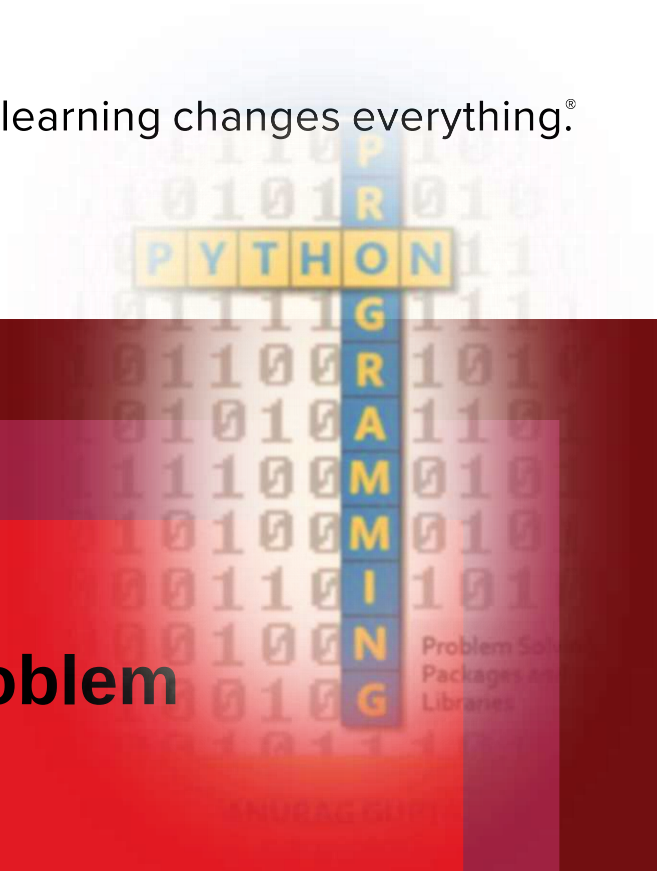
McGraw Hill

Because learning changes everything.®

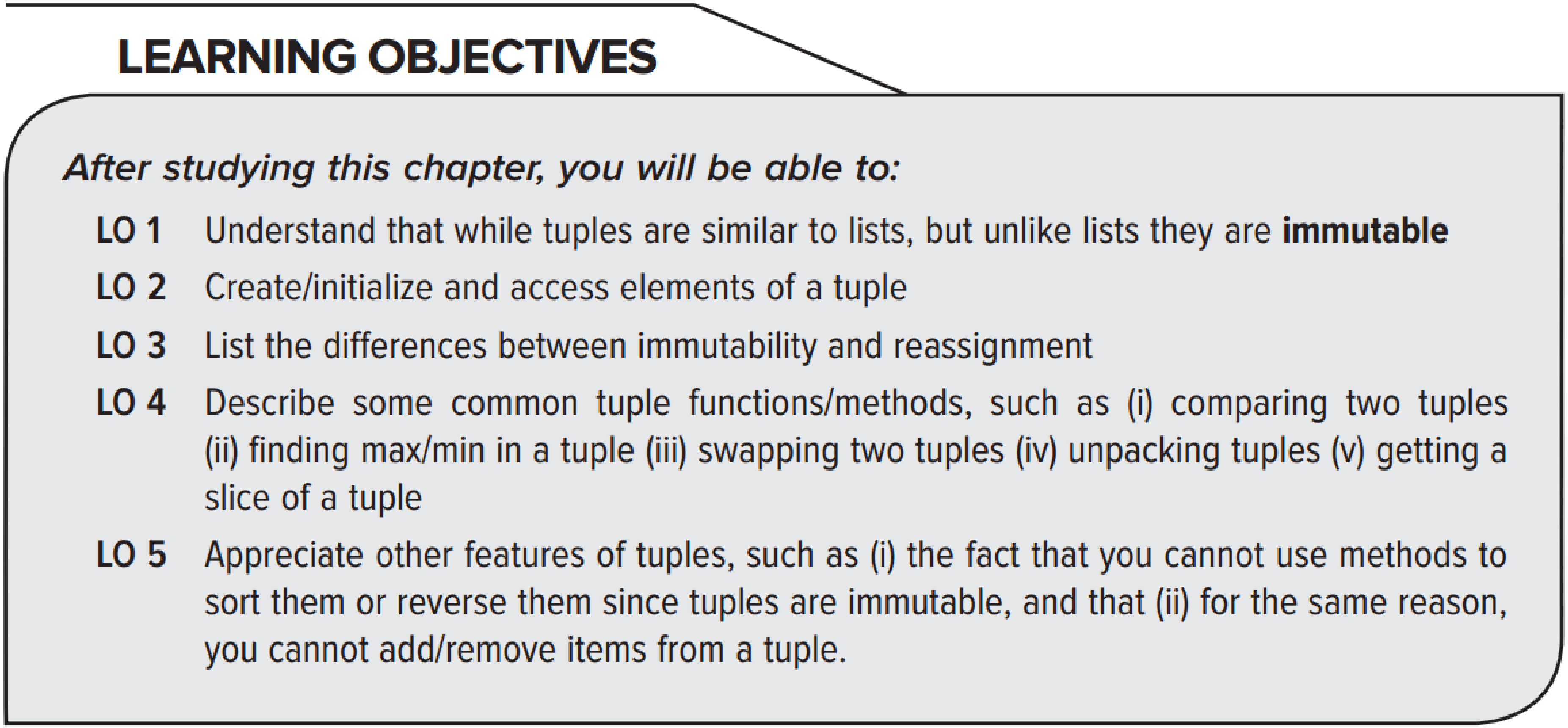


TITLE: Python Programming: Problem Solving, Packages and Libraries

Lecture PPT

Edition

Tuples – Learning Objectives



LEARNING OBJECTIVES

After studying this chapter, you will be able to:

