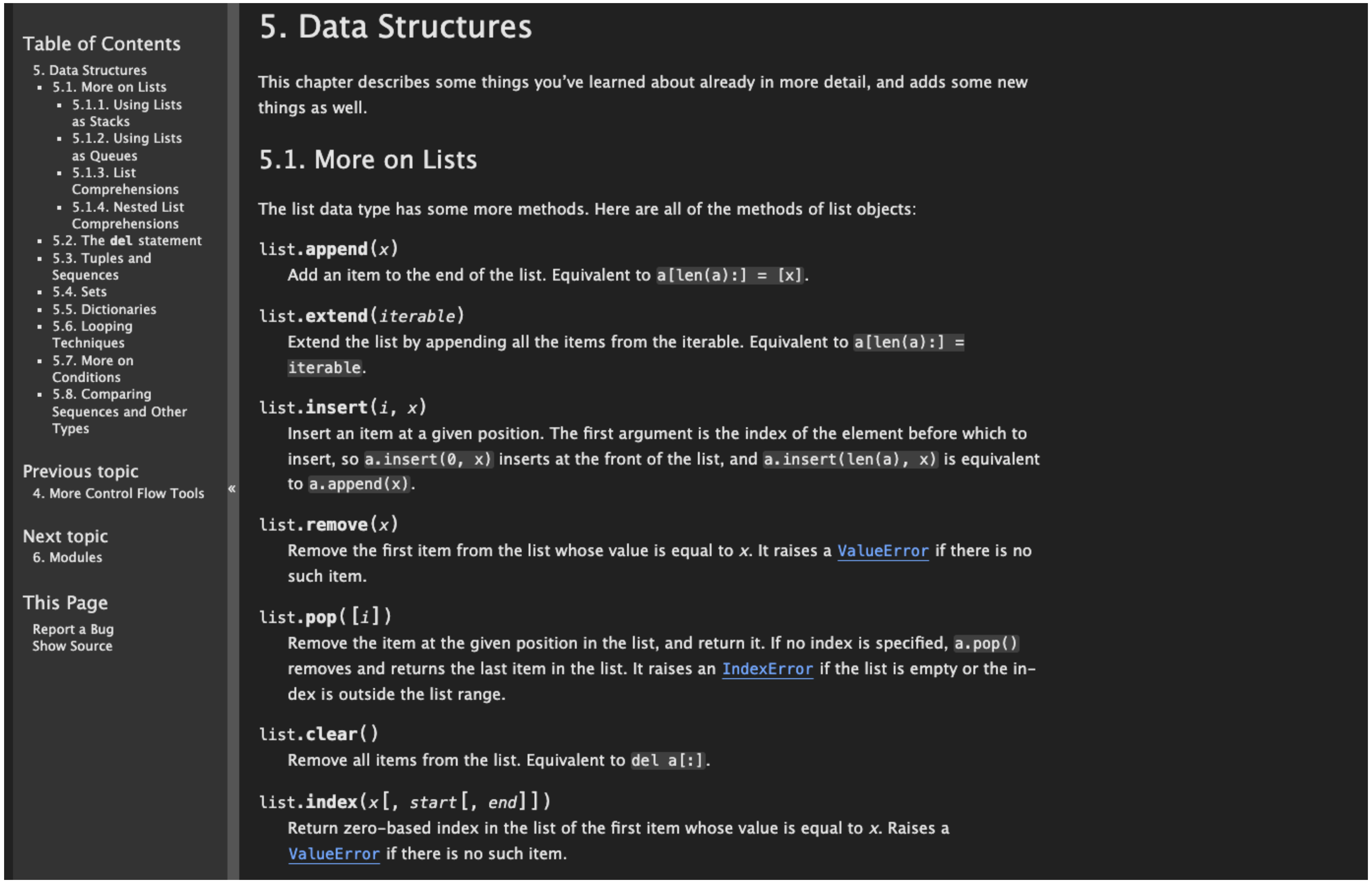


```
In [9]: A = [1,2,4]

In [10]: B = ['A',3,7]

In [11]: A+B
Out[11]: [1, 2, 4, 'A', 3, 7]

In [12]: A*3
Out[12]: [1, 2, 4, 1, 2, 4, 1, 2, 4]

In [13]:
```

Exercise

Other functions for the list

Documentation for list



Previous topic

4. More Control Flow Tools

Next topic

6. Modules

This Page

Report a Bug
Show Source

5. Data Structures

This chapter describes some things you've learned about already in more detail, and adds some new things as well.

5.1. More on Lists

The list data type has some more methods. Here are all of the methods of list objects:

`list.append(x)`
Add an item to the end of the list. Equivalent to `a[len(a):] = [x]`.

`list.extend(iterable)`
Extend the list by appending all the items from the iterable. Equivalent to `a[len(a):] = iterable`.

`list.insert(i, x)`
Insert an item at a given position. The first argument is the index of the element before which to insert, so `a.insert(0, x)` inserts at the front of the list, and `a.insert(len(a), x)` is equivalent to `a.append(x)`.

`list.remove(x)`
Remove the first item from the list whose value is equal to *x*. It raises a `ValueError` if there is no such item.

`list.pop([i])`
Remove the item at the given position in the list, and return it. If no index is specified, `a.pop()` removes and returns the last item in the list. It raises an `IndexError` if the list is empty or the index is outside the list range.

`list.clear()`
Remove all items from the list. Equivalent to `del a[:]`.

`list.index(x[, start[, end]])`
Return zero-based index in the list of the first item whose value is equal to *x*. Raises a `ValueError` if there is no such item.

Making functions with list inputs

- Making function for the list

```
def func(p,..):
    operation in p
```

Exercise

function that finds mean of the list p

Making functions with list inputs

- Making function for the list

```
def func(p):
    **operation** in p
```

Using list for loops

Case 1 : matching all the indices of the list

```
for i in range(len(A)):
    Operations for A[i]
```

Case 2 : Directly using the values

```
for a in A:
    Operations for a
```

Exercise

checking the value : print

Using list for loops

Case 1 : matching all the indices of the list

```
for i in range(len(A)):
    Operations for A[i]
```

- when you have to deal with the indices the list

Case 2 : Directly using the values

```
for a in A:
    Operations for a
```

- More straightforward/convenient

⇒ None is absolutely better than the other one !! You have to make choice!!!

A = [2,4,7,10,11]

Adding up all the squared values for A

Practice

Updating schemes

start with the empty list

```
empty_list=[ ]
```

In each iteration/operations or in certain conditions, update the value by ‘append’ing the values

Exercise

Make function finding maximum index in the list p

ex) A = [3,4,1,-5,4,2] ⇒ 1, 4

Practice

Make function that only extract the even number and the indices

ex) A = [3,4,1,-5,4,2] ⇒ [4,4,2], [1,4,5]