Collection Types – Sequences and Dictionaries

Anastasis Oulas
Evangelos Pafilis
Jacques Lagnel

What are data collections

- We can group data types in a container – this is defined as a collection
- There are 2 main types of collections in python:
  - **Sequences** – can be indexed i.e. String, Lists, ranges.
  - **Mappings** – can not be indexed – i.e. Dictionaries

Types of python sequences

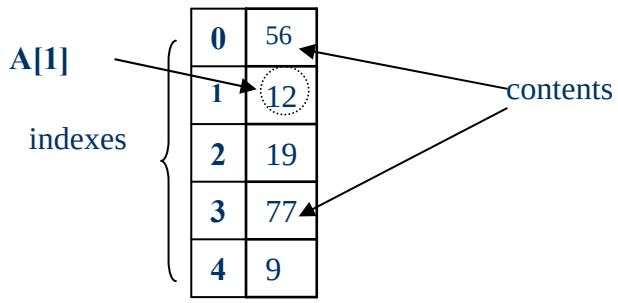
- String – array of characters i.e.
  - ‘ACGT’
- Range - Integers i.e.
  - {2,6,1,0}
- List – any element i.e. both
  - numeric based
    - [1,56,7,128]
  - character based
    - [‘AC’, TATA’, ‘GT’]

Basic sequence functions

- Referencing an element in a sequence is done so using square brackets – known as ***indexing***
  - `a_seq[0]`
- The number of elements in the sequence a_seq
  - `len(a_seq)`

Sequence Functions

The following shows a graphical representation of a sequence of type List called A, with 5-elements (integers).



To refer to specific element in the List one can write A[i], where A is the name of the List and i (takes values from 0 until the length of the List) is the index of the element in the List.

For example **`print(A[1])`**, the result is : 12.

`len(A)` will return the length of the list : 5

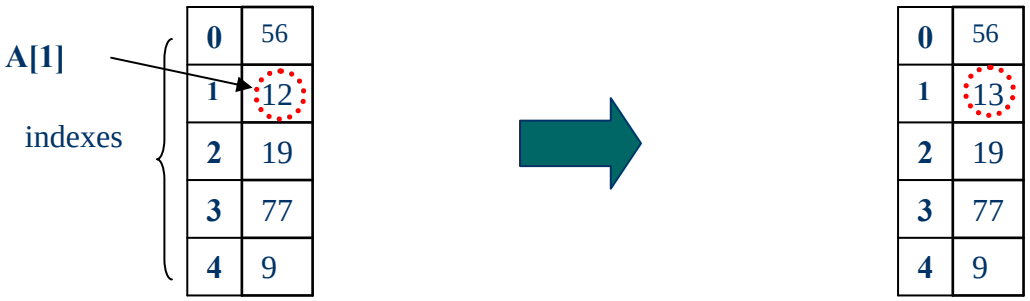
Creating a python List

- A list of strings:
  - `a_seq = ['acgt', 'ttttt', 'gcgcgcg']`
- A list of integers:
  - `Nums = [12,-4,23,1,0]`

