

Exercise 1

1. create a dictionary which contains 5 keys: first_name, last_name, age, number, address. For the values just fill an empty string.
2. ask the user to complete those values
3. according to the user age tell whether he is over 21 or not.

Exercise 2 ¶

1: Capitals Make a dictionary containing three countries and their capitals. One key-value pair might be 'France': 'Paris'.

- Use a loop to print a sentence about each country, such as The capital of France is Paris
- Use a loop to print the name of each capital included in the dictionary.
- Use a loop to print the name of each country included in the dictionary

Exercise 3

In this exercise we're going to ask to the user to enter a name and give him the age of the person it's related to. for example :{'Mark':32, ' Nick':27}

But if the name does not exist we are going to ask to the user to create a new key-value pair and add it to the dictionary.

Bonus : let the user search for a name or an age and show every people with that age

Exercise 4

1) Create a variable called my_groceries with the values banana, orange, apple, pear 2) Copy these two dictionaries:

Entrée [3]:

```
stock = { "banana": 6, "pear ": 2, "apple": 0, "orange": 32 }  
prices = { "banana": 4, "pear ": 2.8, "apple": 2, "orange": 1.5 }
```

3. the variable my_groceries is a dictionary all the values are initialized at 0
4. ask to the user how many items does he want from each keys
5. add this value to the dictionary
6. after the user is done, calculate how much does he have to pay
7. BONUS:

- 1: when the user picks for instance 3 bananas then remove 3 from the banana stock
- 2: if the user asks a number of items that is higher than the one in stock tell him to take less and show him the number in stock

Exercise 5

1. make the user write a sentence.
2. using dictionary create a key for each letter, there is in his sentence
3. the value of those keys should be the number of time the letter appeared in the sentence

Exercise 6 - mini (hard) project

1. create a list of dictionaries that looks like this

Entrée [6]:

```
network = [  
    {  
        'username': 'David',  
        'password': '12345',  
        'email': 'david@super.com',  
        'message_received': [],  
        'message_sent': []  
    },  
    {  
        'username': 'Mark',  
        'password': '23412',  
        'email': 'mark@super.com',  
        'message_received': [],  
        'message_sent': []  
    },  
    {  
        'username': 'Jacob',  
        'password': '23821',  
        'email': 'jacob@super.com',  
        'message_received': [],  
        'message_sent': []  
    },  
    {  
        'username': 'Joseph',  
        'password': '62797',  
        'email': 'joseph@super.com',  
        'message_received': [],  
        'message_sent': []  
    }  
]
```

Part 1

1. ask the user to give his credentials.
2. Find him in the list and check if both of the username and the password are correct.
3. if the password is not correct tell him to enter it again (give him 3 trials).
4. Then create a variable called connected user which value is going to be the user's dictionary.
5. BONUS : if the username is not right, ask the user if he want's to create an account. And create it.

Part 2

A message looks like that

Entrée [5]:

```
{
  'From': '',
  'To': '',
  'content': ''
}
```

Out[5]:

```
{'From': '', 'To': '', 'content': ''}
```

1. Now ask the user if he wants to read his messages or send one.
2. if he want's to read them show all the content of his 'message_reveived'.
3. if he wants to send a message first ask him to who.
4. then ask him what he wants to write
5. then complete the dictionary of a message ('from' is the connected user, 'to' is the one he wants to send it to)
6. finally append this message to the 'message_sent' of the connected user and to the message received of the receiver.

Entrée [8]:

Entrée []: