

Collection Data Types in Python

Python has 4 built-in data structures that can be used to hold a collection of objects, they are list, tuple, set, and dictionary. They can be distinguished into mutable, immutable, set type, and mappings respectively.

Lists

Lists are ordered mutable sequences that can be changed after they have been created by adding, removing, or changing objects. Lists can be declared by using square brackets “[]” following the variable name.

```
prices = [20, 40, 10, 5]
```

Tuples

Tuples are ordered immutable sequence that stores multiple items in a single variable, meaning it cannot be changed after it has been created. A tuple can be created by a pair of parenthesis and comma-separated objects, following the variable name.

```
fruits_tup = ("apple", "banana", "cherry")
```

Sets

Sets are an unordered immutable set of unique objects that do not support duplicated objects and as such, they cannot be indexed. Mathematical operations such as intersection, union, difference, and symmetric difference, can be carried out on set data types. Sets can be created by using curly brackets following the variable names or using the `set()` constructor.

```
letters = {'b', 'a', 't', 'i'}
```

```
letters = set(('b', 'a', 't', 'i'))
```

Dictionary

A dictionary is an unordered set of key/value pairs. Each unique key has a value associated with it in the dictionary, and dictionaries can have any number of pairs. The values associated with a key can be any object. Like sets, dictionaries are unordered.

```
age = {"Mike": 10,  
      "Leo": 19,  
      "Dea": 5  
}
```

```
print(age)
```

```
{'Mike': 10, 'Leo': 19, 'Dea': 5}
```