Specific List Functions

- Modify element in List

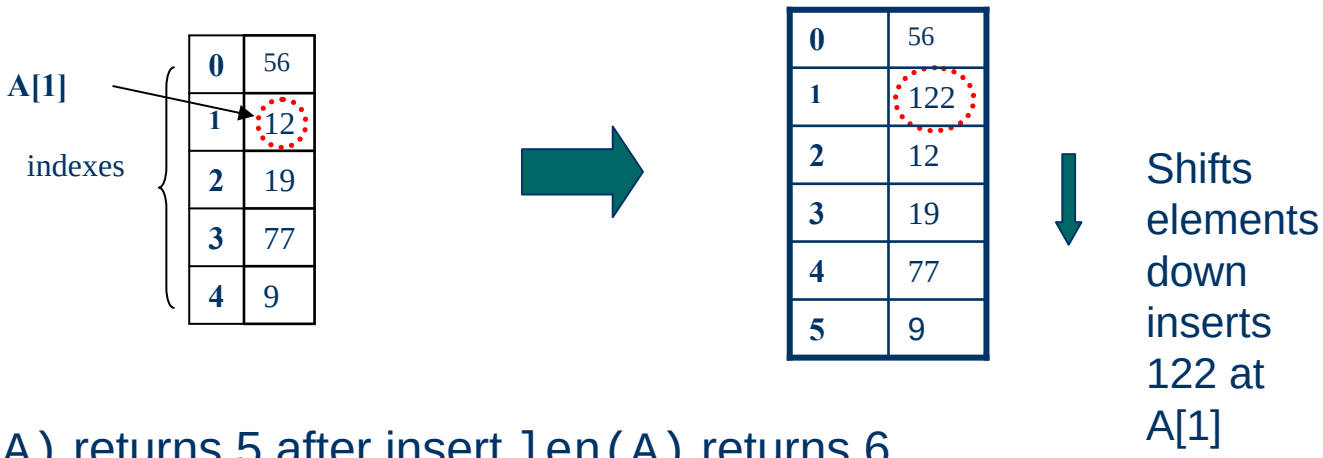
```
A[1] = 13
```



Specific List Functions - continue

- Add element in list

  `A.insert(1, 122)` – where pos is an integer indicating the position (index) where the new element, 122, will be added.

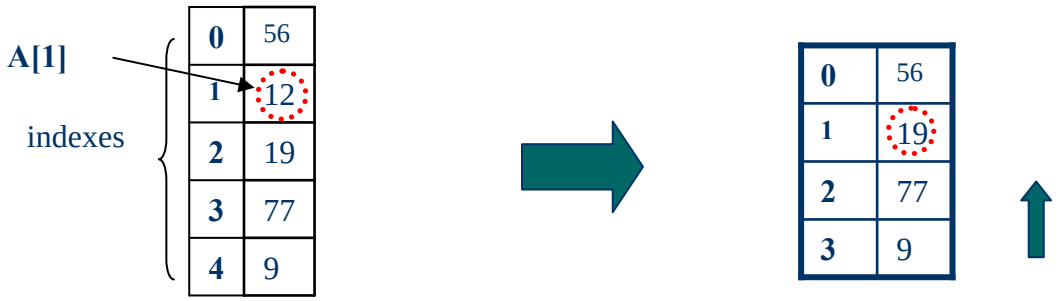


- `len(A)` returns 5 after insert `len(A)` returns 6

Specific List Functions - continue

- Remove element from list

  `A.pop(1)` – i is the index of the element to be removed.



Shifts elements up removes element at A[1]

- `len(A)` returns 5 and after remove `len(A)` returns 4

Creating a python List using keyboard input

- To read input form the keyboard directly into a python list a loop structure is necessary. i.e. read 5 numbers and store them in a list.

```
A = [] #initialize your list
for i in range(5):
   x = input('Enter a number')
   A.insert(i,x)
print(A)
```

Iterating through the elements of a python List

- In order to iterate through the elements of a python List a loop structure is necessary.

```
a_seq = ['asgt', 'ttttt', 'gcgcgcg']
for i in range(0,len(a_seq)):
   print(a_seq[i])
```

Copying the contents of one List into another

```
A = [-12,67,1,89,-3,0,100,23,65,34,-64,98,12,27]
B = []
pos = 0
for i in range(len(A)):
  B.insert(pos,A[i])
  pos = pos+1
print(B)
```

Don’t forget!

- You can think of the integer i as a link for every element in the python sequence.
- It starts from 0 as the first element of any sequence is denoted as A[0] (where A is the name of the List)
- A list with 10 elements will have indexes from 0 to 9

Example algorithms – sum

- Finding the average of the elements in a List of integers

```
seq_int = [34,45,56,67,12,5,24]
sum =0
lenth_seq_int = len(seq_int)
for i in range(0, lenth_seq_int):
  sum = sum+seq_int[i]
average = sum/lenth_seq_int
print(average)
```

Example algorithms – min

- Finding the minimum element in a sequence of integers:

```
seq_int = [34,45,56,67,12,5,24]
min =seq_int[0]
length_seq_int = len(seq_int)
for i in range(1, length_seq_int):
  if seq_int[i] < min:
       min = seq_int[i]
print(min)
```

#notice that we start from index 1 (second element in the list)?

