

## Dictionaries

- Dictionaries are a different kind of collection. They are Python's built-in mapping type. A map is an unordered, associative collection. The association, or mapping, is from a key, which can be any immutable type, to a value, which can be any Python data object.
- Dictionaries are unordered ,and they use keys to locate the value pairs
- For example, assigning a dictionary a value using the key `eng2sp['one'] = 'uno'`
- Just like lists, dictionaries are mutable, eg

```
inventory = {'apples': 430, 'bananas': 312, 'oranges': 525, 'pears': 217}
```

```
inventory['pears'] = 0
```

- The len function also returns the number of value key pairs
- We can use the word list to convert an inventory into a list:  

```
ks=list(inventory.keys())
```
- Tuples are useful for getting the key and value while looping through dictionaries.
- The in and not in can test if a key is present or not in a dictionary `print("oranges" in inventory)`
- Dictionaries are mutable
- Just like aliasing in lists, changing a value in one object affects the change in the other object.

Method	Parameters	Description
--------	------------	-------------

keys	none	Returns a view of the keys in the dictionary
------	------	--

values	none	Returns a view of the values in the dictionary
--------	------	--

items	none	Returns a view of the key-value pairs in the dictionary
get	key	Returns the value associated with key; None otherwise
get	key, alt	Returns the value associated with key; alt otherwise