- A faire avant à la maison :
- A faire en classe :
  - Exercice 1 : avec une fonction (partie 1)
  - Exercice 2 : avec une fonction (partie 2)
  - Exercice 3 : avec deux fonctions
  - Exercice 4 : une fonction et deux appels
- A faire après à la maison :
  - Exercice 5 : trois programmes pour un même résultat
- A faire en classe la fois suivante :

A faire avant à la maison :

- Revoir le cours de Première sur les fonctions.
- Cette révision peut s'accompagner éventuellement d'une petite fiche d'exercices de base sur les fonctions.

A faire en classe :

Le professeur fait un point rapide sur ce que les élèves ont revu à la maison sur les fonctions.

Il donne ensuite le notebook *NSI Récursivité (1/2) : notions de fonctions* que les élèves auront à traiter en autonomie. Le professeur passe dans les rangs pour vérifier l'avancée des travaux, répondre aux questions et donner des conseils. Il n'hésitera pas à faire des points bilans régulièrement (par exemple à la fin de chaque exercice)

L'activité proposée dans ce notebook a pour but de revoir et compléter quelques notions sur les appels de fonctions et de prendre en main quelques outils simples de visualisations graphiques : **Python Tutor** et **Recursion Visualizer**.

Elle est composé de cinq exercices progressifs sur la notion de fonction.

Exercice 1 : avec une fonction (partie 1)

On considère ici le code suivant :

```
def f(x):
    return 2*x + 1

def ajouter(x):
    return 2 + f(x)
```

Dans un premier temps, les élèves calculent à la main les appels successifs sur ces deux fonctions. Ils écrivent ensuite la description de ces appels en utilisant, d'une part, Python Tutor (Figure 1) et, d'autre part, RecursionVisualizer (Figure 2).

