



## Introduction à la programmation (C5-101115-INFO)

Licence 1 - Année 2022/2023

### TP 10 - Fonctions, Listes, Dictionnaires

B. Besserer, C. Demko, P. Franco, L. Mascarilla, C. Saint-Jean, E. Zahzah



#### Exercice 1: Validation sur Moodle

Validez vos fonctions de l'exo 3 du TP9 sur moodle.

#### Exercice 2: Affichage d'une liste avec la tortue(turtle\_liste.py)

On a déjà vu comment afficher une liste dans la console avec la fonction *print*. On souhaite maintenant tracer une représentation d'une liste sous la forme d'un histogramme à l'aide du module *turtle*. Pour information :

```
setup(w, h) # Ouvre une fenêtre de taille w x h
setposition(x, y) # positionne la tortue en position (x, y)
penup() # relève le stylo
pendown() # abaisse le stylo
```

1. Définir une fonction *rect* à 4 paramètres qui trace un rectangle de largeur *l* et de hauteur *h* depuis la position (*x*, *y*). Par défaut, cette position sera (0,0). Tester votre fonction.
2. Utiliser cette fonction pour tracer 3 rectangles accolés de même largeur mais de hauteur différente.
3. Écrire la fonction *histo* à 4 paramètres qui prend comme entrées une liste, une largeur et les coordonnées du tracé (en bas à gauche). Celle-ci doit afficher la liste une succession de rectangles de même largeur *l* et dont la hauteur dépend des valeurs dans la liste.

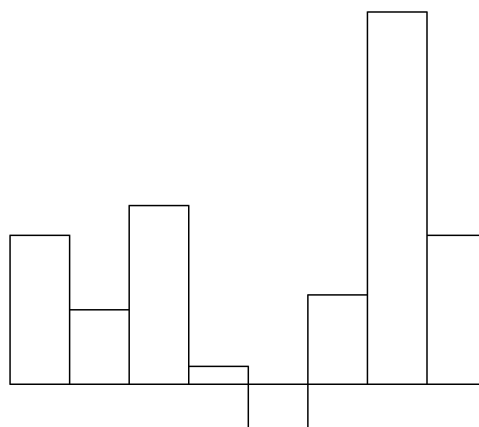


FIGURE 1 – Exemple de tracé pour  $L = [100, 50, 120, 12, -30, 60, 250, 100]$

4. (Bonus) Personnalisez votre affichage (axes, couleur, remplissage, hauteur maximale, ...)

### Exercice 3: Magasin (magasin.py)

---

Dans cet exercice, nous allons utiliser des dictionnaires pour gérer le stock d'un magasin.

Désignation	Prix Unitaire	Stock
Épée en mousse	5€	10
Masque Dark Vador	30€	4
Hand spinner 3d	10€	10
Console Minux	150€	2
Lego Footix	15€	5

1. À partir du tableau ci-dessous, définir les deux dictionnaires  $d_{pu}$  et  $d_{stock}$  dans lesquels la clé est la désignation du produit.

*Indication : Faites du "copier-coller".*

2. Écrire une fonction  $verification(d1, d2)$  à valeur booléenne qui vérifie que deux dictionnaires ont exactement les mêmes clés.

Exemple d'appel :

---

```
>>> verification(d_pu, d_stock)
True
```

---

3. Écrire une fonction  $achat\_possible(k, v, d)$  à valeur booléenne qui indique si l'on peut acheter le produit  $k$  en quantité  $v$ . On testera la présence de la clé  $k$  dans  $d$  et si la quantité disponible est suffisante.

Exemples d'appel :

---

```
>>> achat_possible("Hand spinner 3d", 2, d_stock)
True
>>> achat_possible("Console Minux", 5, d_stock)
False
```

---

4. On souhaite généraliser cette fonction au cas où les achats sont passés sous la forme d'un dictionnaire.

Exemples d'appel :

---

```
>>> achats_possibles({"Hand spinner 3d": 2}, d_stock)
True
>>> d_achats = {"Hand spinner 3d": 2, "Console Minux": 5}
>> achats_possibles(d_achats, d_stock)
False
```

---

(bonus) Enfin, réaliser une fonction  $achats$  qui :

- (a) Vérifie via la fonction précédente si l'achat est possible.
- (b) Affiche une facturation à la demande.
- (c) Met à jour le stock.

Exemples d'appel :

---

```
>>> achats({"Hand spinner 3d": 2, "Lego Footix": 1}, d_pu, d_stock, True)
Hand spinner 3d : 2 x 10 = 20
Lego Footix : 1 x 15 = 15
-----
Prix total: 35
>>> print(d_stock)
{'Épée en mousse': 10, 'Masque Dark Vador': 4, 'Hand spinner 3d': 8,
 'Console Minux': 2, 'Lego Footix': 4}
```

---