

Rappel: Nous travaillons sur spyder après installation via la distribution anaconda

Ce TD est constitué de deux parties

la première partie est décrite dans ce document, la deuxième est à réaliser sur la plateforme [knowledgeable.com](https://www.knowledgeable.com)

Privilégiez l'utilisation des section (### nomSection, et ctrl+enter pour exécuter) cette façon de coder permettra de préserver le fruit de la séance et ne pas passer son temps à commenter le code une fois l'exercice terminé

Vous pouvez aussi utiliser une approche basée sur un main. dans un premier temps je préfère l'utilisation des sections

```
# %% zone de l'exo zéro elle doit contenir l'Exo 0 et les méthodes afférentes
def Exo0():
    """
    Il s'agit d'une méthode à titre d'exemple
    l'instruction help(Exo0) vous retourne cette documentation
    grâce à l'utilisation des triples guillemets
    ou bien dans la console de spyder saisissez Exo0?
    """
    print("voilà mon Exo de bienvenue")

# %% zone de l'exo1
def Exo1():
    print("Exo1")

# %% zone du main

if __name__ == '__main__':
    #Exo0()
    Exo1()
```

Exo 1: Affectez les variables temps et distance par les valeurs 6.892 et 19.7.

- Calculez et affichez la valeur de la vitesse.
- Améliorez l'affichage en imposant deux chiffres après le point décimal.

Exo 2: Écrire un programme calculant le maximum et le minimum de deux nombres.

- avec un test à une alternative
- avec un test simple
- avec un test ternaire

Exo 3: Ecrire la fonction **volBoite(x1,x2,x3)** de manière à ce qu'elle puisse être appelée avec un, deux, ou trois arguments.

- 1 argument : la boîte est considérée comme cubique (l'argument étant l'arête de ce cube).

- 2 arguments : la boîte est considérée comme un prisme à base carrée. (Dans ce cas le premier argument est le côté du carré, et le second la hauteur du prisme).

- 3 arguments : la boîte est considérée comme un parallélépipède.

Par exemple :

```
print volBoite() # donne -1 ou None (-> erreur).
```

```
print volBoite(5.2) # donne 140.608
```

```
print volBoite(5.2, 3) # donne 81.12
```

```
print volBoite(5.2, 3, 7.4) # donne 115.44
```

Exo 4: Définissez une fonction **eleMax(liste,debut,fin)** qui renvoie l'élément ayant la plus grande valeur dans la liste transmise. Les deux arguments **debut** et **fin** indiqueront les indices entre lesquels doit s'exercer la recherche, et chacun d'eux pourra être omis (cf. exercice précédent).

Exemples de la fonctionnalité attendue :

```
>>> serie = [9, 3, 6, 1, 7, 5, 4, 8, 2]
```

```
>>> print(eleMax(serie))
```

```
9
```

```
>>> print(eleMax(serie, 2, 5))
```

```
7
```

```
>>> print(eleMax(serie, 2))
```

```
8
```

```
>>> print(eleMax(serie, fin=3, debut=1))
```

```
6
```

Exo 5: Soient les listes suivantes :

```
t1 = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
```

```
t2 = ['Janvier', 'Février', 'Mars', 'Avril', 'Mai', 'Juin', 'Juillet', 'Août', 'Septembre', 'Octobre', 'Novembre', 'Décembre']
```

Écrivez une fonction qui retourne une nouvelle liste t3.

Celle-ci contiendra tous les éléments des 2 listes en les alternant :

```
t3 = ['Janvier', 31, 'Février', 28, 'Mars', 31, 'Avril', 30, etc...]
```

Exo 6: Ecrivez une fonction qui recopie une chaîne de caractères (dans une nouvelle variable) en l'inversant. Exemple : "zorglub" devient "bulgroz"

Exo 7: Que fait cette fonction? testez-là

```
def mafonction(arr):
    n = len(arr)
    for i in range(n):
        for j in range(0, n-i-1):
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]
```

Exo 8: Countdown

You are provided with an integer named **start**, the goal of the exercise is to display the countdown from start to 1 included, then "GO!"

take for example, start = 3, we will display "3 2 1 GO!"

Exo 9 : Condition : palindrome

Read a word from the standard input.

if the word is palindrome, display to screen "<word> is palindrome".

otherwise display "<word> is not palindrome"

Exo 10 : List : combinaisons

You have at your disposal a list named `my_list=[0, 1, 2, 3, 4]`

Create:

a list `l1` containing the elements of `my_list`, followed by the elements of `my_list` backwards

a list `l2` containing 3 times the first three elements of `my_list`

a list `l3` containing all the elements of `my_list` whose index is a multiple of 3

Exo 11 : Parity test

Read an integer in a variable `x`. If `x` is even, display "pair", otherwise display "impair".

Examples:

- if `x` is **0**, we display **"pair"**
- if `x` is **10**, we display **"pair"**
- if `x` is **-5**, we display **"impair"**
- if `x` is **3**, we display **"impair"**

a list `l3` containing all the elements of `my_list` whose index is a multiple of 3

Exo 12 : Acronyme

Read a string **phrase from the standard input**. This non-zero and non-empty string will contain letters and spaces.

The purpose of this exercise is to generate and display the acronym for this sentence, in capital letters.

For example :

- if phrase is "Société Nationale Chemin Fer", we will display ***SNCF***
- if phrase is "Train grande vitesse", we will display ***TGV***

Exo 13 : You are provided with a string `chaine`, and a character `c`.

The purpose of this exercise is to count the number of times `c` appears in **`chaine`**.

You will then display `'<c>' apparaît <nb> fois dans "<chaine>"`

For example :

- if `c` is 'S' and **`chaine`** is "Hi people", we will display: *'S' apparaît 1 fois dans "Hi people"*
- if `c` is 'e' and **`chaine`** is "Hi people", we will display: *'e' apparaît 2 fois dans "Hi people"*
- if `c` is '?' and **`chaine`** is "Hi people", we will display: *'?' apparaît 0 fois dans "Hi people"*

Exo 14 : Comptage de voyelles

Ecrire une fonction **`vowel_count`**, qui prend en paramètres une **`chaine`** de caractères, et retourne le nombre de voyelles qu'elle contient.

Pour rappel, les voyelles sont a,e,i,o,u,y (majuscules et minuscules).

Exo 15: Dictionnaires : creation

Create a dictionary (associative array) named **my_dict**, which contains the keys:

- "pi", associated with the real equal to 3.14
- "mot", associated with the character string "mot"
- "nombre", associated with the integer 42
- "liste", associated with the list [1,2,3]

Exo 16 : key in dict

Read a word on standard input using **input ()**.

If this word corresponds to a key in my_dict, display "<mot> vaut <valeur> dans my_dict".

Otherwise display "<mot> n'est pas une cle de my_dict"

Exo 17 :Dictionnaires : modifications

You have at your disposal a dictionary (associative array) named **my_dict**.

In order :

- add a key "hello", associated to the value "world"
- modify the key "nombre", to give it the value "0"
- delete the key "pi"

Exo 18: création de set

On vous fournit une liste **values**

Créez :

1. un set **s1** contenant 1,2,3
2. un set **s2** contenant les caractères de "Hello World"
3. un set **s3** contenant les valeurs de la liste **values=[10,20,30,40,10,2,40]**
4. un set **s4** contenant les entiers allant de 5 à 15 compris

Partie 2 sur Nowledgeable <https://nowledgeable.com>