## 1. What is a Tuple?

A tuple holds an ordered collection of items. A tuple is an immutable object, i.e. we cannot change the items of the tuple.

### Creating a Tuple

- Created by enclosing elements within (round) brackets.
- Each item is separated by a comma.

**Code**

```python
a = 2
tuple_a = (5, "Six", a, 8.2)
print(type(tuple_a))
print(tuple_a)
```

**Output**

```
<class 'tuple'>
(5, 'Six', 2, 8.2)
```

### Tuple with a Single Item

**Code**

```python
a = (1,)
print(type(a))
print(a)
```

**Output**

```
<class 'tuple'>
(1,)
```

### Accessing Tuple Elements

Accessing

`tuple` elements is also similar to string and list accessing.

**Code**

```python
a = 2
tuple_a = (5, "Six", a, 8.2)
print(tuple_a[1])
```

**Output**

```
Six
```

2. Are Tuples mutable?

No, Tuples are immutable. They cannot be modified.

**Code**

PYTHON

```python
tuple_a = (1, 2, 3, 5)
tuple_a[3] = 4
print(tuple_a)
```

**Output**

```
TypeError: 'tuple' object does not support item assignment
```

3. How to Unpack a tuple?

The values of any sequence can be directly assigned to variables. The number of variables on the left should match the length of the sequence.

**Code**

PYTHON

```python
tuple_a = ('R', 'e', 'd')
(s_1, s_2, s_3) = tuple_a
print(s_1)
print(s_2)
print(s_3)
```

**Output**

```
R
e
d
```

4. What is the difference between List and Tuple?

| Basis for comparison | List | Tuple |
|---|---|---|
| Type | Lists are mutable | Tuples are immutable |
| Time Consumption | The list iteration is much slower compared to the tuple | The tuple iteration is much faster compared to the list |
| Appropriate Usage | It is very helpful in the case of deletion and insertion operations | It is comparatively helpful in the case of read-only operations, such as accessing elements |

### 5. What are Sets?

Sets are the unordered collection of items.

Every set element is

- Unique (no duplicates)
- Must be immutable

#### Creating a Set

- Created by enclosing elements within *{curly}* brackets.
- Each item is separated by a comma.

#### Code

PYTHON

```python
a = 2
set_a = {5, "Six", a, 8.2}
print(type(set_a))
print(set_a)
```

#### Output

```
<class 'set'>
{8.2, 2, 'Six', 5}
```

Set items need not be in the same order as defined.

#### No Duplicate Items

Sets contain unique elements

#### Code

PYTHON

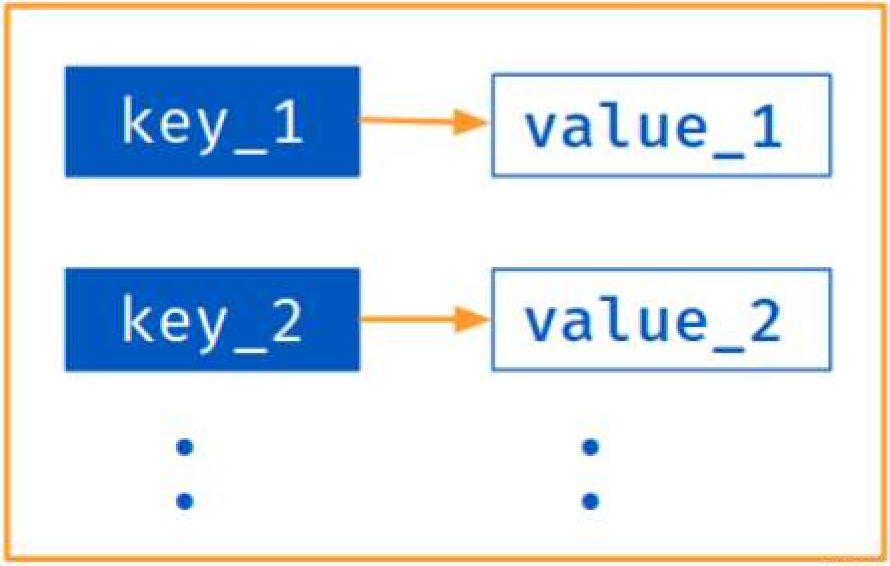
```python
set_a = {"a", "b", "c", "a"}
print(set_a)
```

#### Output

```
{'b', 'a', 'c'}
```

### 6. What is a Dictionary?

A Dictionary is an unordered collection of items. Every dictionary item is a **Key-Value** pair.



7. How to create a Dictionary?

A dictionary is created by enclosing items within **{curly}** brackets.

Each item in the dictionary has a Key-Value pair separated by a **comma**.

Code

PYTHON

```
dict_a = {
  "name": "Teja",
  "age": 15
}
```

In the above dictionary, the

- `keys` are `name` and `age`
- `values` are `Teja` and `15`

8. How to get the keys of a dictionary?

The keys() method allows us to get the keys of dictionary. It returns a view object of the type

`dict_keys` that holds a list of all keys.

Code

PYTHON

```
dict_a = {
  "name": "Teja",
  "age": 15
}
print(dict_a.keys())
```

Output

```
dict_keys(['name', 'age'])
```

9. Why List can't be used as a key in the dictionary/ Are the dictionary keys mutable?

A Python dictionary can have only keys of immutable types. Lists cannot be used as keys in a dictionary because they are mutable.

10. How to create an empty dictionary?

Using `dict()` built-in function

code

PYTHON

```python
dict_a = dict()
print(type(dict_a))
print(dict_a)
```

Output

```
<class 'dict'>
{}
```

Using `{}` symbol

code

PYTHON

```python
dict_a = {}
print(type(dict_a))
print(dict_a)
```

Output

```
<class 'dict'>
{}
```

11. How to combine two dictionaries?

We can combine two dictionaries using

`update()` method:

Syntax

```
dictionary.update(iterable)
```

code

PYTHON

```python
dict_1 = {'a': 1, 'b': 2}
dict_2 = {'c': 3, 'd': 4}

dict_1.update(dict_2)
print(dict_1)
```

Output

```
{'a': 1, 'b': 2, 'c': 3, 'd': 4}
```

### 12. What are the differences between Lists, Tuples, Sets and Dictionaries?

| Lists | Tuples | Sets | Dictionaries |
|---|---|---|---|
| A list is an ordered collection of data | A tuple is an ordered collection of data | A set is an unordered collection | A dictionary is an unordered collection of data that stores data in Key-Value pairs |
| Lists are mutable | Tuples are immutable | Sets are mutable | Dictionaries are mutable |
| Lists are enclosed within square braces `[]` | Tuples are enclosed within parenthesis `()` | Sets are enclosed within curly brackets `{}` | Dictionaries are enclosed within curly brackets `{}` in the form of Key-Value pairs |
| Example: [1, 2, 3, 4, 5] | Example: (1, 2, 3, 4, 5) | Example: {1, 2, 3, 4, 5} | Example: { a:1, b:2, c:3, d:4, e:5} |
| List allows duplicate elements | Tuple allows duplicate elements | Set will not allow duplicate elements | Dictionary doesn't allow duplicate keys |
| List can be created using `list()` function | Tuple can be created using `tuple()` function | Set can be created using `set()` function | Dictionary can be created using `dict()` function |

○ MARK AS COMPLETED

Submit Feedback

Notes Discussions