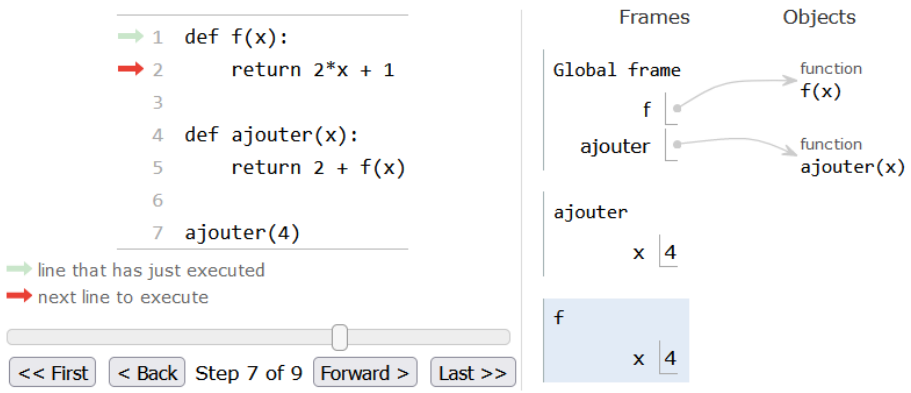


Figure 1 : exemple d'affichage obtenu à l'aide de Python Tutor

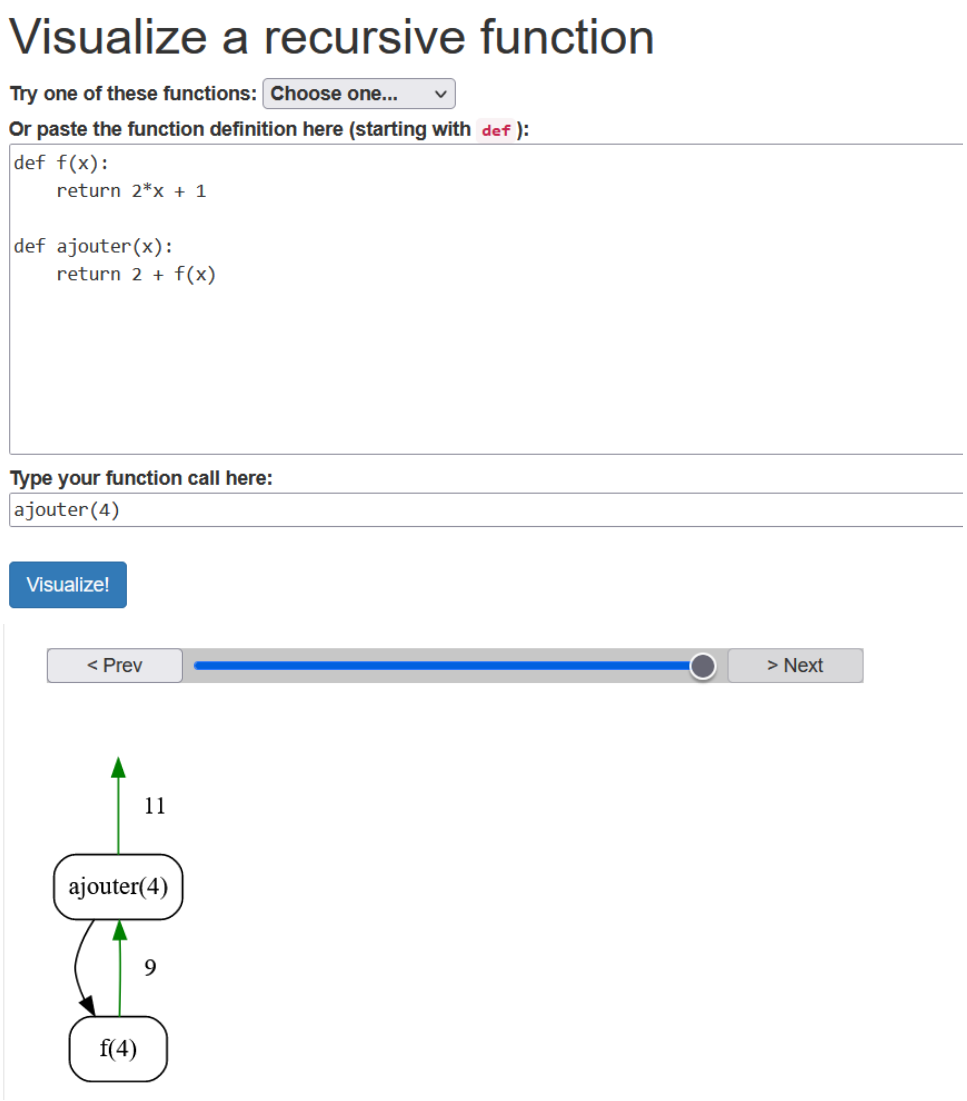


Figure 2 : exemple d'affichage obtenu à l'aide de RecursionVisualizer

L'intérêt de ces deux outils numériques permet aux élèves de bien comprendre l'ordre des appels et des opérations (ici : appel, puis addition).

Exercice 2 : avec une fonction (partie 2)

On considère à présent le code Python suivant :

```
def f(x):
    return 2*x + 1

def dilatation(x):
    return f(3*x)
```

Comme dans le premier exercice, les élèves calculent tout d'abord à la main les appels successifs sur ces deux fonctions, puis écrivent la description de ces appels en utilisant, d'une part, Python Tutor (Figure 1) et, d'autre part, RecursionVisualizer (Figure 2).

