CHAPTER 8

TUPLES

Learning Objectives

After completing this chapter, the reader will learn to understand the usage of tuples and answer the questions related to the following

- Understand the need for tuple.
- Understand the creation of tuples.

- Understand the indexing and slicing of tuples.
- Know the basic tuple operations.
- Understand the built-in tuple functions.
- Understand the built-in tuple methods.

8.1. Introduction to Tuples

The characteristics of a tuple are given below:

- Tuples are used to store heterogeneous data types.
- Tuples are immutable and hence can be used as keys for the dictionary.
- Iteration of tuples, compared to lists,is considerably faster.

- *Tuples are ordered.* A tuple is ordered if the objects in it have a specified order, and that order will not change in the future.
- *Tuples are immutable.* Immutability is the hallmark of unchangeable-Tuples, which means that nothing can be changed, added, or removed once created.

- *Allow for duplication:* Indexed tuples allow the insertion of components with an identical value that is already in the tuple.
- *Tuples are efficient.* Tuples are built on unmodifiable structures, making them more straight forward regarding memory utilization and performance than lists. This is because the unmodifiable structures cannot be changed.

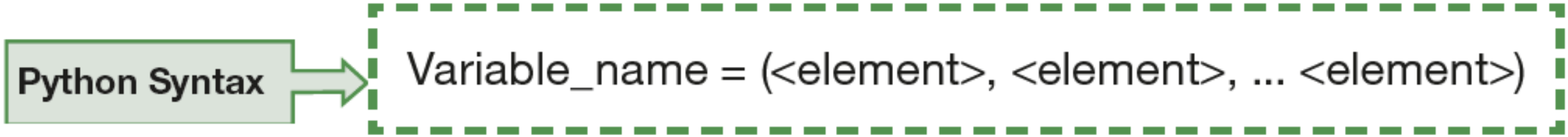
Table 8.1: Similarities Between Lists and Tuples.

Lists	Tuples
Lists can hold heterogeneous types	Tuples too can hold heterogeneous types
Lists can have duplicate elements	Tuples can accommodate duplicate elements
Lists' elements can be accessed using indexing and slicing	Tuples' elements too can be accessed using indexing and slicing
Lists have many functions, such as max,min, and len that are same as tuples.	Tuples too have many functions such as max,min,and len that are same as lists.

Table 8.2: Differences Between Lists and Tuples.

Lists	Tuples
Lists are mutable	Tuples are immutable
Lists' elements are enclosed by square brackets.	Tuples are items that are separated by commas enclosed by brackets (). Even though () is not necessary, it is better to have the brackets.
Lists' elements can be accessed using many functions and methods.	Tuples' elements too can be accessed by many functions and methods like list but are lesser in number compared to lists as tuples are immutable.

8.2 Creating a Tuple



A tuple can be empty like t1 =(). An empty tuple has an opening and closing parentheses () to form an Empty tuple.

8.3 Indexing and slicing in a tuple.

C	A	T	A	L	O	G	U	E
0	1	2	3	4	5	6	7	8
-9	-8	-7	-6	-5	-4	-3	-2	-1

Fig. 8.1 Tuples showing the indexing

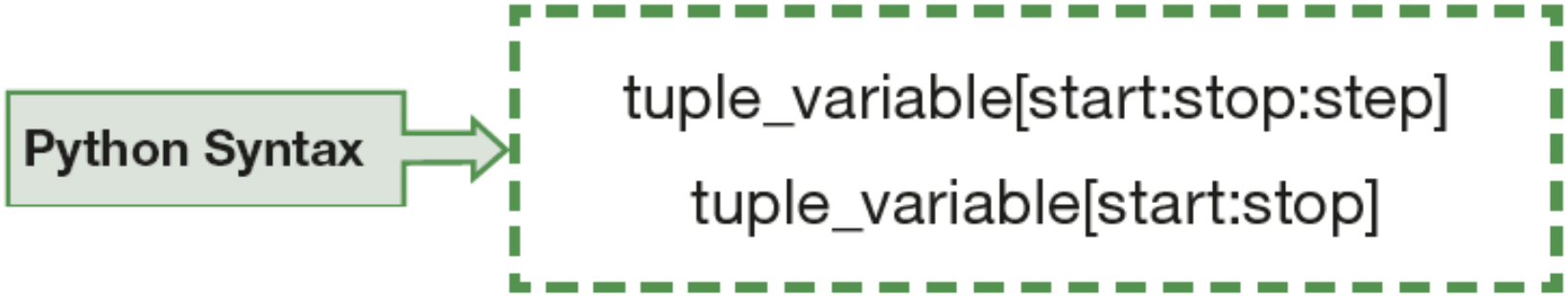


Table 8.3: Slicing Operators.

