

# Dictionnaires

## 1. Définition et création

Les dictionnaires ressemblent aux listes( ils sont modifiables comme elles), mais ce ne sont pas des séquences. Les éléments que nous allons y enregistrer ne seront pas disposés dans un ordre immuable. En revanche, nous pourrons accéder à n'importe lequel d'entre eux à l'aide d'un index que l'on appellera une clé, laquelle pourra être alphabétique ou numérique.

Comme dans une liste, les éléments mémorisés dans un dictionnaire peuvent être de n'importe quel type( valeurs numériques, chaînes, listes ou encore des dictionnaires, et même aussi des fonctions).

Exemple :

```
In [8]: dico1={}
dico1['nom']='Paris-Orly Airport'
dico1['ville']='Paris'
dico1['pays']='France'
dico1['code']='ORY'
dico1['gps']=(48.7233333,2.3794444)

print(dico1)
print(dico1['code'])
print(dico1['gps'])

{'nom': 'Paris-Orly Airport', 'ville': 'Paris', 'pays': 'France', 'code': 'ORY', 'gps': (48.7233333, 2.3794444)}
ORY
(48.7233333, 2.3794444)
```

Remarques :

- Des accolades délimitent un dictionnaire.
- Les éléments d'un dictionnaire sont séparés par une virgule.
- Chacun des éléments est une paire d'objets séparés par deux points : une clé et une valeur.
- La valeur de la clé 'code' est 'ORY'.
- La valeur de la clé 'gps' est un tuple de deux flottants.

### Exercice 1:

Voici des données concernant l'aéroport international de Los Angeles :

- "Los Angeles International Airport","Los Angeles","United States","LAX",33.94250107,-118.4079971

En utilisant les mêmes clés que dans l'exemple précédent, créer le dictionnaire 'dico2' qui contiendra les données ci-dessus

```
In [9]: #Réponse
dico2={}
dico2['nom']='Los Angeles International Airport'
dico2['ville']='Los Angeles'
dico2['pays']='United States'
dico2['code']='LAX'
dico2['gps']=(33.94250107,-118.4079971)

print(dico2)

{'nom': 'Los Angeles International Airport', 'ville': 'Los Angeles', 'pays': 'United States', 'code': 'LAX', 'gps': (33.94250107, -118.4079971)}
```

## 2. Méthodes

```
In [10]: #Afficher les clés
print(dico1.keys())

#Afficher les valeurs
print(dico1.values())

#Afficher les éléments
print(dico1.items())

dict_keys(['nom', 'ville', 'pays', 'code', 'gps'])
dict_values(['Paris-Orly Airport', 'Paris', 'France', 'ORY', (48.7233333, 2.3794444)])
dict_items([('nom', 'Paris-Orly Airport'), ('ville', 'Paris'), ('pays', 'France'), ('code', 'ORY'), ('gps', (48.7233333, 2.3794444))])
```

### 3. Parcours

On peut traiter par un parcours les éléments contenus dans un dictionnaire, mais attention :

- Au cours de l'itération, **ce sont les clés** qui servent d'index
- L'ordre dans lequel le parcours s'effectue est imprévisible

Exemple :

```
In [11]: for element in dico1:  
         print(element)
```

```
nom  
ville  
pays  
code  
gps
```

#### Exercice 2 :

Modifier le programme ci-dessus pour obtenir l'affichage ci-dessous :

```
In [12]: for element in dico1:  
         print(element, ' : ', dico1[element])
```

```
nom  : Paris-Orly Airport  
ville : Paris  
pays : France  
code : ORY  
gps  : (48.7233333, 2.3794444)
```

### 4. A Vous !

#### Exercice 3



Sur le site <https://openflights.org/data.html> (<https://openflights.org/data.html>), on trouve des bases de données mondiales aéronautiques. Le fichier `airports.txt` présent dans le dossier de cette feuille contient des informations sur les aéroports.