Ici, on inverse l'ordre des appels et opération : multiplication et appel

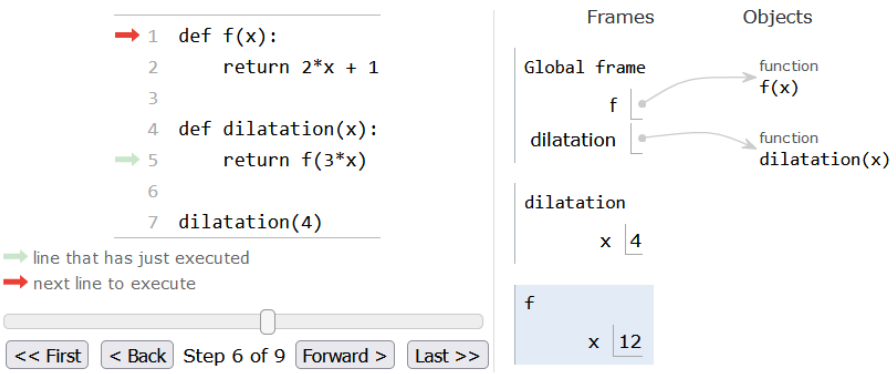
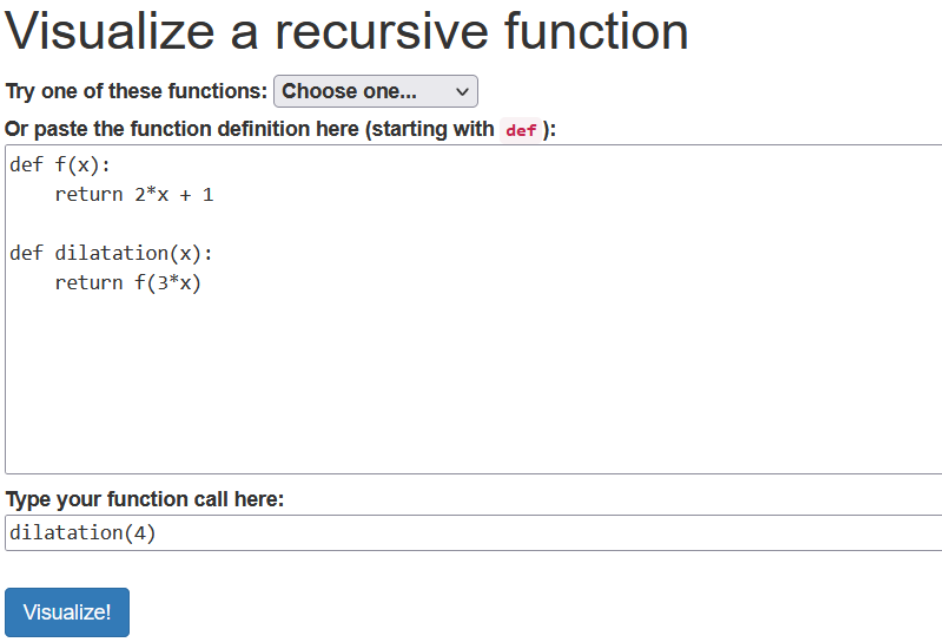


Figure 1 : exemple d'affichage obtenu à l'aide de Python Tutor



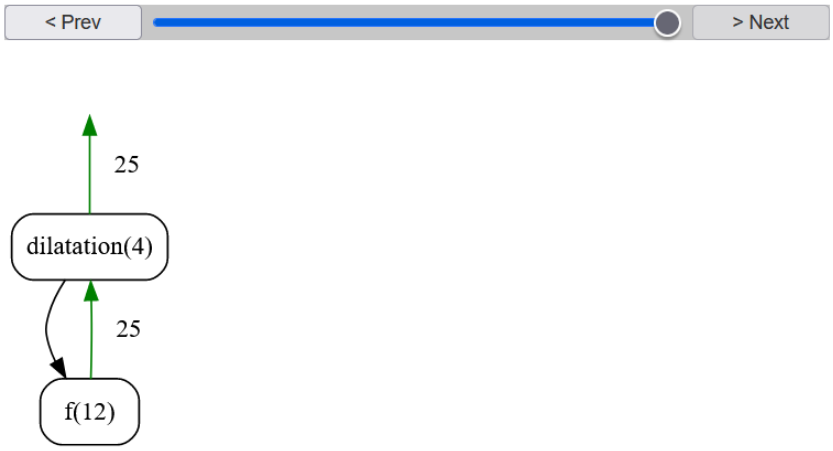


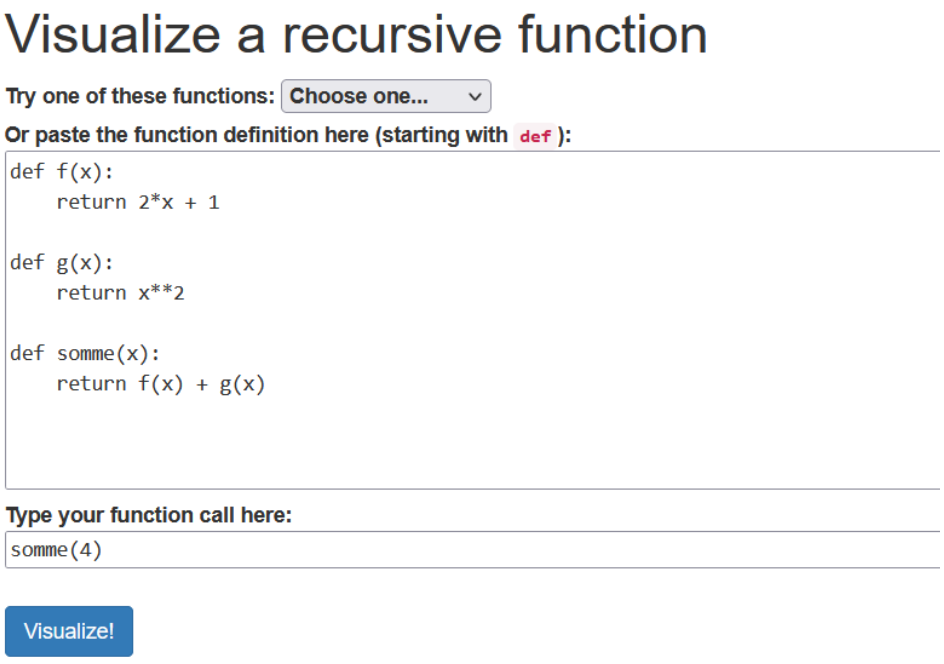
Figure 2 : exemple d'affichage obtenu à l'aide de RecursionVisualizer

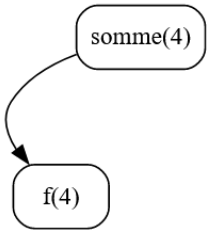
Exercice 3 : avec deux fonctions

Dans cet exercice, on considère le code Python suivant :

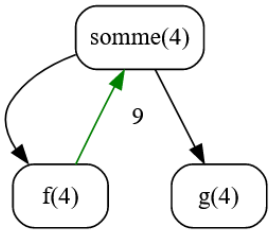
```
def f(x):
    return 2*x + 1

def g(x):
    return x**2

def somme(x):
    return f(x) + g(x)
```

Dans un premier temps, les élèves utilisent l'outil RecursionVisualizer afin de déterminer l'ordre des appels effectués comme le montre la figure ci-dessous (ici : appel puis appel puis addition).

