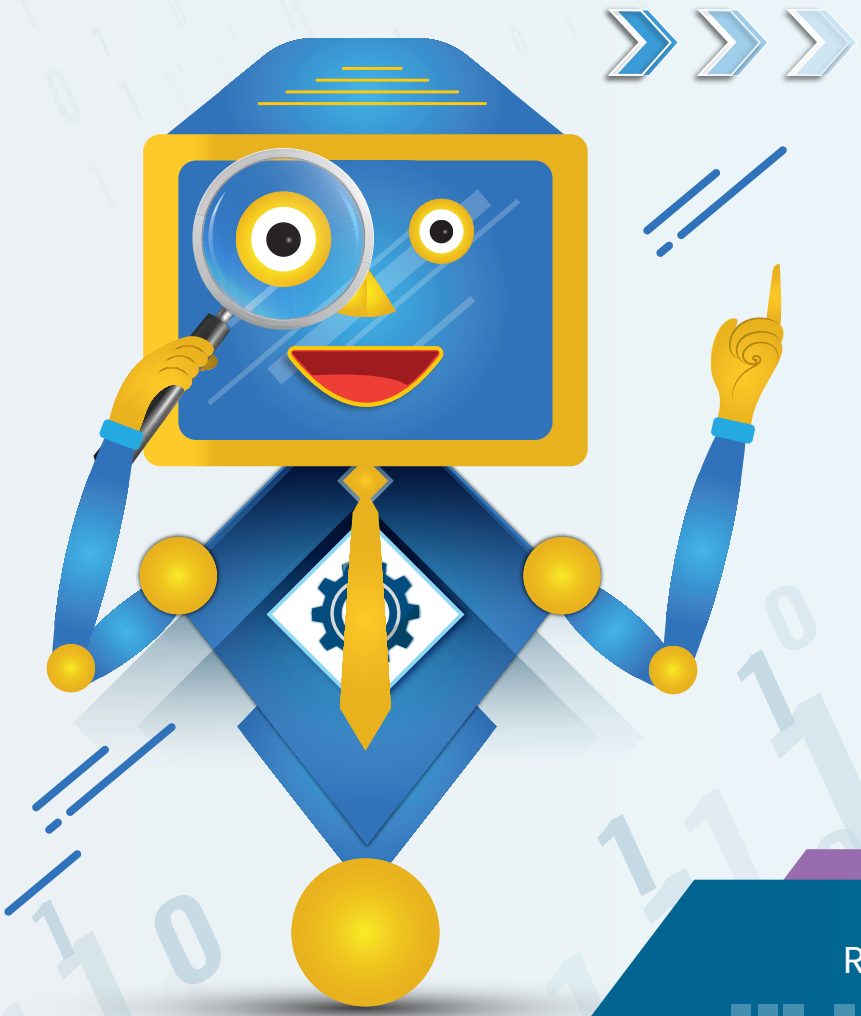
Learning Python

Intermediate Level

Python Run

Student's Book

Level 2



Rana Dajani

LKD EDUCATIONAL RESOURCES

PythonRun- Intermediate Level

Published by **LKD Educational Resources 2022**

Amman - Jordan
Tel: +962 6 5374141
Fax: +962 6 5516404
P.O.Box: 851346
Email: info@lkd.com.jo
Website: www.lkd.com.jo

