

Python objet – TP 1

1. Installation de Thonny et documentation

Pendant tout ce module, je vous propose de travailler avec l'éditeur Python Thonny : il est facile à prendre en main, et aide au débogage. Vous pouvez bien sûr utiliser un autre éditeur si vous préférez mais je vous aiderai moins efficacement en cas de bug.

Pour installer Thonny : <https://thonny.org/>

Documentation de Thonny : <https://realpython.com/python-thonny/>

Si vous avez des questions sur Python (syntaxe, méthodes existantes, techniques de programmation), je vous recommande les ressources suivantes :

le site officiel de Python : <https://docs.python.org/fr/3.7/index.html>

le livre « Apprendre à programmer avec Python 3 » de Gérard Swinnen (un certain nombre de modules et d'exemples seront tirés de ce livre) : http://inforef.be/swi/download/apprendre_python3_5.pdf

Nous utiliserons Python 3 (plus précisément la version 3.7.4). Vérifiez que vous utilisez bien au moins cette version.

2. Première partie : rappels de Python et d'algo

Exercice 1 – Calculs

- Ecrire la fonction `moyenne(a,b)` qui calcule et retourne la moyenne de deux entiers `a` et `b`.
- Ecrire la fonction `moyenne_ponderee(a,coef_a,b,coef_b)` qui calcule et retourne la moyenne de deux entiers `a` et `b` avec leurs coefficients.
- Ecrire la fonction `moyenne2(a,b)` qui calcule et retourne la moyenne de deux entiers `a` et `b` en utilisant la fonction `moyenne_ponderee`.

Exercice 2 – Conditions

- Ecrire la fonction `max2(a,b)` qui calcule et retourne le maximum de deux entiers `a` et `b`. On la nomme `max2` car il y a déjà une fonction `max` de base dans Python.
- Ecrire la fonction `max3(a,b,c)` qui calcule et retourne le maximum de trois entiers `a`, `b` et `c`.

Exercice 3 – Conditions avancées

- Ecrire la fonction `prixCinema(age)`, qui calcule et retourne le prix d'un ticket de cinéma : 5 euros pour les moins de 14 ans, 6,70 euros entre 14 et 18 ans, 10 euros pour les adultes.
- Ecrire la fonction `uneMinuteEnPlus(h,m)` qui retourne l'heure une minute après celle passée en paramètre. Exemple : `uneMinuteEnPlus(14,25)` retourne (14,26). Attention à ce qui se passe pour `uneMinuteEnPlus(23,59)` !

Exercice 4 – Listes

- Ecrire la fonction `sommeCarresListe(L)` qui calcule et retourne la somme des carrés des entiers d'une liste.
- Ecrire la fonction `moyenneListe(L)` qui calcule et retourne la moyenne des entiers de la liste.
- Ecrire la fonction `nbPairsListe(L)`, qui retourne le nombre d'entiers pairs de la liste.
- Ecrire la fonction `sommeCarres(L)` qui calcule et retourne la somme des carrés de 1 à `n` compris.
- Ecrire la fonction `estPremier(n)`, qui retourne `True` si `n` est premier, `False` sinon. On rappelle qu'un nombre premier `n` est divisible que par 1 et par lui-même. Par convention, 1 n'est pas premier, 2 l'est.

Notes et remarques

Le reste d'une division entre deux entiers est récupéré par l'opérateur « modulo », qui est noté `%`.

La fonction `range(a,b)` énumère la liste des nombres de `a` inclus à `b` exclu.

3. Deuxième partie : classes, objets, attributs, méthodes

Exercices 1 à 4 : voir feuille.

Exercice 5 – Géométrie

Définissez une classe Point() qui permette d'instancier des objets simulant un point géométrique à deux dimensions (une coordonnée x et une coordonnée y). Vous définirez une méthode afficher() qui permet de retourner les coordonnées du point.

Ecrivez une fonction (pas une méthode !) distance(p1,p2) qui permet de calculer la distance entre deux points donnés. Souvenez-vous du théorème de Pythagore... (Attention, vous aurez besoin d'importer le package math : import math pour utiliser la fonction sqrt(x) qui retourne la racine carrée d'un nombre).

Définissez une classe Rectangle() qui permette d'instancier des objets représentant des rectangles géométriques. Ils seront définis par un Point() (leur coin supérieur gauche), et deux dimensions : une hauteur et une largeur.

Ecrivez deux méthodes dans la classe Rectangle() : la méthode surface() renvoie l'aire du rectangle, la méthode afficher() renvoie le texte suivant : « Le rectangle est défini à partir du point (3,2) ». Attention : la fonction afficher() de Rectangle() doit utiliser celle de la classe Point() !

Exercice 6 – Etudiants et groupes

Définissez une classe Etudiant() qui permette d'instancier des objets simulant un étudiant, qui possède un nom, un prénom, un âge. Vous écrirez une méthode afficher() permettant de renvoyer un texte décrivant l'étudiant.

Définissez une classe Groupe() qui permette d'instancier des objets représentant des groupes d'étudiants. Chaque groupe possède un nom et une taille maximale (à fixer à l'instanciation), ainsi qu'une liste d'étudiants, vide au départ.

Ecrivez une méthode ajouter_etudiant(etu), qui ajoute un nouvel étudiant dans la liste. Evidemment, il faut afficher un message d'erreur si la taille maximale est dépassée !

Ecrivez une méthode age_moyen() qui retourne l'âge moyen du groupe. Attention au cas d'un groupe vide.

Enfin, écrivez une méthode afficher() qui décrit le groupe et chacun des étudiants qui le composent. Il faudra faire appel à la fonction afficher() de la classe Etudiant().