Chaque ligne de ce fichier est formatée comme l'exemple ci-dessous :

```
1989,"Bora Bora Airport","Bora Bora","French Polynesia","BOB","NTTB",-16.444400787353516,-151.75100708007812,10,-  
10,"U","Pacific/Tahiti","airport","OurAirports"
```

On souhaite extraire les informations suivantes pour chaque aéroport:

- Sa référence unique, un entier.
- Le nom de l'aéroport, une chaîne de caractères.
- La ville principale qu'il dessert, une chaîne de caractères.
- Le pays de cette ville, une chaîne de caractères.
- Le code IATA de l'aéroport composé de 3 lettres en majuscules.
- Ses coordonnées gps (latitude puis longitude), un tuple de 2 flottants.
- Son altitude en pieds, par rapport au niveau de la mer, un entier.

## 1. Compléter les champs ci-dessous pour l'aéroport cité en exemple:

- ref :
  - nom :
  - ville :
  - pays :
  - code :
  - gps :
  - alt :
- 
- ref : 1989
  - nom : Bora Bora Airport
  - ville : Bora Bora
  - pays : French Polynesia
  - code : BOB
  - gps : (-16.444400787353516,-151.75100708007812)
  - alt : 10

## 2. La fonction `data_extract` doit parcourir le fichier et extraire les données demandées qu'elle renvoie sous forme d'une liste de dictionnaires.

- Chaque élément de la liste est donc un dictionnaire qui correspond à un aéroport.
- Les clés sont les noms des champs que l'on souhaite extraire et les valeurs sont celles associées à chaque aéroport.

Dans le code ci-dessous, remplacer tous les champs "A compléter" pour que la fonction fasse ce que l'on attend d'elle.

```
In [43]: from random import choice

#2.
def data_extract(chemin):
    fichier = open(chemin, "r", encoding='utf-8')
    res = [] # Le résultat sera une liste de dictionnaires
    for ligne in fichier:
        datas = ligne.strip().split(",") # chaque ligne du fichier est une liste
        res.append(
            { "ref": int(datas[0]),
              "nom": datas[1][1:-1],
              "ville": "A compléter",
              "pays": datas[3][1:-1],
              "A compléter": datas[4][1:-1],
              "gps" : "A compléter",
              "alt" : "A compléter"
            })

    fichier.close()
    return res

#La Liste de tous les aéroports, chaque aéroport est un dictionnaire
airports=data_extract('airports.txt')

#nombre d'aéroports référencés
print("A compléter")

#un aéroport au hasard
print(choice(airports))

A compléter
{'ref': 3972, 'nom': 'Kopitnari Airport', 'ville': 'A compléter', 'pays': 'Georgia', 'A compléter': 'KUT', 'gps':
'A compléter', 'alt': 'A compléter'}
```

```
In [60]: #2. Réponse
def data_extract(chemin):
    fichier = open(chemin, "r", encoding='utf-8')
    res = [] # Pour contenir le résultat qui sera une liste de dictionnaires
    for ligne in fichier:
        datas = ligne.strip().split(",")
        res.append(
            { "ref": int(datas[0]),
              "nom": datas[1][1:-1],
              "ville": datas[2][1:-1],
              "pays": datas[3][1:-1],
              "code": datas[4][1:-1],
              "gps": (float(datas[6]), float(datas[7])),
              "alt": int(datas[8])
            })

    fichier.close()
    return res

#La liste de tous les aéroports, chaque aéroport est un dictionnaire
airports=data_extract('airports.txt')

#nombre d'aéroports référencés
print(len(airports))

#un aéroport au hasard
print(choice(airports))

7698
{'ref': 6916, 'nom': 'Pangborn Memorial Airport', 'ville': 'Wenatchee', 'pays': 'United States', 'code': 'EAT', 'g
ps': (47.3988990784, -120.207000732), 'alt': 1249}
```

