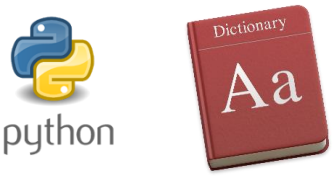
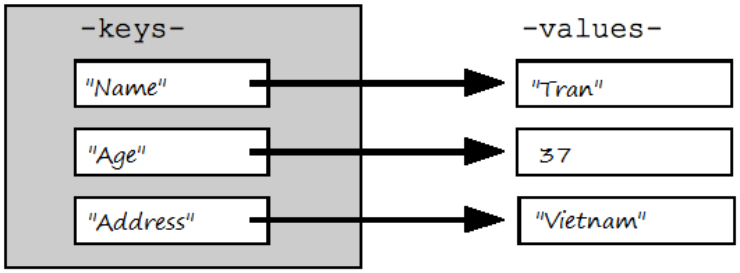
Dictionaries Intro



In python, dictionaries are a data type that is *similar* to a **list** but uses **keys** to access information instead of number an i**ndex**. Example:



You can think of a dictionary as a fancy **unordered** list where **you get to create specific names** for each location of data instead of using an index number.

This is great for humans. We like identifying things by name rather than numbers. If I want to look up information about "Tommy", I can ask for information about Tommy instead of having tommy's information stored in index box 112343.

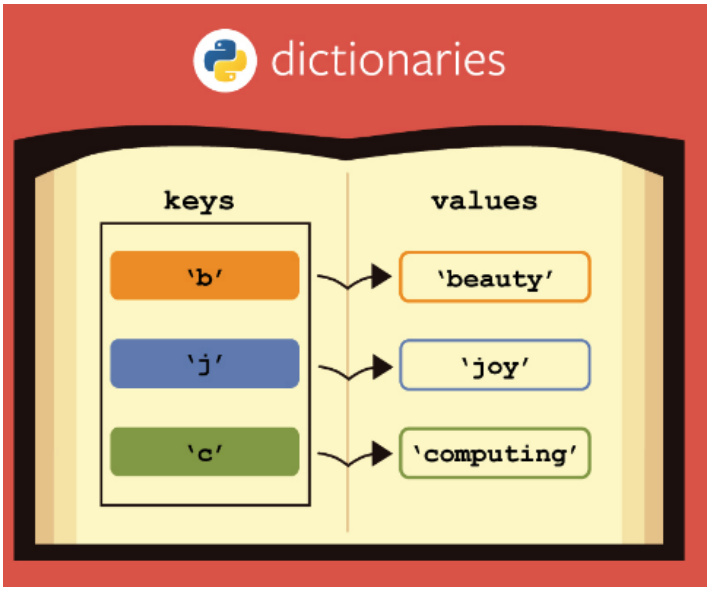
The values in a dictionary aren't numbered - they aren't in any specific order, either. You *can* add, remove, and modify the values and keys in dictionaries.

The **dictionary** data type in Python is known in other programming languages as **associative arrays**, **hashes**, or **mapping**. The concept is the same for all languages and is important to understand.

Creating a dictionary:

Properties of Dictionary:

- There is **no defined order** to keys and values (you cannot sort a dictionary using keys)
- Can't **locate values with an index** (no slicing etc).
- Only one entry per `key` is allowed.
- The `values` in the dictionary can be any data type (lists, strings, ints, floats)
- `keys` must be **immutable** like numbers, tuples or strings.



Exericse#1 – Enter the following examples into an IDE of your choice. Examine the input and output carefully to see what is going on. Please type them in (so you get practice with the syntax). **Don't copy and paste**.

