

Travaux Dirigés du Module Informatique

Exercices élémentaires

Les exercices suivants ont été conçus pour vous faire manipuler les primitives de base du langage Python et vous faire ainsi découvrir leurs possibilités. Les exercices les plus simples peuvent se réduire à une seule ligne de code. Bonne initiation...

Manipulation de nombres

Ex 1. Soient $a = 25$ et $b = 3$.

- Calculer $c = a/b$ par la division classique, $d = a/b$ par la division entière et le reste r .
- Afficher c en imposant 2 chiffres après la virgule.
- Calculer a^2 , \sqrt{a} .

Ex 2. Écrire un programme qui affiche pour chaque nombre pair inférieur ou égal à 20 :

n : 2 | Carré : 4 | Cube : 8

- en utilisant une boucle `for`,
- en utilisant une boucle `while`.

Ex 3. Écrire une fonction qui affiche l'aire d'un cercle.

Ex 4. Construire une fonction qui affiche les caractéristiques d'un cercle au format suivant :

```
Cercle :
- Rayon : 2
- Périmètre : 12,57
- Aire : 12,57
```

Ex 5. Soit $a = 2.451 \times 10^6$.

- Afficher a au format scientifique en imposant 2 chiffres après la virgule.
- Ex 6.** Soit $a = 2 + 3i$ et $b = 3 + 7i$. Calculer $c = a + b$.
- Afficher la partie réelle et la partie imaginaire de c .
- Afficher c en coordonnées polaires.

Manipulations de listes

Ex 7. Effectuer les opérations suivantes :

```
maliste = ['Lun', 'mar', 'Sam', 'Dim']
```

- Calculer la longueur de la liste `maliste`.
- Ajouter 'jeu' à `maliste`.
- Afficher le deuxième élément, l'avant-dernier élément.
- Construire une copie de cette liste. Inverser l'ordre de la copie et vérifier si vous n'avez pas changé l'ordre de la liste initiale.
- Trier cette liste donnée sous forme alphabétique croissante puis décroissante.

Ex 8. Effectuer les opérations suivantes :

```
maliste = [1, 2, 3, 4]
```

- Extraire le premier élément de la liste.
- Extraire le dernier élément de la liste.

Structure imbriquées, fonctions

Ex 9. Utilisez l'instruction `break` pour interrompre une boucle `for` d'affichage des entiers de 1 à 10 compris, lorsque la variable de boucle vaut 5.

Ex 10. Utilisez l'instruction `continue` pour modifier une boucle `for` d'affichage de tous entiers de 1 à 10 compris, sauf lorsque la variable de boucle vaut 5.

Ex 11. Écrire une fonction qui retourne $n!$:

- en utilisant une boucle `for`,
- en utilisant une boucle `while`,
- avec un algorithme récursif.

Ex 12. Construire une fonction qui retourne le minimum, le maximum et la moyenne des nombres contenus dans une liste passée en argument :

- Par parcours de la liste et comparaison
- A l'aide des opérations génériques Python

Manipulations de chaînes de caractères

Ex 13. Soit `ch='bonjour à tous...'`

- Calculer la longueur de la chaîne.
- Passer toute la chaîne en capitales.
- Remplacer toutes les occurrences de la lettre `o` par la lettre `a` dans `ch`.
- Retourner la liste de tous les mots de cette chaîne.

Ex 14. Modifier la fonction de l'Ex 4. pour qu'elle retourne le résultat à l'aide d'une unique chaîne de caractères.

Manipulations de dictionnaires

Ex 15. Soit `monDict` un dictionnaire contenant 3 items : (1,'un'), (2,'deux') et (4,'quatre').

- Retourner la valeur de la clé 1.
- Ajouter l'item : (3,'trois'). Supprimer l'item associé à la clé 2. Calculer la longueur du dictionnaire.
- Afficher le contenu du dictionnaire dans l'ordre décroissant des clés sous le format suivant :

Clé : 3, Valeur : trois

- Tester l'appartenance de 1 aux clés du dictionnaire ? aux valeurs du dictionnaire ?
- Éliminer tous les éléments du dictionnaire.

Ex 16. Écrire une fonction `compterLettres` ayant pour argument une chaîne de caractères et qui renvoie un dictionnaire contenant la fréquence de toutes les lettres.

Manipulations d'ensembles

Ex 17. Pour $S1 = \{a, b, c, d\}$ et $S2 = \{s, b, d\}$, afficher les résultats suivants :

- Ensemble initiaux,
- la valeur de $|S1|$, la valeur du test $a \in S1$
- les ensembles $S1 \cup S2$, $S1 \cap S2$, $S1 - S2$, $S2 - S1$

Gestion des entrées/sorties

Ex 18. Écrire une fonction qui demande la saisie d'un entier $n \in [1, 10]$ (en cas d'erreur de saisie, une nouvelle saisie est demandée). Affichez la valeur saisie.

Ex 19. Écrire une fonction qui demande le nom d'un fichier texte à ouvrir et retourne tout son contenu sous la forme d'une unique chaîne de caractères.

