**9.4. Data Structures Tuple**

* 1. Numbers Page 1
  2. String Page7
  3. List Page18
  4. Tuple Page 22
  5. Dictionary Page 25
  6. Set Page 31

**CRUD 🡪 Data Structure -> State Behavior**

DATA – **HOMO / HETERO**

ORDER – **INSERTION** / ASCENDING / DESCENDING / RANDOM

DUPLCATES – **ALLOWED** / NOT ALLOWED

CHNGEABLE – **IMMUTABLE** / MUTABLE

**List1 = [10, 10, 33, 14, 21, 12.5, True]**

**tup1 = (10, 10, 33, 14, 21, 12.5, True)**

Every value in Python has a datatype. Since everything is an object in Python programming, data types are actually classes and variables are instance (object) of these classes.

Type represents the kind of value and determines how the value can be used. All data values in Python are encapsulated in relevant object classes. Everything in Python is an object and every object has an identity, a type, and a value.

Like another object-oriented language such as Java or C++, there are several data types which are built into Python. Extension modules which are written in C, Java, or other languages can define additional types.To determine a variable's type in Python you can use the **type()** function.

**1. Memory allocation  
2. Sequence  
3. Immutable  
4. Builtin functions**

**Sequences :**

**There are 6 sequence types:**

1. String
2. List
3. **Tuple**
4. Bytearrays
5. buffers and
6. range **objects**

There are certain things you can do with **all sequence types**.

These operations include

* **indexing**
* **slicing**
* adding
* multiplying and
* checking for membership.

Python has built-in functions for :**finding the length of a sequence** and

**for finding its largest and smallest elements.**

**5.4 Tuples: CI**

A tuple is a sequence of **immutable** Python objects.

Tuples are sequences, just like lists.

The differences between tuples and lists are, the tuples c**annot be changed** unlike lists and tuples use **parentheses**, whereas lists use square brackets.

Creating a tuple is as simple as putting different comma-separated values.

Optionally you can put these comma-separated values between parentheses also.

For example −

tup1 = ('physics', 'chemistry', 1997, 2000)

tup2 = (1, 2, 3, 4, 5 )

tup3 = "a", "b", "c", "d"

The empty tuple is written as two parentheses containing nothing −

tup1 = ()

To write a tuple containing a single value you have to include a comma, even though there is only one value −

tup1 = (50,)

Like string indices, tuple indices start at 0, and they can be sliced, concatenated, and so on.

**Accessing Values in Tuples:**

To access values in tuple, use the square brackets for slicing along with the index or indices to obtain value available at that index. For example −

#!/usr/bin/python

tup1 =('physics','chemistry',1997,2000)

tup2 =(1,2,3,4,5,6,7)

print"tup1[0]: ", tup1[0]

print"tup2[1:5]: ", tup2[1:5]

When the above code is executed, it produces the following result −

tup1[0]: physics

tup2[1:5]: (2, 3, 4, 5)

## Updating Tuples:

Tuples are immutable which means you cannot update or change the values of tuple elements. You are able to take portions of existing tuples to create new tuples as the following example demonstrates −

tup1 =(12, 34, 56)

tup2 =('abc', 'xyz')

# Following action is not valid for tuples

# tup1[0] = 100;

# So let's create a new tuple as follows

tup3 = tup1 + tup2;

print tup3;

When the above code is executed, it produces the following result −

(12, 34.56, 'abc', 'xyz')

## Delete Tuple Elements:

Removing individual tuple elements is not possible. There is, of course, nothing wrong with putting together another tuple with the undesired elements discarded.

To explicitly remove an entire tuple, just use the del statement.

For example −

tup=('physics','chemistry',1997,2000);

printtup;

del tup;

print"After deleting tup : ";

printtup;

This produces the following result. Note an exception raised, this is because after  **deltup** tuple does not exist any more −

('physics', 'chemistry', 1997, 2000)

After deleting tup :

Traceback (most recent call last):

File "test.py", line 9, in <module>

printtup;

NameError: name 'tup' is not defined

## Basic Tuples Operations:

Tuples respond to the + and \* operators much like strings; they mean concatenation and repetition here too, except that the result is a new tuple, not a string.

In fact, tuples respond to all of the general sequence operations we used on strings in the prior chapter −

|  |  |  |
| --- | --- | --- |
| **Python Expression** | **Results** | **Description** |
| **len**((1, 2, 3)) | 3 | Length |
| (1, 2, 3) **+** (4, 5, 6) | (1, 2, 3, 4, 5, 6) | Concatenation |
| ('Hi!',)**\*** 4 | ('Hi!', 'Hi!', 'Hi!', 'Hi!') | Repetition |
| 3 in (1, 2, 3) | True | Membership |
| for x **in** (1, 2, 3): print x, | 1 2 3 | Iteration |

## Indexing, Slicing, and Matrixes

Because tuples are sequences, indexing and slicing work the same way for tuples as they do for strings. Assuming following input −

L =('spam','Spam','SPAM!')

|  |  |  |
| --- | --- | --- |
| **Python Expression** | **Results** | **Description** |
| L[2] | 'SPAM!' | Offsets start at zero |
| L[-2] | 'Spam' | Negative: count from the right |
| L[1:] | ['Spam', 'SPAM!'] | Slicing fetches sections |

## No Enclosing Delimiters:

Any set of multiple objects, comma-separated, written without identifying symbols, i.e., brackets for lists, parentheses for tuples, etc., default to tuples, as indicated in these short examples −

print'abc',-4.24e93,18+6.6j,'xyz';

x, y =1,2;

print"Value of x , y : ",x,y;

When the above code is executed, it produces the following result −

abc -4.24e+93 (18+6.6j) xyz

Value of x , y : 1 2

## Built-in Tuple Functions

Python includes the following tuple functions −

|  |  |
| --- | --- |
| **S.No.** | **Function with Description** |
| 1 | [**cmp**(tuple1, tuple2)](https://www.tutorialspoint.com/python/tuple_cmp.htm)Compares elements of both tuples. |
| 2 | [**len**(tuple)](https://www.tutorialspoint.com/python/tuple_len.htm)Gives the total length of the tuple. |
| 3 | [**max**(tuple)](https://www.tutorialspoint.com/python/tuple_max.htm)Returns item from the tuple with max value. |
| 4 | [**min**(tuple)](https://www.tutorialspoint.com/python/tuple_min.htm)Returns item from the tuple with min value. |
| 5  6  7 | [**tuple**(seq)](https://www.tutorialspoint.com/python/tuple_tuple.htm) Converts a list into tuple.  **Count**  **index** |