Author

Rana Dajani

ISBN: 978-9923-781-06-7

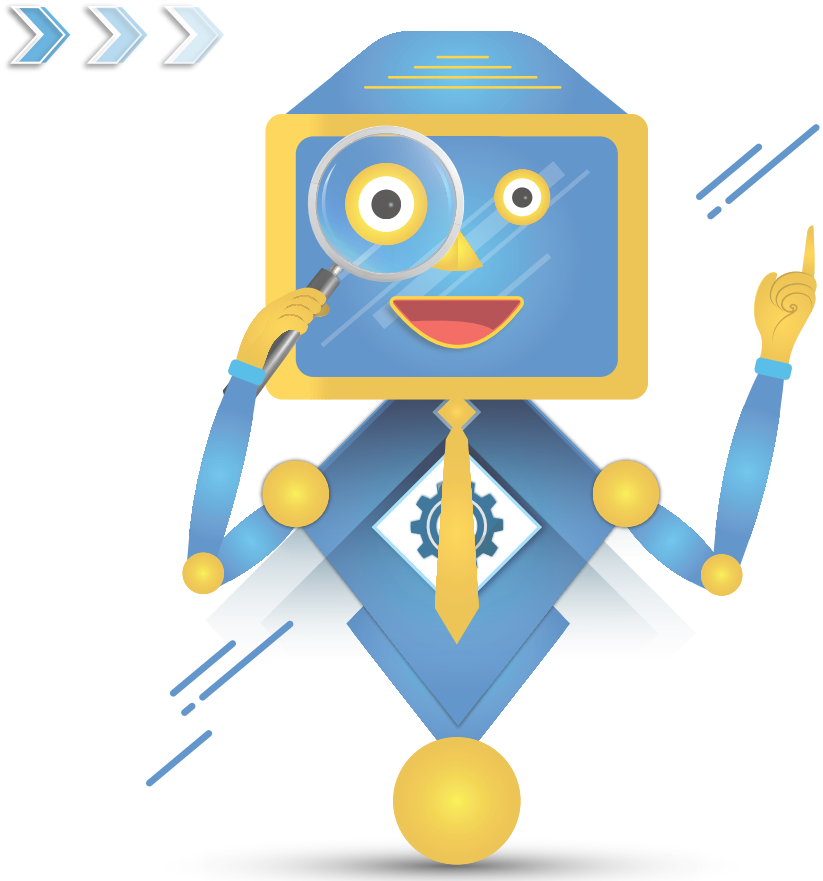
Learning Python

Intermediate Level

PythonRun

Level

2



LKD
EDUCATIONAL RESOURCES

A guide to learning Python programming language

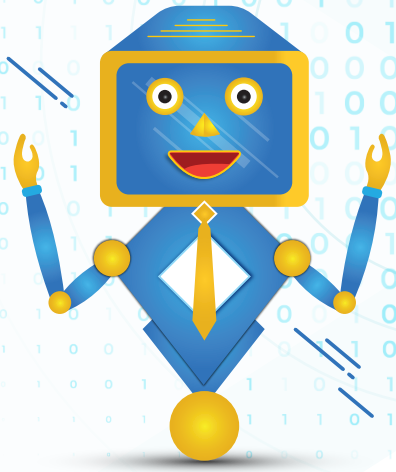
Introduction

This book is the intemediate level book in the ***PythonRun*** series. To get the most of this title, you should be familiar with the Python programming language concepts and know about variables, while loops, for loops, if, elif and else statements.

You will build on the knowledge that you have gained, from the beginner level book, and delve deeper in learning more Python syntax fundamentals, data types and functions.

Feel free to experiment with the code you write and see what else you can make it do.

If you try all the challenges, exercises and ideas, and generally play with the code, this will help you learn how to write code like a professional.



START CODING NOW! .....

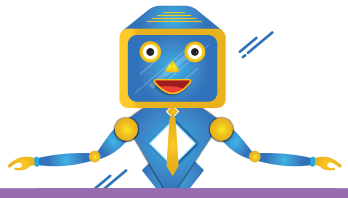


Table of Contents

Unit 1 >> Lists in Python

Unit 2 >> Lists For Loops

Unit 3 >> Tuples, Maps and Dictionaries

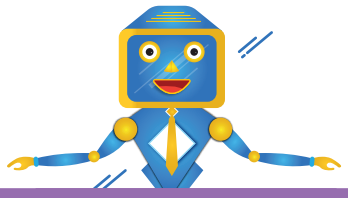


Table of Contents

Unit 4 >> Functions

Unit 5 >> Libraries in Python

>> Project - Based Assessments

\

Unit 1 »

» Lists in Python

» Lists Manipulations
» List Methods
» Join Method
» Len(), Del() Functions
» List Arithmetic
» Min(), Max() Functions
» Quick Tests

Lists in Python

1.1 Lists Manipulations

Lists are a datatype you can use to store a collection of different pieces of information as an ordered sequence of items under a single variable name.

All the items in a list do not need to be of the same type.

