

	Evaluation	
	Evaluation formative Python	
	Lycée Jules Haag – salle 265	

Consignes

Faire les exercices suivants, dans l'ordre. Un exercice par cellule dans Jupyter Notebook que vous nommerez « **Evaluation_python2** » et que vous enregistrerez dans votre session locale.

Internet autorisé

Exercices python piscine autorisés

Exercice 1 : Température

Le but de cet exercice est de créer une fonction **indicateur_temperature** qui pour une température donnée renvoie un indicateur de température selon le tableau suivant :

Température	Indicateur
$] -\infty ; -40]$	-1
$] -40 ; -20]$	0
$] -20 ; -5]$	1
$] -5 ; 5]$	2
$] 5 ; 12]$	3
$] 12 ; 20]$	4
$] 20 ; 50]$	5
$] 50 ; +\infty[$	-1

→ Attention : l'indicateur des plages surlignées en rouges sont des erreurs. Elles doivent être traitées en tant que tel.

→ Attention : on peut donner à la fonction un nombre décimal. Porter une attention particulière à la définition des intervalles.

Entrée : Une température entière ou décimale entre $-\infty$ et $+\infty$.

Sortie : L'indicateur du tableau correspondant.

	Evaluation	
	Evaluation formative Python	
	Lycée Jules Haag – salle 265	

Exercice 2 : compte_lettres

Le but de cet exercice est de créer une fonction **compte_lettres** qui compte le nombre de lettres particulières dans un mot.

Par exemple :

- **compte_lettres**("bonjour", 'o') → 2
- **compte_lettres**("bonjour", 'j') → 1

Entrée : Un **mot** et la lettre que l'on souhaite compter dans le mot.

Sortie : Le nombre de lettres comptées dans le mot fourni.

Exercice 3 : somme_paire

Le but de cet exercice est de créer une fonction **somme_paire** qui renvoie True si la somme des nombres de la **liste** donnée en entrée est paire et False sinon.

Entrée : Une **liste** de nombres entiers.

Sortie : True si la somme des nombres de la **liste** est binaire, False sinon.

Exemple :

- **somme_paire**([2, 3, 10, 5, 45]) → False
- **somme_paire**([7, 50, 13, 8, 56]) → True

	Evaluation	
	Evaluation formative Python	
	Lycée Jules Haag – salle 265	

Exercice 4 : gestion de dictionnaire.

Écrire une fonction qui analyse, complète et génère un dictionnaire.

La fonction prend en entrée un dictionnaire regroupant des informations sur des individus (âge, taille, poids, couleur des yeux, etc.).

La fonction retourne un dictionnaire avec les mêmes informations mais complétées avec :

- l'IMC de chaque personne
- Le remplacement de la couleur des yeux par un caractère (bleu → B ; rouge → R ; vert → V ; etc.).

En plus de cela, la fonction renvoie en fin de dictionnaire les statistiques de la liste de personnes :

- La moyenne des âges
- La moyenne des tailles
- La moyenne des poids
- Liste des couleurs des yeux
- Liste des métiers
- Liste des IMC

Par exemple si en entrée on fournit :

```
personnes = {
  "personne1":{
    "age":18,
    "taille":180,
    "poids":80,
    "couleur_yeux":"bleus",
    "metier":"enseignant"
  },
  "personne2":{
    "age":87,
    "taille":190,
    "poids":59,
    "couleur_yeux":"rouges",
    "metier":"vampire"
  }
}
```

	Evaluation	
	Evaluation formative Python	
	Lycée Jules Haag – salle 265	

En retour, la fonction doit renvoyer :

```
personnes = {
    "personne1":{
        "age":18,
        "taille":180,
        "poids":80,
        "couleur_yeux":"B",
        "metier":"enseignant",
        "imc":24.7
    },
    "personne2":{
        "age":87,
        "taille":190,
        "poids":59,
        "couleur_yeux":"R",
        "metier":"vampire",
        "imc":18.5
    },
    "statistiques":{
        "moyenne_ages":52.5,
        "moyenne_tailles":185,
        "moyenne_poids":69.5,
        "liste_couleurs":["B", 'R'],
        "liste_metiers":["enseignant", "vampire"],
        "liste_IMC":[24.7, 18.5]
    }
}
```