Creating Dictionaries and getting Key Values:

```
Dict1 = {'Tim': 18,'Charlie':12,'Tiffany':22,'Robert':25}
print (Dict1['Tiffany'])
print (Dict1['Robert'])
```

Adding to a Dictionary:

From previous `Dict1` try:

```
Dict1['Phil']=31
Print(Dict1)
```

Enter the following:

```
basket = { 'oranges': 12, 'pears': 5, 'apples': 4 }
basket['bananas'] = 5
print(basket)
```

`Update` method:

```
dict = {'Tim':18,'Charlie':12,'Tiffany':22,'Robert':25}
dict.update({"Sarah":9})
print (dict)
```

`del` method. Deleting values from a dictionary:

`try:`

```
basket = {'apple': 4,'bananda':2,'tomoato':1,'carrot':9}
del basket['apple']
print (basket)
```

Exercise#1 continued
Iterating through a dictionary:

Try:

```
backpack = {'binder':1, 'pencils':5, 'eraser':1, 'textbook':3}
for x in backpack:
  print(x)

for x in backpack:
  print(backpack[x])
```

Also Try:

```
backpack = {'binder':1, 'pencils':5, 'eraser':1, 'textbook':3}

print(backpack.values())
print(backpack.keys())
```

Exercise#2 (you try)

Create a **dictionary** that has movie titles for **`keys`** and amount of money each moving makes in its opening weekend (in millions) for each **`value`**. Use at least 5 movies.

Then do the following:

1. Print out just the keys
2. Print out just the values
3. Delete a movie from your dictionary
4. Add a movie from your dictionary
5. Iterate through your dictionary values and print out the movie with the highest money made in opening weekend.
6. Create a **list** from your dictionary that has the movies in order of how much money each made (greatest to least)

A dictionary of Lists:

A dictionary connects two pieces of information. Remember **`values`** in a dictionary can be any kind of data type. Let's try using **`lists`** as **`values`**. This a common practice.

Exercise#3

The program stores people's favorite numbers, and displays them.
Please try the following: type it in and see how the input and output relate.

```
favorite_numbers = {'eric': [3, 11, 19, 23, 42],
                    'dave': [2, 4, 5],
                    'willie': [5, 35, 120],
                    }

print(favorite_numbers)
print(favorite_numbers['dave'])
print(favorite_numbers['dave'][0])
print(favorite_numbers['dave'][1])
print(favorite_numbers['dave'][2])
print(favorite_numbers['willie'][2])

favorite_numbers['dave'][1]=9
print(favorite_numbers)
```

Here is another way to create a dictionary with list as values:

Please try:

```
list1 = ['a', 'b', 'c']
list2 = [1, 2, 3, 4]
list3 = ['Jim', 'Tom', 'Bill'
my_dict = {'letters': list1, 'numbers': list2, 'names': list2}

print(my_dict['letters'])
print(my_dict['letters'][0])
print(my_dict['letters'][1])
print(my_dict['numbers'])
print(my_dict['numbers'][2])
print(my_dict['names'][2])
print(my_dict)
```

Exercise#4

The following **dictionary** is supposed to represent a characters attributes in a fantasy video game.

Given the following dictionary:

```
inventory = {
              'gold' : 500,
              'pouch' : ['flint', 'twine', 'gemstone'],
              'backpack' : ['flute','dagger','bedroll','bread loaf']
             }
```

Try to do the following:

1. Add a key to inventory called 'pocket'.
2. Set the value of 'pocket' to be a **`list`** consisting of the following items: 'seashell', 'berries', 'Lint', 'sand'.
3. `.sort()`the items in the list stored under the 'backpack' key.
4. Then `.remove`('dagger') from the list of items stored under the 'backpack' key.
5. Change the number stored under the 'gold' key to 250.

Exercise#5

Use all the dictionary methods below on the character's inventory dictionary (from Exercise#3): Make sure you show some output that is the result of how you used each method. Add a print statement to clarify to the user what each method does.