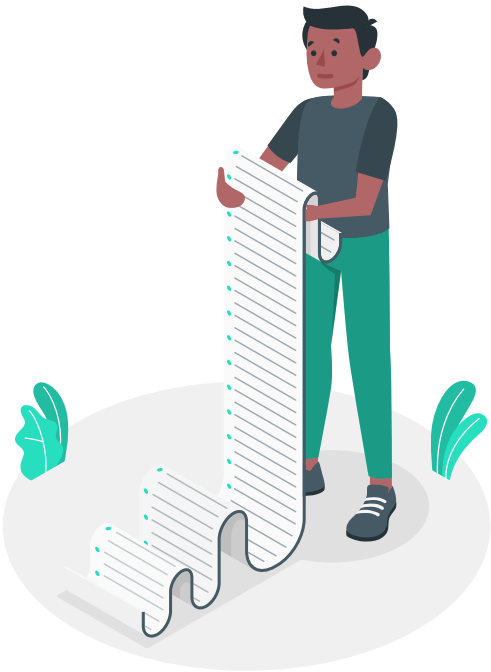
We use lists all the time in life, like your birthday wish-list, or a bucket list of the best types of buckets!

Here is what a shopping list would look like written as a list in Python:

```
>>> shopping_list = ['eggs', 'chicken', 'apples', 'juice', 'pasta']
```

To create a list variable we first assign it a name followed by an equals sign, then *square brackets []* with the items inside separated by comas.

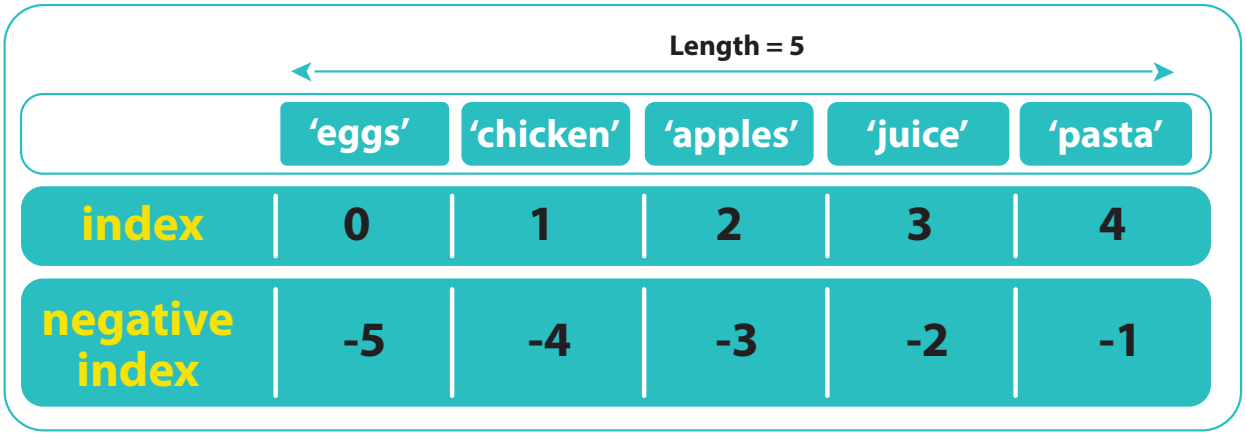
We can manipulate a list by finding items, subsets and the length of the list, and replace, add and remove items from the list.



List Index

You can access an individual item on the list by its index.

An index is like an address that identifies the item's place in the list. The index appears directly after the list name, in between brackets, like this: **list_name[index].**



To find the third item in a list:

```
>>> shopping_list = ['eggs', 'chicken', 'apples', 'juice', 'pasta']
>>> print(shopping_list [2])
apples
>>> print(shopping_list [-3])
apples
```

The **index()** method searches an element in the list and returns its position/index.:

```
>>> print(shopping_list. index('pasta'))
4
```

Replace Items

You can save over or replace an item in a list using the index number of the item you want to replace.

To replace an item:

```
>>> shopping_list = ['eggs', 'chicken', 'apples', 'juice', 'pasta']
>>> print(shopping_list [2])
apples
>>> shopping_list[2] = 'bananas'
>>> print(shopping_list)
['eggs', 'chicken', 'bananas', 'juice', 'pasta']
```

Practice: List Manipulations

Make a list containing 4 zoo animals and print the second and the last one. Also, change the first item to your favourite animal and print the new list.

```
>>> zoo_animals = [“pangolin”, “ostrich”, “sloth”, “lion”]
>>> print (zoo_animals[1], zoo_animals[3])
ostrich lion
>>> zoo_animals[0] = ‘penguin’
>>> print (zoo_animals)
[“penguin”, “ostrich”, “sloth”, “lion”]
```