### 3. On souhaite récupérer la liste des villes françaises desservies par un aéroport. Compléter le code ci-dessous

```
In [19]: #3. Réponse
res=[airport['ville'] for ... in airports if ... == ... ]
print(res)

['Calais', 'Peronne', 'Nangis', 'Bagnole-de-l'orne', 'Albert', 'Le Tourquet', 'Valenciennes', 'Amiens', 'Agen', 'C
azaux', 'Bordeaux', 'Bergerac', 'Toulouse', 'Cognac', 'Poitiers', 'Montlucon-gueret', 'Limoges', 'Mont-de-marsan',
'Niort', 'Toulouse', 'Pau', 'La Rochelle', 'Tarbes', 'Angoulême', 'Brive', 'Perigueux', 'Biarritz-bayonne', 'Cahor
s', 'St.-girons', 'Arcachon', 'Albi', 'Castres', 'Toulouse', 'Millau', 'Graulhet', 'Rodez', 'Ussel', 'Villeneuve-s
ur-lot', 'Royan', 'Mimizan', 'Aire-sur-l'adour', 'Montauban', 'Auch', 'Libourne', 'Pamiers', 'Marmande', 'Rochefor
t', 'Pontivy', 'Aubigny-sur-nere', 'Guiscriff-scaer', 'Ancenis', 'Brienne-le Chateau', 'Colmar', 'Beaune', 'Dole',
'Joigny', 'Verdun', 'Aubenas-vals-lanas', 'Le Puy', 'St.-flour', 'Bourg', 'Vilefrance', 'Moulins', 'St.-afrique-be
lmont', 'Cassagnes-beghones', 'Metz', 'Bastia', 'Calvi', 'Figari', 'Ajaccio', 'Propriano', 'Solenzara', 'Corte', '
Auxerre', 'Chambery', 'Clermont-Ferrand', 'Bourges', 'Chambery', 'Chalon', 'Annemasse', 'Lyon', 'Macon', 'St.-yan
', 'Roanne', 'Annecy', 'Grenoble', 'Montlucon', 'Valence', 'Vichy', 'Aurillac', 'Chateauroux', 'Lyon', 'Aix-les-mi
lles', 'Le Luc', 'Cannes', 'St-Etienne', 'Istres', 'Carcassonne', 'Marseille', 'Nice', 'Orange', 'Perpignan', 'Le
Castellet', 'Ales', 'Montpellier', 'Beziers', 'Avignon', 'Salon', 'Lezignan-corbieres', 'Mende', 'Carpentras', 'Av
ord', 'Beauvais', 'Chateaudun', 'Saumur', 'Evreux', 'Le Havre', 'Abbeville', 'Orleans', 'Chalons', 'Rouen', 'Tours
', 'Cholet', 'Laval', 'Orleans', 'Paris', 'Creil', 'Paris', 'Coulommiers', 'Melun', 'Toussous-le-noble', 'Paris',
'Pontoise', 'Villacoublay', 'Reims', 'Troyes', 'Luneville', 'Etain', 'Autun', 'Nevers', 'Cambrai', 'Maubeuge', 'Be
sancon-la-veze', 'Phalsbourg', 'Lille', 'Merville', 'Charleville', 'Vesoul-frotey', 'Brest', 'Cherbourg', 'Dinard
', 'La Baule', 'Granville', 'Deauville', 'Lorient', 'La Roche-sur-yon', 'Landivisiau', 'Caen', 'Lanvedoc', 'Le Man
s', 'Rennes', 'Lannion', 'Quimper', 'Nantes', 'St.-brieuc Armor', 'Morlaix', 'Vannes', 'St.-nazaire', 'Mulhouse',
'Colmar', 'Dijon', 'Metz', 'Epinal', 'Haguenau', 'St.-dizier', 'Montbeliard', 'Nancy', 'Nancy', 'Pontarlier', 'Rei
ms', 'Strasbourg', 'Luxeuil', 'Cuers', 'Hyeres', 'Nimes', 'Amberieu', 'Damblain', 'Enghien-moisselles', 'Île d'Ye
u', 'Angers/Marcé', 'La Môle', 'Gustavia', 'Courcheval', 'Andernos-Les-Bains', 'La Rochelle', 'Ouessant', 'Frejus
', 'Saint-Cyr', 'Strasbourg Neudorf', 'Verdun', 'Ajaccio', 'Pont Sur Yonne', 'St Florentin Cheu', 'Saulieu', 'Meau
x', 'Chateau-Thierry', 'Belfort', 'La Ferte Alais', 'Marigot', 'Marcq En Baroeul', 'Tremblay-en-France', 'La Grand
Combe', 'Malbouhans', 'Blois/Le Breuil', 'Lognes', 'Saint Pierre d'Oleron', 'Semur en Auxois']
```

```
In [ ]: #3. Réponse
res=[airport['ville'] for airport in airports if airport['pays']=='France']
print(res)
```

