TP n°2 : Dictionnaires

Instruction générale : hormis pour les exercices corrigés collectivement, vous ferez valider votre travail par l'enseignant.

1 Dictionnaire

Un **dictionnaire** en Python est un objet contenant des paires de **clés-valeurs**.

— Les **clés** sont des éléments non modifiables et doivent être uniques. Elles ont pour types des objets immuables comme les entiers, les chaînes de caractères, les tuples.

— Les **valeurs** associées au clés sont elles modifiables et peuvent être de n'importe quel type.

Remarques :

— Un dictionnaire est non ordonné.

— Un dictionnaire est modifiable, on peut modifier ses valeurs, ajouter ou supprimer des éléments après sa création.

Exemples :

1. On crée un dictionnaire vide soit en utilisant la commande dict(), soit en utilisant des accolades {}.

```
dico1=dict() # on crée un dictionnaire vide.
dico2={} # on crée un autre dictionnaire vide/
```

2. Pour créer un dictionnaire directement avec des paires clés-valeurs, on utilise exclusivement les accolades. Une valeur est associée à une clé selon la syntaxe **clé : valeur** et les différente paires sont séparées par des virgules.

```
pokemon1={'nom':'bulbizarre','type':'plante'}
pokemon2={'nom':'herbizarre','type':'plante'}
pokedex={1:pokemon1,2:pokemon2} # dictionnaire de dictionnaires !
```

On remarquera ici que les clés sont bien uniques au sein de leurs dictionnaires même si des dictionnaires différents les partagent.

Exercice 1. L'objectif de cet exercice et des suivants est de compléter notre podédex.

1. Créer un dictionnaire vide pokemon3.
2. Créer un dictionnaire pokemon4 pour Salamèche avec la clé 'nom' (on s'occupera du type plus tard).

2 Opérations sur les dictionnaires

2.1 Ajout d'un élément à un dictionnaire

On peut **ajouter** un couple clé-valeur à un dictionnaire à condition que la clé soit bien unique. Pour cela, il suffit de faire **dictionnaire[clé]=valeur**.

Exemple : On ajoute au dictionnaire pokemon3 le couple clé-valeur 'nom'-'florizarre' en faisant :

```
pokemon3['nom']='florizarre'
```

Exercice 2. Ajouter au dictionnaire pokemon3 le type de Florizarre puis le dictionnaire pokemon3 au podédex.

2.2 Modification d'un élément d'un dictionnaire

On peut **modifier** la valeur associée à une clé mais pas celle-ci. Pour cela, on utilise à nouveau la commande **dictionnaire[clé]=valeur**.

Exemple : On a crée un dictionnaire pour Carapuce mais celui-ci n'a pas le bon type, on le modifie donc pour corriger cela.

```
pokemon7={'nom':'carapuce','type':'foudre'}
pokemon7['type']='eau'
```

Exercice 3. Modifier le dictionnaire suivant afin de corriger l'erreur sur le type de Reptincel.

```
pokemon5={'nom':'reptincel','type':'psy','exp.':9999}
```

2.3 Suppression d'un élément d'un dictionnaire

On peut **supprimer** un couple clé-valeur d'un dictionnaire en indiquant la clé du couple à supprimer grâce à la commande **dico1.pop(clé)**.

Exemple : On a récupéré un dictionnaire pour Dracaufeu mais celui-ci comporte l'élément 'plat favori' dont on ne veut pas, on le supprime donc.

```
pokemon6={'nom':'dracaufeu','type':'feu','plat favori':'risotto'}
pokemon6.pop('plat favori')
```

Exercice 4. Supprimer l'élément 'exp.' du dictionnaire de Reptincel.

2.4 Fusion de deux dictionnaires

On peut **fusionner** deux dictionnaires grâce à la commande **dico1.update(dico2)**.

Exemple : On crée le dictionnaire typePlante et on le fusionne avec pokemon3.

```
typePlante={'type':'plante'}
pokemon3.update(typePlante)
```

Exercice 5. Créer un dictionnaire typeFeu sur le modèle de typePlante puis le fusionner avec le dictionnaire de Salamèche.

2.5 Parcours d'un dictionnaire

On peut **parcourir** un dictionnaire selon ses clés, ses valeurs ou ses couples clé-valeur grâce aux commandes suivantes.

Parcours selon	Clés	Valeurs	Couples clé-valeur
Commande	dico.keys()	dico.values()	dico.items()

Exemples :

— L'algorithme suivant permet de parcourir le dictionnaire pokemon1 selon ses clés.

```
for cle in pokemon1.keys():
    print(cle)
```

Il renvoie donc

```
nom
type
```

— L'algorithme suivant permet de parcourir le dictionnaire pokemon1 selon ses valeurs.

```
for valeur in pokemon1.values():
    print(valeur)
```

Il renvoie donc

```
bulbizarre
plante
```

— L'algorithme suivant permet de parcourir le dictionnaire pokemon1 selon ses couples clé-valeur.

```
for cle,valeur in pokemon1.items():
    print(cle,valeur)
```

Il renvoie donc

```
nom bulbizarre
type plante
```

Exercice 6.

1. Que va renvoyer le dictionnaire pokedex selon qu'il soit parcouru par ses :

   (a) clés ? (b) valeurs ? (c) couples ?

2. Programmer en Python le parcours du pokédex.

Exercice 7. Écrire un programme permettant d'afficher le nom de chaque pokémon contenu dans notre pokédex et son type dans une phrase au format :

« nomPokémon est du type typePokémon. »

Exercice 8. [Longueur des mots d'une phrase] Écrire un programme prenant en entrée un texte et comptant le nombre de mots de chaque longueur possible puis stockant le résultat dans un dictionnaire au format

```
occurencesLongueursMots={longueur:nombreMots}
```

Par exemple, avec la phrase « je pense donc je suis », on obtiendra

```
occurencesLongueursMots={1:0,2:2,3:0,4:2,5:1,6:0,...}
```

Afin de simplifier, on ne tiendra pas compte de la ponctuation et on considérera une seule catégorie pour les mots de plus de vingt lettres. Enfin, on affichera le résultat sous la forme d'un histogramme.

Indication : il est possible d'initialiser le dictionnaire `occurencesLongueursMots` en compréhension grâce à la commande

```
occurencesLongueursMots={i : 0 for i in range(1,21)}
```

Exercice 9. [Fréquence des lettres dans une phrase] Écrire un programme prenant en entrée un texte et déterminant la fréquence d'apparition de chaque lettre puis stockant le résultat dans un dictionnaire au format

```
frequenceLettres={lettre:fréquence}
```

On affichera le résultat sous forme d'un histogramme.

Indication : il est possible d'initialiser le dictionnaire `occurencesLettres` en compréhension grâce à la commande

```
occurencesLettres={chr(97+i) : 0 for i in range(26)}
```