Indexing and Slicing Operator	Interpretation
tuple_variable[3]	Element of positive index 3
tuple_variable[-3]	Tuples can be referred to by negative index also
tuple_variable[-1]	Last element
tuple_variable[-2]	The second last element
0:5:1	starting from 0 to 5 and step 1 of the slice.
0: 5	starting from 0 to 4 with default step 1
2:5:1	starting from 2 to 5 and step 1 of the slice.
::4	starting from 0 with step 4

Table 8.3: Slicing Operators.

Indexing and Slicing Operator	Interpretation
tuple_variable[3]	Element of positive index 3
tuple_variable[-3]	Tuples can be referred to by negative index also
tuple_variable[-1]	Last element
tuple_variable[-2]	The second last element
0:5:1	starting from 0 to 5 and step 1 of the slice.
0: 5	starting from 0 to 4 with default step 1
2:5:1	starting from 2 to 5 and step 1 of the slice.
::4	starting from 0 with step 4

8.4 Operations in Tuples

Table 8.4: Summarization of Tuple Operations

Python Expression	Results	Description
len((6, 2, 9))	3	Length
(6, 8, 2) + (1, 5, 6)	(6,8,2,1,5,6)	Concatenation
('Hi!',) * 4	('Hi!', 'Hi!', 'Hi!', 'Hi!')	Repetition
3 in (1, 2, 3)	True	Membership

Concatenation

- Utilizing operators allows users to perform operations such as concatenating or multiplying tuples.

Membership Operator

Operator	syntax	Description
in	<tuple_element> in <tuple_ variable_name>	It yields true if it identifies a tuple element in the specified tuple; otherwise, it returns false.
not in	<tuple_element> not in <tuple_variable_name>	If that doesnot identify a tuple element in the specified tuple, it returns true; otherwise, it returns false.

8.5 Immutability in Tuples

To change the element, the following procedure can be used:

1) Convert a tuple to a list.

2) Perform the required operations like append, remove on lists.

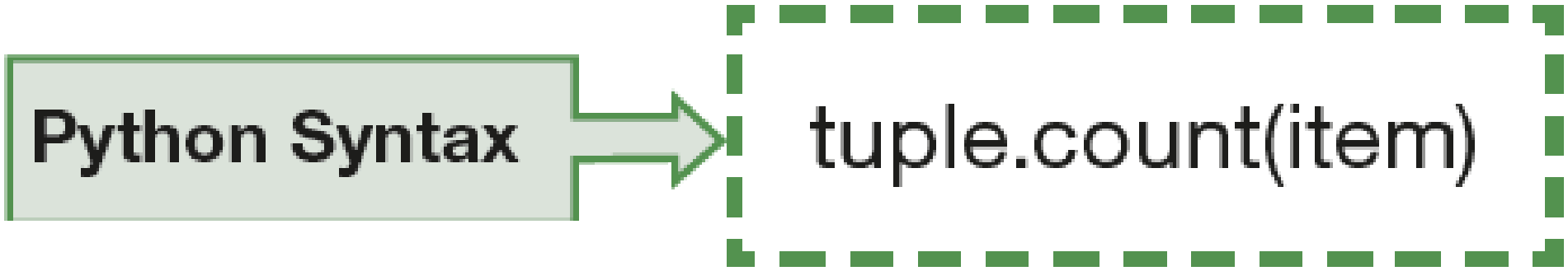
3) Convert the list back to the tuple.

Deleting a tuple

The del statement can be used to remove an entire tuple. To explicitly remove the tuple as a whole, use the del statement.

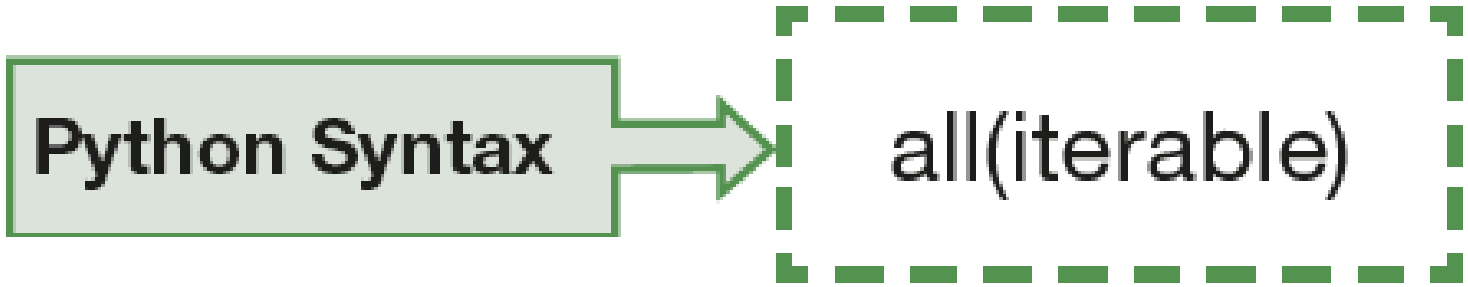
8.6 Built-in function and tuples

- **count()-**count() method provides a return result, the cumulative frequency of the value passed as a parameter.



