

TRAVAUX PRATIQUES
La programmation en Python
T.P. n°8 : Les dictionnaires

BTS CIEL
2 <sup>ème</sup> année
Page 1 sur 3

## Les dictionnaires

```

1  mesLangages = {}
2  print(mesLangages)
3  print(type(mesLangages))
4  mesLangages2 = dict()
5  print(mesLangages2)
6  print(type(mesLangages2))
7  moi = {"prenom": "Ilyas", "age": 45, "profession": "Professeur"}
8  print(moi["prenom"])
9  print(moi["age"])
10 print(moi["profession"])
11 moi["langage"] = "Python"
12 print(moi)
13 print(moi["langage"])
14 for key in moi:
15     print(key, ":", moi[key])
16 print(moi.keys())
17 print(moi.values())
18 langageHautNiveau = {
19     "langage": "Python",
20     "web": "Django",
21     "Gui": "Pyside",
22 }
23 print(langageHautNiveau.keys())
24 print(langageHautNiveau.values())
25 print(langageHautNiveau.items())
26 for cle, valeur in langageHautNiveau.items():
27     print(cle, "=>", valeur)
28
29 if "Python" in langageHautNiveau.values():
30     print("Python est dans le dictionnaire")
31 else:
32     print("Python n'est pas dans le dictionnaire")
33
34 if "langage" in langageHautNiveau:
35     print("La clé langage est dans le dictionnaire")
36 else:
37     print("La clé langage n'est pas dans le dictionnaire")
38
39 langageBasNiveau = {"langage": "C++", "web": "CppCMS", "Gui": "Qt",}
40 langages = [langageHautNiveau, langageBasNiveau]
41 print(langages)
42 for element in langages:
43     print(element["langage"])
44     print(element["web"])
45     print(element["Gui"])
46

```

TRAVAUX PRATIQUES
La programmation en Python
T.P. n°8 : Les dictionnaires

BTS CIEL
2 <sup>ème</sup> année
Page 2 sur 3

## Exercice 1: Gestion des données géographiques des communes

Vous êtes chargé(e) de développer un programme qui permet de stocker et de manipuler des données géographiques pour différentes communes. Chaque commune a un nom et un code postal associé.

1. **Créez** un dictionnaire vide appelé `donnees_geographiques` pour stocker les données géographiques des communes.
2. **Ajoutez** les données géographiques pour trois communes : Paris, Marseille et Lyon. Pour chaque commune, enregistrez les codes postaux associés.
3. **Affichez** les données géographiques pour chaque commune.
4. **Permettez** à l'utilisateur de rechercher les données géographiques pour une commune donnée en saisissant le nom de la commune. **Affichez** un message approprié si la commune n'est pas trouvée.
5. **Ajoutez** une fonction pour mettre à jour les données géographiques pour une commune donnée.

## Contraintes :

**Utilisez** des fonctions pour organiser votre code.

**Assurez-vous** que le programme affiche un message approprié lorsque la commune recherchée n'est pas dans le dictionnaire.

L'utilisateur devrait pouvoir choisir s'il souhaite **mettre à jour les** données ou **quitter** le programme après avoir consulté les données.

TRAVAUX PRATIQUES
La programmation en Python
T.P. n°8 : Les dictionnaires

BTS CIEL
2 <sup>ème</sup> année
Page 3 sur 3

## Exercice 2: Gestion des données météorologiques

Vous avez été chargé(e) de développer un programme qui permet de stocker et de manipuler des données météorologiques pour différentes villes. Chaque ville a un nom, une température actuelle et un niveau d'humidité.

1. **Créez** un dictionnaire vide appelé `donnees_meteo` pour stocker les données météorologiques et **ajoutez** les données météorologiques pour trois villes : Paris, New York et Tokyo. Pour chaque ville, enregistrez la température actuelle et le niveau d'humidité.

2.

```
donnees_meteo = {
    "Paris": {"temperature": 20.0, "humidite": 60},
    "New York": {"temperature": 25.0, "humidite": 70},
    "Tokyo": {"temperature": 28.0, "humidite": 80}
}
```

3. **Affichez** les données météorologiques pour chaque ville.
4. **Permettez** à l'utilisateur de rechercher les données météorologiques pour une ville donnée en saisissant le nom de la ville. **Affichez** un message approprié si la ville n'est pas trouvée.
5. **Ajoutez** une fonction pour mettre à jour les données météorologiques pour une ville donnée.

### Contraintes :

**Utilisez** des fonctions pour organiser votre code.

**Assurez-vous** que le programme affiche un message approprié lorsque la ville recherchée n'est pas dans le dictionnaire.

L'utilisateur devrait pouvoir choisir s'il souhaite **mettre à jour** les données ou **quitter** le programme après avoir consulté les données.