List Slicer

You can extract subsets from the main list using the slice notation in Python, like this: **list[start: stop : step]**

To extract the third to last items:

```
>>> shopping_list = ['eggs', 'chicken', 'apples', 'juice', 'pasta']
>>> print(shopping_list [2: ])
['apples', 'juice', 'pasta']
```

To extract the second to fourth items:

```
>>> shopping_list = ['eggs', 'chicken', 'apples', 'juice', 'pasta']
>>> print(shopping_list [1:4])
['chicken', 'apples', 'juice']
>>> # [1:4] shows the items from index 1 up to (but not including) index 4 (in other words, items 1,2 and 3).
```

List Step Slicer

To see all even index items of a list:

```
>>> shopping_list = ['eggs', 'chicken', 'apples', 'juice', 'pasta']
>>> print(shopping_list [0: :2])
['eggs', 'apples', 'pasta']
```

To see the reverse of a list:

```
>>> shopping_list = ['eggs', 'chicken', 'apples', 'juice', 'pasta']
>>> print(shopping_list [0: :-1])
['pasta', 'juice', 'apples', 'chicken', 'eggs']
```

Practice: List Slicer

Given a list

suitcase = ["sunglasses", "hat", "passport", "laptop", "suit", "shoes"]

- Create a list called first, containing only the two first items from suitcase.

- Create a list called middle, containing only the two middle items from suitcase.

- Create a list called last, made up only of the last two items from suitcase.

```
>>> first =suitcase[0:2]
>>> print (first)
['sunglasses', 'hat']
>>> middle =suitcase[2:4]
>>> print (middle)
['passport', 'laptop']
>>> last =suitcase[4:6]
>>> print (last)
['suit', 'shoes']
```

1.2 List Methods

There are several built-in methods that can be invoked on lists.

.pop(index) - will remove the item at index from the list and return it to you.

```
>>> shopping_list = ['eggs', 'chicken', 'apples', 'juice', 'pasta']
>>> print (shopping_list.pop(4))
pasta
>>> # This item can be saved into a variable.
>>> food = shopping_list.pop(4)
```

.remove(item) - will remove the actual item if it finds it.

```
>>> shopping_list = ['eggs', 'chicken', 'apples', 'juice', 'pasta']
>>> shopping_list.remove('apples')
>>> print (shopping_list)
['eggs', 'chicken', 'juice', 'pasta']
>>> # This item is deleted from list and it's value isn't returned
so it can't be stored into a variable.
```

.append(item) - will add an item to the end of the list.

```
>>> shopping_list = ['eggs', 'chicken', 'apples', 'juice', 'pasta']
>>> print (shopping_list.append ('bread'))
['eggs', 'chicken', 'apples', 'juice', 'pasta', 'bread']
```

.insert(index, item) - will insert a list item at a specified index.

```
>>> shopping_list = ['eggs', 'chicken', 'apples', 'juice', 'pasta']
>>> shopping_list.insert (2, 'onions')
>>> print (shopping_list)
['eggs', 'chicken', 'onions' 'apples', 'juice', 'pasta']
```

Practice: List Methods

From the list below remove 'dagger', choosing the method you like and assign it to variable, weapon, and replace the second item in the list to 'pocket knife'. Print out the modified list.

```
>>> backpack = ['xylophone', 'dagger', 'tent', 'bread loaf']
>>> weapon = backpack.pop(1)
>>> backpack.insert(1, 'pocket knife')
>>> print (backpack)
['xylophone', 'pocket knife', 'tent', 'bread loaf']
```

1.3 Join Method

The **join()** method parameter takes an iterable (List, Tuple, String, Dictionary) and returns a string in which the elements of sequence have been joined by string_characters of the method. Like so: ***string_characters.join(iterable)***

```
>>> string = 'lll'
>>> print ('o'.join(string))
lolol
```

Or

```
>>> alphabet_list = ['A', 'B', 'C', 'D', 'E']
>>> print (alphabet_list)
['A', 'B', 'C', 'D', 'E']
>>> print (' '.join(alphabet_list))
A B C D E
```

Practice: Join Method

Join to the items in the user name list, the email domain string to create a proper email address.

```
>>> username = ['jsmith', 'gmail.com']
>>> email_domain = '@'
>>> print (email_domain.join(username))
```

1.4 Len(), Del() Functions

Length function- len() - returns the number of items in an object.