8.6 Built-in function and tuples

- **all()-**used to check whether all items are True.



- **any() -**always return a value

- **enumerate ()** -enumerate(iterable, start=0) returns an enumerate object

- **min() -**returns the value that is the smallest among the many arguments of the tuple.

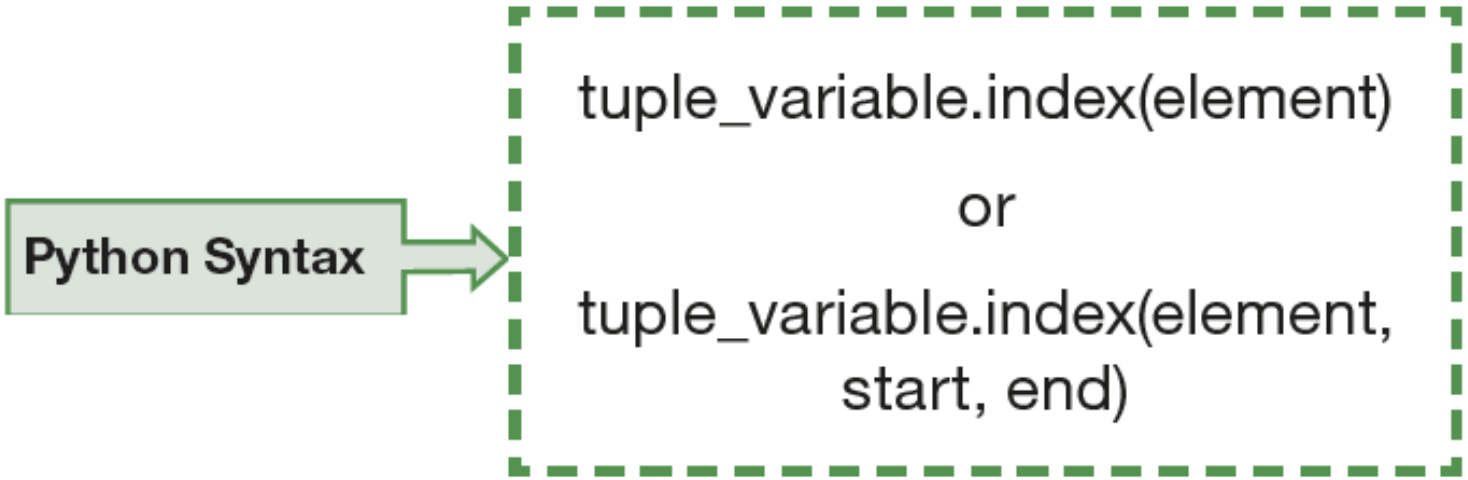
- **sum() -**finds the sum of all elements of the tuple

8.7 Built-in methods in tuples

Table. 8.6: Built-in Methods in Tuple.

Sr. No.	BIM (Built-in Methods)	Description
1.	Index(element) Or Index(element, beg, end)	This method returns the first instance of the given value or the index at which obj is found in the tuple. After the start and before the end, this function returns the index of the first occurrence of an element.
2	Sorted(tuple_variable_name)	Performs both sorting in ascending/descending order

- **index()**


Python Syntax
tuple_variable.index(element)
or
tuple_variable.index(element, start, end)

- **sorted()** Python organizes all the elements in the list either in an ascending or descending order and are placed in the iterable

8.8 Nested Tuples

Table 8.6: Examples of Nested Lists.

Description	Tuple
A nested tuple	t_6 = (("Vijayanagara","Karnataka", 560040), ("Amirstar","Punjab",143001), ("Khajuraho", "Madhya Pradesh", 471606))

Description	Tuple
A deeply nested tuple	t_7 = ("Levelone", ("Leveltwo ", ("Levelthree ", ("Levelfour ", " Levelfive ", 786))))
Another deeply nested tuple with single quotes.	t_8 = ('Apple', ('Bananas', ('Cherries', 'Dragon Fruit'), 'Eggfruit', 'Fig'), 'Guava', 'Honeydewmelon')

8.9 Selection

The evaluation of tuples in Python is accomplished through relational operators, which individually compare each character in the tuple. Each character is mapped to a corresponding numeric value in the computer's memory.

8.10 Looping with Tuples

In real-world situations, the ability to interact with every item on a tuple is required. Python's built-in for-loop helps handle each entry in a list.Python's for-loop implements process lists and other iterations.