
Outils numériques
TP1 : Les bases de Python, structures de contrôle et conteneurs

Il est recommandé de créer un dossier TP1 dans lequel vous enregistrerez un fichier Python par exercice.

Exercice 1 : Prise en main de Pyzo et Python.

1. Ouvrir Pyzo sur les machines de la fac, puis familiarisez-vous avec son environnement (éditeur de texte, interpréteur, etc.).
2. Entrez la commande `print("Salut !")` dans l'interpréteur Python (c'est le *shell* Python directement intégré dans Pyzo).
3. Ouvrez un fichier `exo1.py`, qui contiendra la même commande. Enregistrez le fichier puis exécutez-le à l'aide de Pyzo.
4. Essayez de prédire le résultat de chacune des instructions suivantes, puis vérifiez le dans l'interpréteur Python de Pyzo :
 - `str(4) * int("3")`
 - `int("3") + float("3.2")`
 - `str(3) * float("3.2")`
 - `str(3/4) * 2`
5. Quel nombre est plus grand entre $42153 / 124$ et $41753 / 123$? Trouver la réponse à l'aide d'un booléen.

Exercice 2 : Structures de contrôle. Écrire un script qui exécute l'algorithme suivant :

Algorithme 1 : Mon premier algorithme !

Affecter une valeur à la variable x ;
Si elle est négative, lui soustraire 2;
Sinon, si elle est paire, la multiplier par 2 puis lui ajouter 4 ; si elle est impaire, la diviser par 5;
Afficher la valeur finale;

Exercice 3 : Structures de contrôle. Écrire un script qui exécute l'algorithme suivant. Expliquer la raison pour laquelle cet algorithme calcule la précision machine puis le tester.

Algorithme 2 : Mystère...

$\varepsilon \leftarrow 1$;
 $a \leftarrow 1$;
 $b \leftarrow 1$;
while $a \neq b$ **do**
 $\varepsilon \leftarrow \varepsilon/2$;
 $b \leftarrow 1 + \varepsilon$;

Exercice 4 : Structures de contrôle.

1. Écrire un script qui détermine si un nombre entier donné est pair ou impair.
2. Écrire un script qui détermine si un nombre entier donné est premier ou non.

Exercice 5 : Conteneurs.

1. Compter le nombre de voyelles dans la phrase « Python est un langage vraiment chouette, j'adore l'apprendre! »
2. Créer la liste de tous les entiers pairs de 0 à 586, puis extraire (par slicing) tous les éléments d'indice impair de cette liste.

Exercice 6 : Conteneurs.

1. Créer une liste nommée `nourriture` avec deux éléments `'riz'` et `'pâtes'`.
2. Ajouter la chaîne `'haricots'` à `nourriture` en utilisant `.append()`.
3. Ajouter les chaînes `'pain'` et `'pizza'` à `nourriture` en utilisant `.extend()`.
4. Afficher les deux premiers items de `nourriture` en utilisant le slicing.
5. Afficher le dernier élément de `nourriture`.
6. Créer une liste nommée `petit_dejeuner` à partir de la liste `'œufs, fruits, jus d'orange'` en utilisant la fonction `.split()`.
7. Vérifier que la liste `petit_dejeuner` possède bien trois items.
8. Créer, par compréhension, une nouvelle liste qui contient les longueurs de chaque chaîne de caractères de la liste `petit_dejeuner`.

Exercice 7 : Structures de contrôles et conteneurs.

1. Créer une liste contenant 10 entiers.
2. Afficher le minimum, le maximum et la somme de cette liste (sans passer par les fonctions Python associées).

Exercice 8. Voici un dictionnaire qui contient des noms (au format `str`) et des dates d'anniversaire (au format `tuple`).

```
dates = {
    "Alan": (23, 6, 2012),
    "Grace": (9, 12, 2006),
    "Linus": (28, 12, 1999),
    "Guido": (31, 1, 2004),
    "Ada": (10, 12, 2015),
    "Tim": (8, 6, 1995),
    "Dennis": (9, 9, 2000),
    "Hedy": (9, 11, 1980),
    "Steve": (24, 2, 1985)
}
```

1. Ajouter les deux entrées suivantes : Margaret, née le 17 août 1996 et John, né le 28 décembre 2015.
2. Écrire un script permettant de donner le nom de la personne la plus jeune du dictionnaire.