`mydict.clear`	`mydict.get`	`mydict.pop`	`mydict.update`
`mydict.copy`	`mydict.items`	`mydict.popitem`	`mydict.values`
`mydict.fromkeys`	`mydict.keys`	`mydict.setdefault`	

Exercise#6

- Create a new dictionary, where the keys of the dictionary are the names of different mountains in British Columbia, the values of the dictionary should be a **list** of each mountain's: elevation in meters, and then in feet:
  `Mountains={'everest': [8848, 29029]}`
- Use google to get the information.

Then:

1. Print out just the mountains' names.
2. Print out just the mountains' elevations in meters.
3. Print out a "neatly formatted table that shows the name of each mountain, elevation in meters, and elevation in feet.

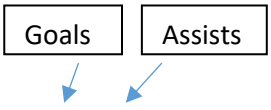
Exercise#7

Write a python program to sum all the **values** in a dictionary.

Exercise#8

Write a Python **function** that takes in a team name as a parameter and outputs the total number of goals *the team* has scored. Make up several **dictionary** teams to choose from. Each dictionary should use the template below.

Example:




```
Team1 = {'jim' : [1, 2], 'alex' : [4, 5], 'todd' : [7, 8]}
```

Expected result for this of the function for the team above would be: **12** goals.

Exercise #9

Write a Python program to find and display the **top three** selling items in a bulk food shop (in grams). Create an extended dictionary like the one below with at least 10 items:

```
grams_sold = {'almonds': 1145.50, 'dried_pineapple':3435.50, 'chocolate_chips':
1841.30, 'candy':7855.21, 'beans': 4327.24}
```

Expected Output for example above:

```
candy  7855.21
beans  4327.24
dried pineapple  3435.50
```

Exercise #10

Write a Python program to **sort** a dictionary from highest to lowest **`value`**.

```
Sample data = {'Math':81, 'Physics':83, 'Chemistry':87}
Expected output: [('Chemistry', 87), ('Physics', 83), ('Math', 81)]
```

Exercise#11

A store would like to keep track of it's inventory. Create a **dictionary** of 10 items that includes the following information (**name of food item, price in dollars**, and **amount in stock**) like the example below:

```
"banana": [1, 15]        #(price in $,  number of items in stock)
"apple":  [2, 35]
"orange": [1.5, 27]
"pear":   [3, 36]
```

Create a program that can print out a key along with its price and stock information when prompted by a user to get information about a particular item. Make sure to have the output in the following format **`EXACTLY`**:

```
What item do you want information about? apple
APPLE
price: $2
In stock: 35
```

Now, create a program that can determine how much money you would make if you sold **all** of your food in the store.

Exercise#12

TeacherEase
Software for the Digital Classroom

Note that the **values** of a dictionaries can be dictionaries themselves (nested dictionaries)!

Create a dictionary called **`Marks_Book`** that **contains** the *dictionaries* shown below. Then create a user interface that will allow a user to generate the following:

1. A list of *all* the students in the class.

2. A list of any students marks in **one** of the categories shown below (homework, quizzes, tests, labs). example: `Dave's test marks are [75.0, 90.0]`

3. An average for any student's marks in a particular category: `Alice' homework average is 97`

Make sure you interface is "user friendly". So that anyone running your program can use it.

Example:

```
"welcome to Marks Book!...what would you like to do?":

Press:

1 - if you want to get a list of your students.
2 – if you want to get a list of a student's grades in a grade category.
3 - if you want to get a student's average for a particular grade category.
```

```
lloyd = {
    "name": "Dave",
    "homework": [90.0, 97.0, 75.0, 92.0],
    "quizzes": [88.0, 40.0, 94.0],
    "tests": [75.0, 90.0]
}
alice = {
    "name": "Alice",
    "homework": [100.0, 92.0, 98.0, 100.0],
    "quizzes": [82.0, 83.0, 91.0],
    "tests": [89.0, 97.0]
}
tyler = {
    "name": "Tyler",
    "homework": [0.0, 87.0, 75.0, 22.0],
    "quizzes": [0.0, 75.0, 78.0],
    "tests": [100.0, 100.0]
}
```