

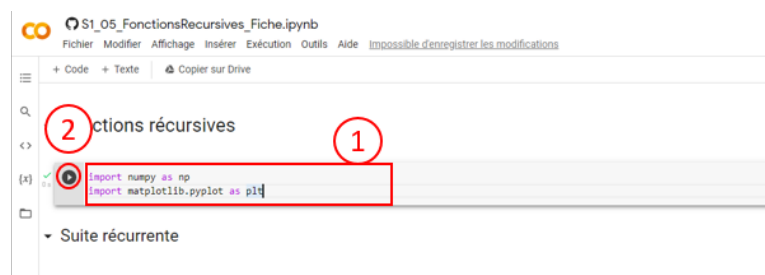
Fonctions récursives – Activités préparatoires.

Activité 1

Tester l'algorithme bubble vu dans le cours.

Activité 2

L'activité est à réaliser sur un « notebook ». En ouvrant le lien suivant vous aurez des activités à traiter dans lesquelles vous pourrez modifier le code python et l'exécuter.



① Cliquer sur la zone de code puis modifier comme vous le souhaitez.

② Cliquer sur le bouton pour tester votre code.

Réaliser les activités disponibles au lien suivant : sont disponibles aux liens suivants : sujet – <https://bit.ly/3rcT81B> et corrige – <https://bit.ly/3z17QKJ>.

Activité 3 – QCM

Question 1 Que retourne la commande suivante `mystere(4)` ?

```
def mystere(n):
    if n>0 :
        return mystere(n-2)
    else :
        return n==0
```

- 1°) 0.
- 2°) False.
- 3°) True.
- 4°) L'exécution génère une erreur.

Question 2 Laquelle de ces fonctions retourne True lorsqu'on exécute `f(5)` ?

```
def f1(n):
```

```
if n==0 :
    return True
else :
    return f1(n-2)
```

```
def f2(n):
    if n<=0 :
        return True
    else :
        f2(n-2)
```

```
def f3(n):
    if n<=0 :
        return True
    return f3(n-2)
```

```
def f4(n):
    if n==0 :
        return True
```

f4(n-2)

- 1°) f1.
- 2°) f2.
- 3°) f3.
- 4°) f4.

Question 3 Quel affichage obtient-on en exécutant `affiche(3)` ?

```
def affiche(n):  
    print(n)  
    if n>=0:  
        affiche(n-1)
```

- 1°) 3, 2, 1, 0 (avec des retours à la ligne entre chaque valeurs).
- 2°) 0, 1, 2, 3 (avec des retours à la ligne entre chaque valeurs).
- 3°) 3, 2, 1, 0, -1 (avec des retours à la ligne entre chaque valeurs).
- 4°) 3.

Question 4 Une seule des fonctions définies ci-dessous retourne 'cccc' à l'appel de `replique(5, 'c')`. Déterminer laquelle.

```
def replique(a,b): # Fonction 1  
    if a==1:  
        return b  
    else :  
        return replique( a-1 , b+b)  
  
def replique(a,b): # Fonction 2  
    if a==1:
```

```
        return b  
    elif a%2 == 0:  
        return replique( a-2 , b+b)  
    else :  
        return b + replique( a-2 , b+b)
```

```
def replique(a,b): # Fonction 3  
    if a==1:  
        return b  
    elif a%2 == 0:  
        return replique( a//2 , b+b)  
    else :  
        return b + replique( a//2 , b+b)
```

```
def replique(a,b): # Fonction 4  
    if a==1:  
        return b  
    else :  
        replique( a-1 , b+b)
```

- 1°) Fonction 1.
- 2°) Fonction 2.
- 3°) Fonction 3.
- 4°) Fonction 4.

Question 5 Que retourne l'instruction `copy(3, 'A')` ?

```
def copy(n,s):  
    if n==0:  
        return s  
    return copy(n-1, s+s)
```

- 1°) 'AAA'.
- 2°) 'AAAAAA'.
- 3°) 'AAAAAAAA'.
- 4°) '3A'.