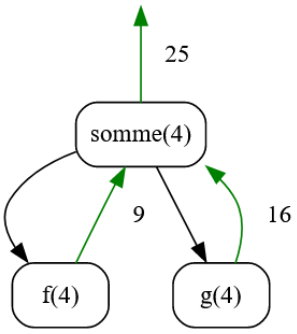




Etape 1 : appel de la fonction f

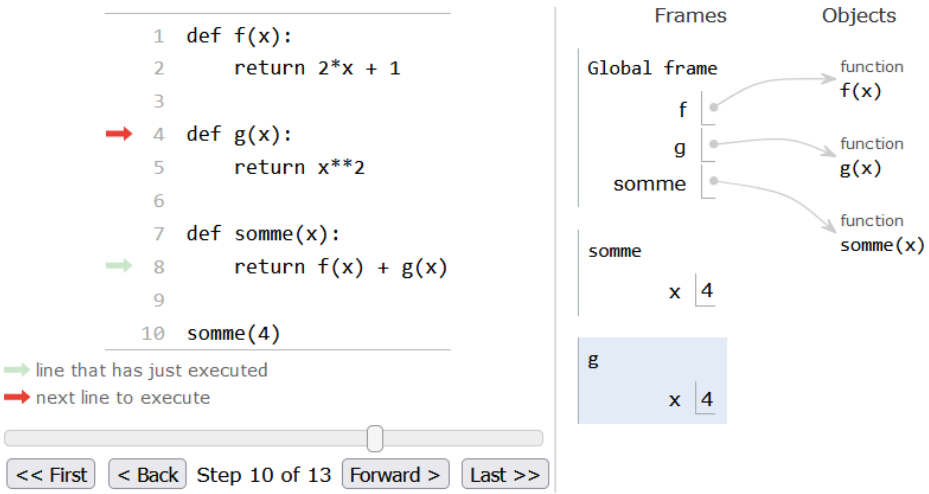


Etape 2 : appel de la fonction g

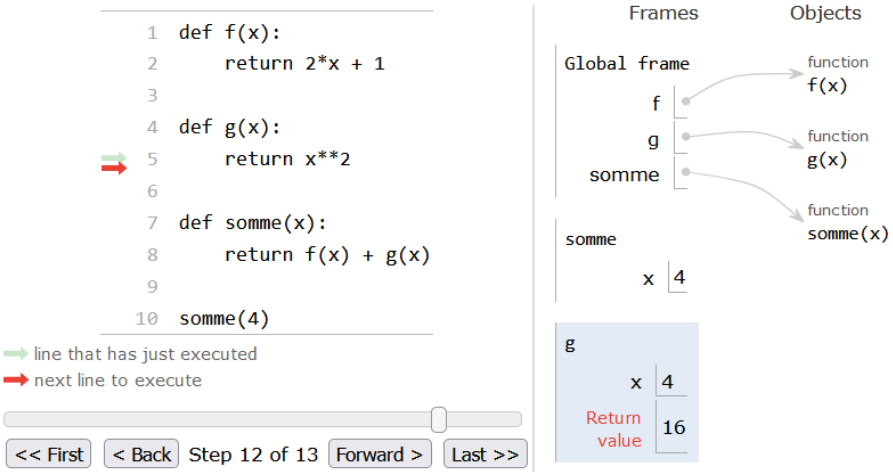


Etape 3 : calcul de la somme f(4) + g(4)

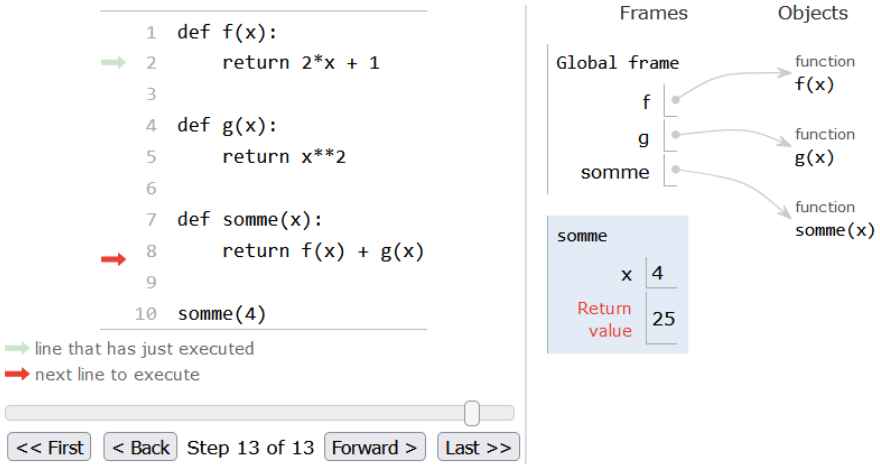
Dans un second temps, les élèves s'appuient sur Python Tutor pour comprendre que lorque le résultat est renvoyé, le contexte dans lequel s'est effectué le calcul est effacé de la mémoire (voir figure ci-dessous). On commence ici à mettre en place la notion de **pile d'appels**.



Etape 1 : appel de la fonction g



Etape 2 : exécution en cours de la fonction g



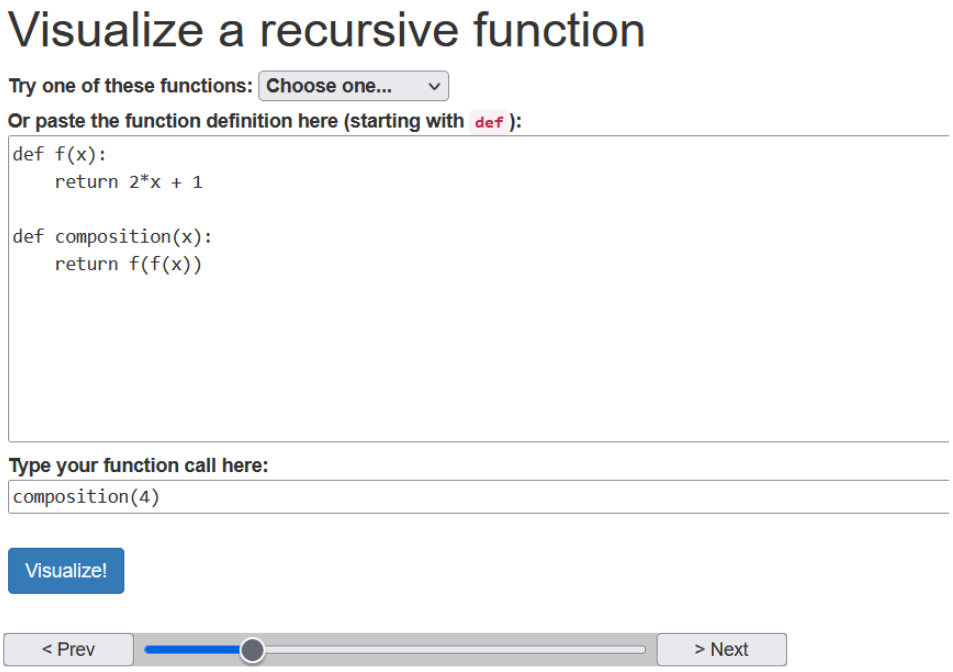
Etape 3 : renvoie et destruction du bloc d'exécution de la fonction g