### 4. Ecrire la fonction infos(airports,ville) .

Elle prend en paramètre la liste des dictionnaires de données extraites et une chaîne de caractères( le nom d'une ville). Elle renvoie les informations du ou des aéroports de cette ville dans une liste de dictionnaires:

In [20]:

```
#4.

print(infos(airports,'Nice'))

[{'ref': 1354, 'nom': "Nice-Côte d'Azur Airport", 'ville': 'Nice', 'pays': 'France', 'code': 'NCE', 'gps': (43.6584014893, 7.215869903560001), 'alt': 12}]
```

In [ ]:

```
#4.
def infos(airports,ville):
    res=[]
    for airport in airports:
        if airport['ville']==ville:
            res.append(airport)
    return res

print(infos(airports,'Nice'))
```

5. Compléter les listes du code ci-dessous pour représenter les points de coordonnées gps de chacun des aéroports de la base de données(liste X des longitudes et liste Y des latitudes)

In [51]:

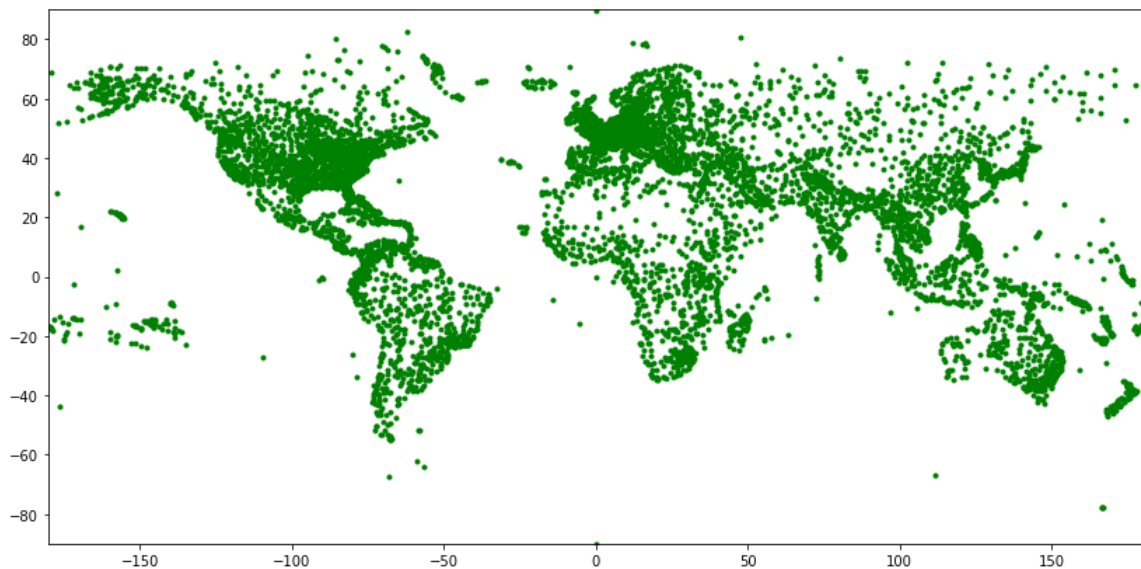
```
#5.Réponse
from matplotlib import pyplot #une bibliothèque pour afficher des graphiques

X=[...] #Liste des Longitudes
Y=[...] #Liste des Latitudes

#paramétrage du graphique
pyplot.figure(figsize = (14,7))
pyplot.xlim(-180, 180)
pyplot.ylim(-90, 90)

#affichage des points
pyplot.plot(X,Y,'.', color='g') # '.' : forme du point , 'g': 'green'
```