Example algorithms – max

- Find the maximum element in a sequence of integers also retaining the position of max:

```
seq_int = [34,45,56,67,12,5,24]
max =seq_int[0]
length_seq_int = len(seq_int)
pos = 0
for i in range(1, length_seq_int):
  if seq_int[i] > max:
        max = seq_int[i]
        pos = i
print(max)
```

2D lists

Column indexes

Row indexes

K[1][2]

	0	1	2	3	4
0	32	0	255	95	51
1	435	68	57	88	13
2	56	45	14	2	85

Creating a 2D List

- A 2D list of integers:
  - `A = [[1,2,3],[4,5,6],[7,8,9]]`
- Referencing an element in a 2D list is done so using square brackets
  - `A[i][j]` – where i is the index for the row and j for the column
- The number of elements in the 2D List
  - `len(A)` – gives number of rows
  - `len (A[i])` – returns the number of columns
- Modify element in List
  - `A[i][j]= 0`
- Add element in list
  - `A.insert(pos,B)` – where B is another List
- Remove element from list
  - `A.pop(i)` – where i is will be the index for a list

Iterating through the elements of a 2D List

```
A = [[1,2,3],[4,5,6],[7,8,9]]
for i in range(0,len(A)):
  for j in range(0,len(A[i])):
     print(A[i][j])
```

Searching algorithm using Lists

```
table = [3,6,11,1,8]
done = False
position = 0
i=0
a_number = 34
while(done == False and i<len(table)):
   if(table[i] == a_number):
        done = True
        position = i
   else:
        i = i + 1
if(done == True):
   print('found ', a_number, ' at postions ', position)
else:
   print(a_number, ' was not found')
```

Sorting algorithm using Lists

```
table = [3,6,11,1,8]
for i in range(1,len(table)):
  for j in range(len(table)-1, i-1, -1):
     if(table[j-1] > table[j]):#increasing order
          temp = table[j-1]
          table[j-1] = table[j]
          table[j] = temp
print(table)
```

Explanation of Sorting algorithm – bubble sort

		i = 1					i = 2				i = 3			i = 4
		j = 4	j = 3	j = 2	j = 1		j = 4	j = 3	j = 2		j = 4	j = 3		j = 4
3		3	3	3	**1**		1	1	1		1	1		1
16		16	16	**1**	**3**		3	3	**3**		3	3		3
11		11	**1**	**16**	16		16	**8**	**8**		8	**8**		8
1		**1**	**11**	11	1		**8**	**16**	16		**11**	**1**		**1**
8		**8**	8	8	8		**11**	11	11		**16**	16		**16**
	change:	No	Yes	Yes	Yes	change:	Yes	Yes	No	change:	Yes	No	change:	Yes

Dictionaries

- Mapping (no index) collection of elements that have a key and a value:
- Example code:

```
dictionary = {}
dictionary['A'] = 'adeninde'
dictionary['C'] = 'cytosine'
dictionary['G'] = 'guanine'
dictionary['T'] = 'thymine'
print(dictionary['A']) # what will this
  print?
```

File I/O – reading from a file

- `F = open('C:\Documents and Settings\Administrator\Desktop\User\Python course\Seq.txt', 'r')`
- F is the file handler allows you to have a direct link to the contents of the file – Seq.txt
- `lines = F.readlines() # command reads all the lines of the file into a list called lines`
- `F.close()`

File I/O – writing to a file

- `F = open('C:\Documents and Settings\Administrator\Desktop\User\Python course\Out.txt', 'w')`
- F is the file handler allows you to have a direct link to the contents of the file – Seq.txt
- `F.write('Hello') # command writes the word “Hello” in the file Out.txt`
- `F.close()`