Dictionaries

can code communities

Dictionaries

In programming, a dictionary is a collection of keys and values.

We can use them to store information that is associated with each other.

They are similar to a non-programming dictionary that is a book of definitions in a language.

- You can think of a key as a word, and a value as a definition.

However, this is just one use for dictionaries. They are a lot more versatile.

Dictionary Methods

`keys()` Returns a list of the dictionary's keys

`values()` Returns a list of the dictionary's values

`clear()` Removes all elements from the dictionary

`copy()` Returns a copy of the dictionary

`pop()` Removes the element with the specified key

`popitem()` Removes the last inserted key-value pair

`items()` Returns a list containing a tuple for each key-value pair

`update()` Update the dictionary with the specified key-value pairs

https://www.w3schools.com/python/python_ref_dictionary.asp

Dictionaries vs Lists

```
list1 = ['a', 'b', 'c', 'd', 'e']
```

```
dict1 = {0: 'a', 1: 'b', 2: 'c', 3: 'd', 5: 'e'}
```

They're called the same way:

```
list1[3]  # returns 'd'
```

```
dict1[3]  # returns 'd'
```

They're updated the same way

```
list1[3] = 'z'
```

```
dict1[3] = 'z'
```

Example

Write some code that takes two lists and converts them into one dictionary.

In:

```
list1 = ['one', 'two', 'three']

list2 = [4, 10, 30]
```

Out:

```
{'one': 4, 'two': 10, 'three': 30}
```

Exercise

Write a dictionary that contains five words and their definitions. Then have your code print the word and their definition one at a time.

Hint: Use the items() method

Dictionaries as Datasets

Sometimes dictionaries are used to represent a dataset.

Each key in the dictionary is a category.

Each value is a list of values for that category.

One datapoint is represented by the n-th index of each value list.

```
fruits = {"name":["apple", "banana", "strawberry", "blueberry"],\
          "size":["medium", "large", "small", "small"],\
          "flavor":["tart", "sweet", "tart", "tart"],\
          "region":["central asia", "southeast asia", "europe", "north america"]
}
```

Exercise

Create a dictionary for an automobile including make, model, year, number of doors, and number of cylinders.

Example

In statistics, the mode is the value that appears most frequently in a dataset.

For example, in this list: `[1,2,4,1,3,4,1,1]` the mode is 1

Write some code that uses a dictionary to calculate the mode of a list.

Exercise

Suppose you have a list of employee records that contain the following information for each employee: name, job title, salary. The records are stored as a list of dictionaries.

Use this list to create a dictionary where the keys are the job titles and the values are the average salaries for each job title.

Example:

```
records = [{'name': 'Bob', 'title': 'manager', 'salary': 50000},\
           {'name': 'Alice', 'title': 'developer', 'salary': 60000},\
           {'name': 'David', 'title': 'developer', 'salary': 65000}]
```

Output: `{'manager': 50000, 'developer': 62500}`

Uses for Dictionaries

- Used as a data structure, when each item needs to have a label (key) and value
  - Keys in dictionaries are more descriptive labels than indexes in a list
  - Easily access specific elements by name (key)
- Used to pass values into another structure

Python Collections

Type	Brackets	Order	Immutable/ Mutable	Allow Duplicates?	Associative? (Key/Value pairs)
List	[ ]	Ordered	Mutable	Yes	No
Tuple	( )	Ordered	Immutable	Yes	No
Set	{ }	Unordered	Immutable*	No	No
Dictionary	{ }	Ordered**	Mutable	No***	Yes

Footnotes:

* Set items are immutable, but you can add and remove items.

** Dictionaries are ordered, but only since Python version 3.6 - they used to be unordered.

*** Dictionaries allow duplicate values, but NOT duplicate keys.

Resources

https://www.w3schools.com/python/python_dictionaries.asp