

Corrigé – Les dictionnaires

Exercice 1

voir feuille de cours

Exercice 2 – (Les données EXIF d'une image)

1. `>>> exif = {"largeur": 4592, "hauteur": 2584}`
2. `>>> exif["hauteur"]`

Exercice 3 – (Un groupe de rock)

1.

```
rockband = {"chanteur": "Julien Casablancas", "guitariste rythmique":  
    ↳ "Albert Hammond Jr", "guitariste principal": "Nick Valensi", "bassiste":  
    ↳ "Nikolai Fraiture", "batteur": "Fabrizio Moretti"}
```
2. `rockband.values()`
3. Écrire une fonction `est_membre(rockband, nom)` renvoyant `True` si `nom` est présent dans le groupe, `False` sinon. On fera un parcours de dictionnaire.

```
def est_membre(rockband, nom):  
    for nom_membre in rockband.values():  
        if nom_membre == nom:  
            return True  
    return False
```

4. Tester si "Nick Valensi" et "Bjork" appartiennent au groupe ou non.

```
>>> est_membre(rockband, "Nick Valensi")  
True  
>>> est_membre(rockband, "Bjork")  
False
```

Exercice 4 – (Manipulation de dictionnaires)

1. `"charlotte"`
2. `21`
3. `print(panier[2]["nombre"])`
4.

```
repertoire = [{"nom": "Dupont", "tel": "5234"},  
    {"nom": "Tournesol", "tel": "5248"}, {"nom": "Dupond", "tel": "3452"}]  
  
for i in range(len(repertoire)):  
    if repertoire[i]["nom"] == "Dupond":  
        print(repertoire[i]["tel"])
```

Exercice 5 – (Recherche de maximum)

```
def recherche_max(dico):
    maxi = 0
    cle_max = None
    for cle in dico.keys():
        if dico[cle] > maxi:
            maxi = dico[cle]
            cle_max = cle
    return cle_max
```

Exercice 6 – (Comptage d'occurrences)

```
1. fonction compte_occurrence(chaine):
    | dico = {}
    | pour chaque caractere c de chaine:
    || si c n'est pas une clé de dico:
    ||| créer la clé dans dico et l'initialiser à 0
    || ajouter 1 à la valeur de la clé c
    | renvoyer dico

2.1 # FONCTIONS
2 def compte_occurrences(chaine):
3     dico_occur = {}
4     for carac in chaine:
5         if carac not in dico_occur:
6             dico_occur[carac] = 0
7             dico_occur[carac] += 1
8     return dico_occur
9
10 # SCRIPT
11 print(compte_occurrences("bonjour"))
12 # affiche : {'b': 1, 'o': 2, 'n': 1, 'j': 1, 'u': 1, 'r': 1}
```

Exercice 7 – (TP : résultats d'une élection)

- La gagnante est Ada Lovelace.
 - 2 votes blancs
 - 1 vote nul (Zuckerberg)
- ```
candidats = ["Alan Turing", "Ada Lovelace", "George Boole"]
occurrences = {"Nul": 0, "Blanc": 0}
for candidat in candidats:
 occurrences[candidat] = 0
print(occurrences)
```

ou

```
occurrences = {cle:0 for cle in candidats}
occurrences["Blanc"] = 0
occurrences["Nul"] = 0
```

3.

```
def est_present(tab, v):
 for elmt in tab:
 if elmt == v:
 return True
 return False
```

Et si on veut tester son programme :

```
tab = [3, 5, 7, 9, 1]
print(est_present(tab, 5)) # doit afficher True
assert est_present(tab, 5)
print(est_present(tab, 22)) # doit afficher False
assert not est_present(tab, 22)
```

4.

```
def depouillement(candidats, votes):
 occurrences = {cle:0 for cle in candidats}
 occurrences["Blanc"] = 0
 occurrences["Nul"] = 0
 for v in votes:
 if est_present(candidats, v):
 occurrences[v] += 1
 elif v == "":
 occurrences["Blanc"] += 1
 else:
 occurrences["Nul"] += 1
 return occurrences
```

5.

```
candidats = ["Alan Turing", "Ada Lovelace", "George Boole"]
votes = ["Alan Turing", "Ada Lovelace", "Ada Lovelace", "", "George Boole", "Mark
→ Zuckerberg", ""]
assert depouillement(candidats, votes) == {'Alan Turing': 1, 'Ada Lovelace': 2,
→ 'George Boole': 1, 'Blanc': 2, 'Nul': 1}
print(depouillement(candidats, votes)) # doit afficher {'Alan Turing': 1, 'Ada
→ Lovelace': 2, 'George Boole': 1, 'Blanc': 2, 'Nul': 1}
```

6.

```
def trouve_maximum(t):
 maxi = 0
 for val in t:
 if val > maxi:
 maxi = val
 return maxi
```

On peut tester cette fonction de cette manière :

```
tab = [3, 5, 7, 9, 1]
assert trouve_maximum(tab) == 9
```

7. 

```
def vainqueur(d):
 nb_vote_max = 0
 élu = ""
 for candidat, nb_voies in d.items():
 if nb_voies > nb_vote_max:
 nb_vote_max = nb_voies
 élu = candidat
 return élu
```
8. 

```
res_election = depouillement(candidats, votes)
resultat = vainqueur(res_election)
print(resultat) # doit afficher "Ada Lovelace"
assert resultat == "Ada Lovelace"
```