Ex 20. Modifier la fonction de l'Ex 19. pour qu'elle retourne le résultat sous la forme d'une liste de lignes.

Mise en forme de données

Ex 21. Construire une fonction qui retourne une chaîne de caractères représentant la table de multiplication d'un nombre n donné en argument. Le nombre de lignes de cette table est égal au maximum de 10 et n .

```

+=====+
| Table du 3 |
+=====+
| 1 x 3 | 3 |
+-----+
...
+-----+
| 10 x 3 | 30 |
+=====+

```

Ex 22. Construire une fonction qui construit automatiquement la page html donnant la table de multiplication d'un nombre n donné en argument. Le nombre de lignes de cette table est égal au maximum de 10 et n .

Problèmes

Les 3 problèmes suivants ont été conçus pour appréhender différentes notions autour de la programmation en Python. Pour chacun d'eux, des éléments de corrections sont donnés sur le site Moodle.

Pb 1. Tours de Hanoï

Notions abordées : Primitives de base, Instructions composées, Fonctions, Principe de la programmation récursive, Structure de données.

Le problème des tours de Hanoï est un jeu de réflexion consistant à déplacer des disques de diamètres différents d'une tour de « départ » à une tour d'« arrivée » en passant par une tour « intermédiaire » et ceci en un minimum de coups, tout en respectant les règles suivantes :

- Seul un disque est déplacé à la fois,
- Un disque ne peut être posé que sur un emplacement vide ou un disque de taille plus importante.

Travail demandé : Le programme recherché doit établir la liste des 127 mouvements à faire pour déplacer une tour de Hanoï composée de 7 disques aux couleurs de l'arc-en-ciel (Violet, Indigo, Bleu, Vert, Jaune, Orange, Rouge).

Chaque mouvement devra être défini par son numéro d'ordre, la couleur du disque à déplacer, le nom de la tour où le disque est pris et le nom de la tour où le disque est déposé.

Coup 26 : Orange de A à B

Conseils : Pour obtenir ce résultat, il est nécessaire de déterminer l'enchaînement des déplacements (de tour A à tour B) d'une part et de mémoriser le contenu courant de chaque tour d'autre part.

L'enchaînement des déplacements peut être déterminé par un algorithme récursif. Pour déplacer un empilement de « n » disques, il faut :

- dégager le disque inférieur, c'est-à-dire déplacer l'empilement de « $n-1$ » disposés au dessus vers le troisième emplacement,
- déplacer le disque inférieur vers la destination prévue de la tour,
- reposer sur ce disque l'empilement de « $n-1$ » disques qui étaient disposés au dessus.

Pb 2. Jeu du pendu

Notions abordées : Accès à un fichier, Fonction aléatoire, Interaction clavier, Manipulation de chaînes de caractères.

Le jeu du pendu consiste à trouver un mot en devinant les lettres qui le composent. Le joueur a droit à 10 propositions incorrectes (Proposer une seconde fois la même lettre doit être comptabilisé comme une proposition incorrecte).

Travail demandé : Le programme recherché doit :

- ouvrir un fichier texte¹ et lire son contenu,
- choisir une ligne au hasard parmi les lignes lues et extraire le mot qu'elle contient,
- transformer ce mot pour éliminer les accents et le mettre en majuscule,
- faire découvrir ce mot en faisant apparaître les lettres qu'il contient lorsque celles-ci sont proposées.

Conseils : Le module random comporte différentes fonctions pour générer des nombres aléatoires suivant différentes distributions.

L'affichage, à l'issue de la neuvième proposition incorrecte, pourrait être de la forme suivante :

```

  | / |
  | 0
  | / \
  | |
  | /
  |
  Mot à deviner : C _ E _ _ U
  ===== Proposez une lettre :

```

Pb 3. Classe « Fraction »

Notions abordées : Programmation d'une classe, Surcharge d'opérateurs

Travail demandé : Le programme recherché doit comporter votre propre module de calcul de fractions. Celles-ci doivent pouvoir être définies :

- par la donnée du numérateur et du dénominateur :

```
f1 = Fraction(2,3)
f2 = Fraction(-5)
f3 = Fraction()
```
- par composition de fractions préalablement définies :

```
f4 = f1+f2*f2
```

Les fractions doivent savoir se comparer afin de profiter des nombreuses fonctionnalités de python comme :

```

maListe = [ f1 , f2 , f3+f4 ]
maListe.sort() # Tri de la liste
print( "Liste triée :", maListe)

```

Conseils : Pour éviter de devoir tout appréhender simultanément, la méthodologie suivante peut être suivie :

- 1) Établir une première version de votre programme permettant de déclarer une fraction et de l'afficher.
- 2) Compléter votre programme pour intégrer la composition par les opérations de base.
- 3) Compléter la méthode `__init__` pour intégrer la simplification de la fraction à l'aide de l'algorithme d'Euclide.
- 4) Intégrer ensuite les méthodes de test.

1. Un fichier avec des noms d'animaux est disponible sur Moodle.