LO 1 Understand that while tuples are similar to lists, but unlike lists they are **immutable**

LO 2 Create/initialize and access elements of a tuple

LO 3 List the differences between immutability and reassignment

LO 4 Describe some common tuple functions/methods, such as (i) comparing two tuples (ii) finding max/min in a tuple (iii) swapping two tuples (iv) unpacking tuples (v) getting a slice of a tuple

LO 5 Appreciate other features of tuples, such as (i) the fact that you cannot use methods to sort them or reverse them since tuples are immutable, and that (ii) for the same reason, you cannot add/remove items from a tuple.

Some important points regarding tuples -- 1

1. You can think of a tuple as an immutable list. You cannot change the contents of a tuple once it is created.
2. A tuple is also a sequence just like a list. Therefore, elements of a tuple also have an order similar to other sequences, such as strings and lists.
3. You can store any object in a tuple. You can even store another tuple inside a tuple.
4. The syntax of a tuple is slightly different from that of a list. Here, you must use round brackets instead of square brackets. The items inside a tuple are comma-separated just like in a list.
5. It is not mandatory to surround a tuple with round brackets, but using round brackets improves readability and helps the reader to easily identify a tuple.
6. Note that a tuple is created with parantheses, that is, (), but its elements are accessed by square brackets, that is, []. This is unlike a list, which is created with square brackets, that is [] and accessed with square brackets.

Some important points regarding tuples -- 2

7. When creating a tuple, you can and should initialize it with values because you cannot mutate it once it is created. You are of course free to create new tuples from existing ones.
8. Tuples also have a zero-base index, just like a list. Therefore, the first element of a non-empty tuple, say, myTuple is always of type myTuple[0].
9. Just as a list, you can use negative indices to access elements of a tuple.
10. Tuples can be sliced just like a list. When you slice a tuple, you will get a new tuple. Note that slicing a tuple does not alter the original tuple since tuples are immutable.
11. Being immutable, tuples do not have methods which can modify their contents. This means that methods, such as insert(), append(), extend(), remove(), and pop() are not available to tuples.

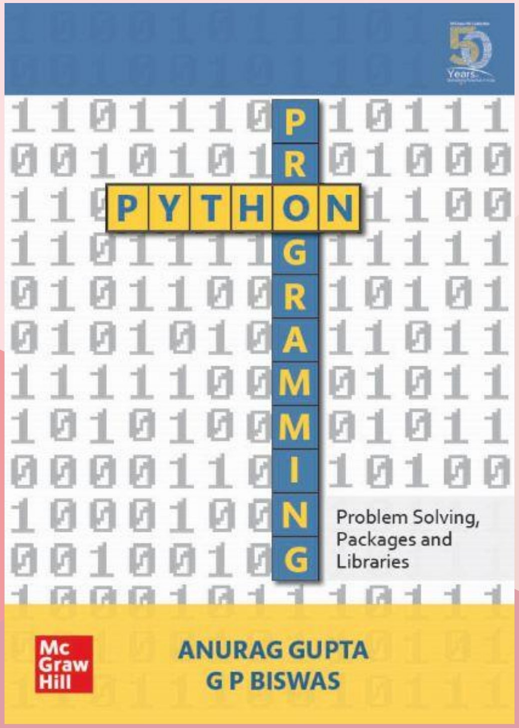
Some other noteworthy points regarding tuples

- An empty tuple (without any element) evaluates to bool False. If a tuple has at least one item (any item will do.), it evaluates to bool True.
- You can convert/ cast a tuple to a list or vice-versa. Use tuple (some_list) to convert the list some_list to a tuple. Similarly, use list(some_tuple) to convert a tuple named some_tuple to a list. You can say that the built-in function tuple(some_list) freezes a list. Similarly, you can say that the built-in function list(some_tuple), thaws a tuple.
- If you need to use a sequence as a dictionary key, then you can use a string or a tuple but not a list. If you have a list and want to use it as a key to a dictionary, then first cast it to tuple using tuple(some_list).
- Since tuples are immutable, they don’t have methods such as sort() or reverse().
- Removing individual tuple elements is not possible. If you want to remove some elements from a tuple, you need to create a new tuple from the existing one, with the undesirable elements removed.

Some common operations on tuples

- Getting length of a tuple t1 using len(t1)
- Concatenation of two tuples t1 and t2 by t1 + t2.
- Multiplication of a tuple t1 with a positive integer say n using t1 * n or n * t1.
- Membership operator, val in t1 (Here, val is a possible item in the tuple t1. If val exists in t1, this evaluates to True, else it is False).
- Using syntax for var in t1 (Where var is a local variable and t1 is a tuple. Used to iterate over the items of a tuple).
- Since many of the common tuple operations are common to other sequences, they don't need detailed explanation.

These operations on tuples are quite straight forward. The book provides example script which will clarify the use of these operations


PYTHON
PROGRAMMING
Problem Solving, Packages and Libraries
ANURAG GUPTA
G P BISWAS
Mc Graw Hill
50 Years

Thank You!

For any queries or feedback contact us at:

support.india@mheducation.com

1800-103-5875

www.mheducation.co.in