Out[51]: [ <matplotlib.lines.Line2D at 0x1bb22f06e20>]



In [ ]:

```
#5.Réponse
from matplotlib import pyplot #une bibliothèque pour afficher des graphiques

X=[airport['gps'][1] for airport in airports ] #Liste des Longitudes
Y=[airport['gps'][0] for airport in airports ] #Liste des Latitudes

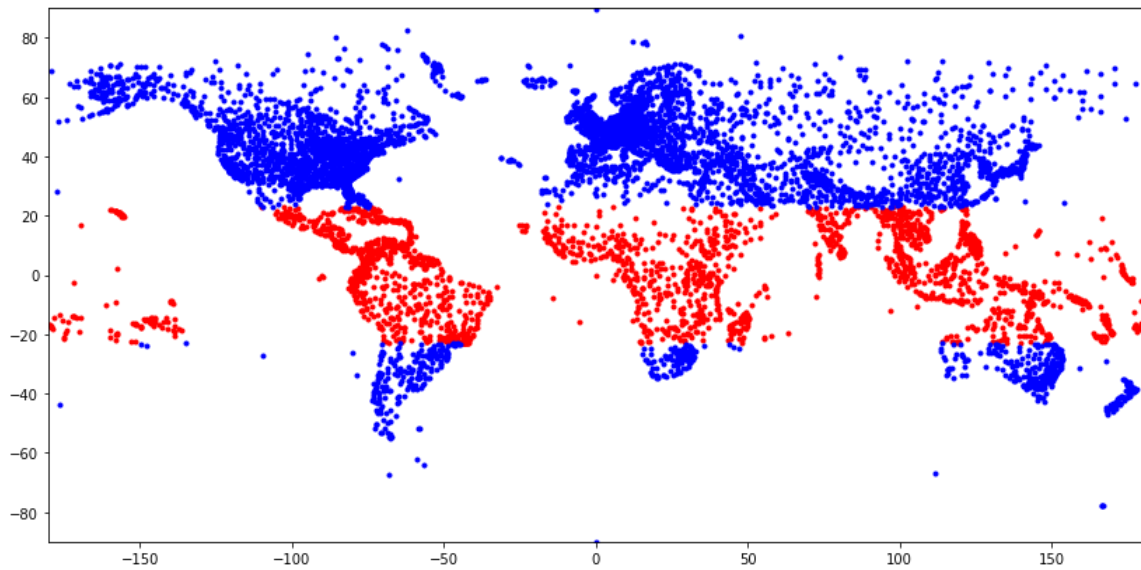
#paramétrage du graphique
pyplot.figure(figsize = (14,7))
pyplot.xlim(-180, 180)
pyplot.ylim(-90, 90)

#affichage des points
pyplot.plot(X,Y,'.', color='g') # '.' : forme du point , 'g': 'green'
```

6. Recopier et modifier le code précédent pour faire apparaître en rouge les aéroports situés en zone tropicale (c'est à dire dont la latitude est comprise entre  $-23$  et  $+23$ ) et en bleu les autres

In [56]: #5.

Out[56]: [<matplotlib.lines.Line2D at 0x1bb22eca160>]



```
#5. Réponse
from matplotlib import pyplot

X=[airport['gps'][1] for airport in airports if (airport['gps'][0]<23 and airport['gps'][0]> -23) ]
Y=[airport['gps'][0] for airport in airports if (airport['gps'][0]<23 and airport['gps'][0]> -23) ]

X2=[airport['gps'][1] for airport in airports if (airport['gps'][0]>23 or airport['gps'][0]< -23)]
Y2=[airport['gps'][0] for airport in airports if (airport['gps'][0]>23 or airport['gps'][0]< -23)]

pyplot.figure(figsize = (14,7))
pyplot.xlim(-180, 180)
pyplot.ylim(-90, 90)

pyplot.plot(X,Y,'.', color='r')
pyplot.plot(X2,Y2,'.', color='b')
```

7. Compléter ce code pour que la couleur de chaque aéroport dépende de son altitude.

Aide : Matplotlib permet d'utiliser de nombreux dégradés de couleurs.