To find the length of a list:

```
>>> shopping_list = ['eggs', 'chicken', 'apples', 'juice', 'pasta']
>>> print (len(shopping_list))
5
```

Delete function - del() - removes the item or an element at the specified index location from the list.

To delete an item of a list using the index:

```
>>> shopping_list = ['eggs', 'chicken', 'apples', 'juice', 'pasta']
>>> del(shopping_list[2])
>>> print (shopping_list)
['eggs','chicken','juice','pasts']
```

Practice: Len() Function

Using a while loop and an if statement; iterate through the list and if there is a 100, print it with its index number. i.e.: "There is a 100 at index no: 12"

```
>>> numbers_lst
=[10, 99, 98, 85, 45, 59, 65, 66, 76, 12, 35, 13, 100, 80, 95]
>>> i = 0
>>> while i < len(numbers_lst):
        if numbers_lst[i] == 100:
            print("There is a 100 at index no:", i)
        i = i+1
```

1.5 List Arithmetic

Lists can be joined together by adding them. They can also be multiplied by a number.

```
>>> list1 = [1, 2, 3, 4]
>>> list2 = ['I', 'declare', 'a', 'thumb', 'war']
>>> print(list1 + list2)
[1, 2, 3, 4, 'I', 'declare', 'a', 'thumb', 'war']
>>> list1 = [1, 2]
>>> print(list1 * 5)
[1, 2, 1, 2, 1, 2, 1, 2, 1, 2]
```

We cannot subtract or divide lists. Same goes for adding anything other than a list to a list.

1.6 Min(), Max() Functions

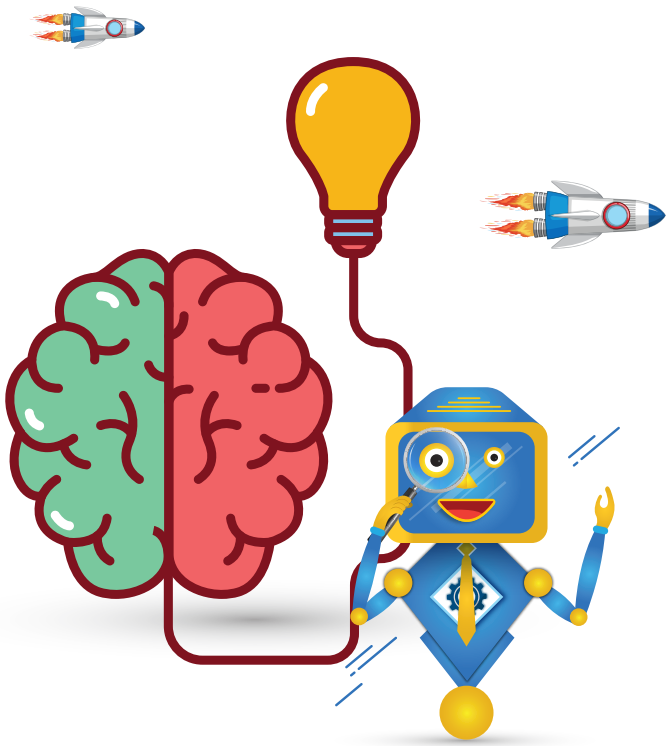
The **min()** and **max()** functions can be used to find the lowest or highest value in an iterable.

```
>>> nums = [541, 86, 2, 34, 98, 596]
>>> t = ("Students", "teachers", "Mutant", "creatures")
>>> print (max(nums), max(t))
596 'teachers'
>>> print (min(nums), min(t))
2 'Mutant'
>>> print("1st word in dict is: %s, and the last is: %s"
% (min(t),max(t)))
1st word in dict is: creatures, and the last is:
teachers
>>> max_lenght_word = max(t, key = len)
>>> print(max_lenght_word)
creatures
```

➤ Letters are ranked alphabetically, and lowercase letters come after uppercase letters, so t is higher in order than T.

➤ key (optional) - refers to argument function to customize the sort order. The function is applied to each item on the iterable.

Quick Tests



Quick Test: Favourites

Make a list of your favourite hobbies and name it games. Now make a list of your favourite foods and name it foods. Join the two lists and name the result favourites. Print the variable favourites.

Quick Test: Hidden List

From a hidden_list a variable you create, can you write a program that can tell you what the first and last items are. Also print out the length of the list or the number of items in the list.


LKD
EDUCATIONAL RESOURCES