

In this module we will be covering dictionaries in PYTHON, how we define and use sets and how we serialize objects.

Dictionaries

This is an object that stores a collection of data. Each element has two parts - a key and an index. You use the key to locate a specific value. We call this key-value pairs.

We create a dictionary by enclosing the elements inside of a set of curly braces { }.

We retrieve a value from a dictionary:

```
dictionary_name[key]
```

in And not in Operators

We use the *in* And *not in* operators to test for a value in a dictionary.

Adding Elements to a Dictionary

```
dictionary_name[key] = value
```

Deleting Elements in a Dictionary

```
del dictionary_name[key]
```

len Function

You can use the function len to get the number of elements in a dictionary.

Creating an Empty Dictionary

You might want to create an empty dictionary where you then add elements to it as the program executes.

Using the for Loop

```
for var in dictionary:
```

```
statement
```

```
statement
```

```
etc.
```

Dictionary Methods

clear - clears the contents of the dictionary

get - gets the value associated with a specified key

items - returns all the keys in a dictionary and their associated values as a sequence of tuples

keys - returns all the keys in a dictionary as a sequence of tuples

pop - returns the value associated with a specified key and removes that key-value pair from the dictionary

popitem - returns a randomly selected key-value pair as a tuple from the dictionary and removes that key-value pair

from the dictionary

values - returns all the values in the dictionary as a sequence of tuples

Sets

A set contains a collection of unique values and works like a mathematical set. 1) all items in a set must be unique (no same values); 2) sets are unordered; 3) the elements that are stored in a set can be of different data types.

We use the built-in function *set* to create a set:

```
myset = set()
```

You can get the number of elements in the set by using the *len* function.

You can add and remove elements in the set. You use the *add* Method to add an element to a set. You use the *remove* Method or the *discard* Method to remove an item from the set.

You can use the *for* Loop to iterate over a set

```
for var in set:
```

```
    statement
```

```
    statement
```

```
    etc.
```

You can use the *in* And *not in* Operators to Test for a Value in a set.

You can use the *union* Method to get the union of two sets.

```
set1.union(set2)
```

You can use the *intersection* Method to get the intersection of two sets.

```
set1.intersection(set2)
```

You can use the *difference* Method to get the difference of two sets. (elements that appear in set 1 but do not appear in set 2).

```
set1.difference(set2)
```

You can use the *symmetric_difference* Method to get the symmetric difference of two sets (elements that are not shared by the sets).

```
set1.symmetric_difference(set2)
```

Serializing Objects

Serializing an object is the process of converting the object to a stream of bytes that can be saved to a file for later retrieval. This is called "pickling". The PYTHON library provides a module named *pickle* that holds various functions that serialize objects.