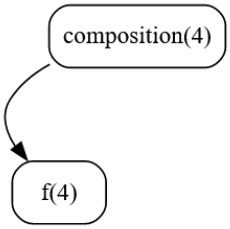
Exercice 4 : une fonction et deux appels

On considère à présent le code Python suivant :

```
def f(x):
    return 2*x + 1

def composition(x):
    return f(f(x))
```

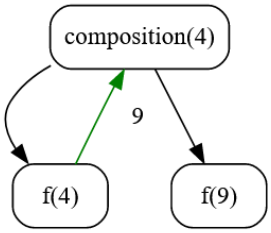
On reprend l'idée de l'exercice précédent : en s'appuyant sur l'outil RecursionVisualizer, les élèves déterminent l'ordre des appels effectués.





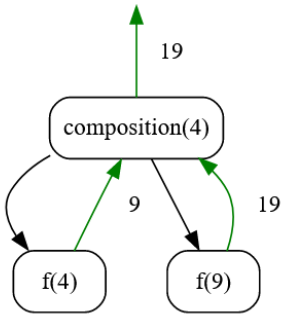
Etape 1 : premier appel de la fonction f avec le paramètre 4

< Prev > Next



Etape 2 : renvoie du premier appel (valeur 9), puis second appel de la fonction f avec le paramètre 9

< Prev > Next



Etape 3 : renvoie du second appel (valeur 19), puis fin de l'exécution de la fonction composition

L'intérêt de cet exercice est de montrer aux élèves qu'une même fonction peut être appelée plusieurs fois. Plus précisément, nous sommes ici dans le cas d'une fonction qui appelle la même fonction avec des paramètres différents. Le professeur peut prolonger ici l'exercice en utilisant Python Tutor pour montrer que lors de ce double appel de la fonction f, on a deux contextes différents en mémoire

A faire après à la maison :

Exercice 5 : trois programmes pour un même résultat

Dans cet exercice, on s'intéresse à la simulation de la suite récurrente $u_{n+1} = 2u_n + 1$ avec $u_0 = 1$.

Trois élèves (Charles, Pascal et Mathilde) proposent chacun trois programmes pour répondre à cette question :

- Programme de Charles :

```
def calcul_terme_direct(n):
    u = 1
    for i in range(1, n+1):
        u = 2*u + 1
    return u
```

- Programme de Pascal :

```
def f(x):
    return 2*x + 1

def calcul_terme_avec_f(n):
    u = 1
    for i in range(1, n+1):
        u = f(u)
    return u
```

- Programme de Mathilde :

```
def f(x):
    return 2*x + 1

def calcul_mystérieux_avec_f(n):
    if n == 0:
        return 1
    else:
        return f(calcul_mystérieux_avec_f(n-1))
```

Pour chacun de ces trois programmes, les élèves doivent préciser dans quels ordres sont effectués les différents appels et renvois entre les deux fonctions en utilisant RecursionVisualizer.

L'objectif de cet exercice est triple :

- Il met en perpective trois visions du problème initial.
- Il permet de manipuler des appels successifs de fonctions.
- La proposition de Mathilde met en évidence le principe de récursivité.

A faire en classe la fois suivante :

Le professeur propose avec l'aide de la classe une correcion de l'exercice 5, puis effectue un bilan de l'activité avec les élèves.