- On pourra consulter <http://www.python-simple.com/python-matplotlib/couleurs-matplotlib.php> (<http://www.python-simple.com/python-matplotlib/couleurs-matplotlib.php>)
- Extrait de la rubrique **Gammes de couleurs (color maps)**
  - Utilisation d'une "color map" pour représenter des couleurs par un nombre entre 0 et 1 :

```
myMap = pyplot.get_cmap('winter')
color = myMap(0.3)
```

```
In [57]: #5. Réponse
from matplotlib import pyplot

#Listes des coordonnées GPS
X= ...

Y= ...

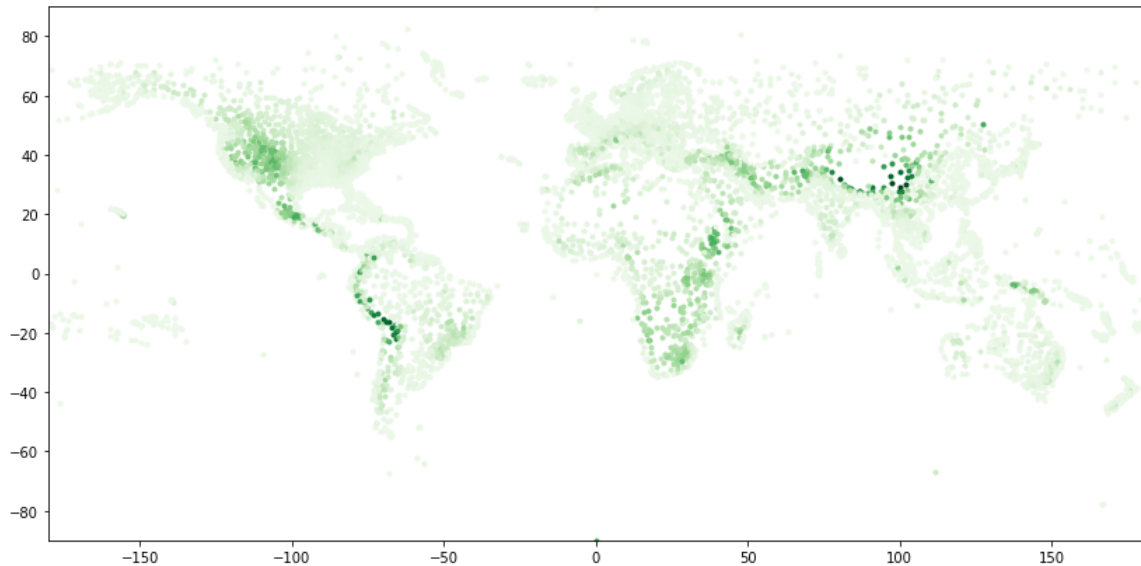
#liste des altitudes
ALT=[ .          ...          ]

#altitude minimales et maximales
alt_min, alt_max= ... , ...

#graphique
myMap = pyplot.get_cmap('Greens') #choix du dégradé de couleur, ici "Greens"
pyplot.figure(figsize = (14,7))
pyplot.xlim(-180, 180)
pyplot.ylim(-90, 90)

#nombre d'aéroports
n=...

#afficher les points
for i in range(n):
    ...
```



```
In [ ]: #5. Réponse
from matplotlib import pyplot

#Listes des coordonnées GPS
X=[airport['gps'][1] for airport in airports ]
Y=[airport['gps'][0] for airport in airports ]

#liste des altitudes
ALT=[airport['alt'] for airport in airports]

#altitude minimales et maximales
alt_min, alt_max=min(ALT),max(ALT)

#graphique
myMap = pyplot.get_cmap('Greens') #choix du dégradé de couleur
pyplot.figure(figsize = (14,7))
pyplot.xlim(-180, 180)
pyplot.ylim(-90, 90)

#nombre d'aéroports
n=len(ALT)

#afficher les points
for i in range(n):
    pyplot.plot(X[i],Y[i],'.', color=myMap((ALT[i]-alt_min)/(alt_max-alt_min)))
```