TP2 : Fonctions et boucle for

1 Exercice préparatoire

Exercice 1 *Autour de la commande diag()*

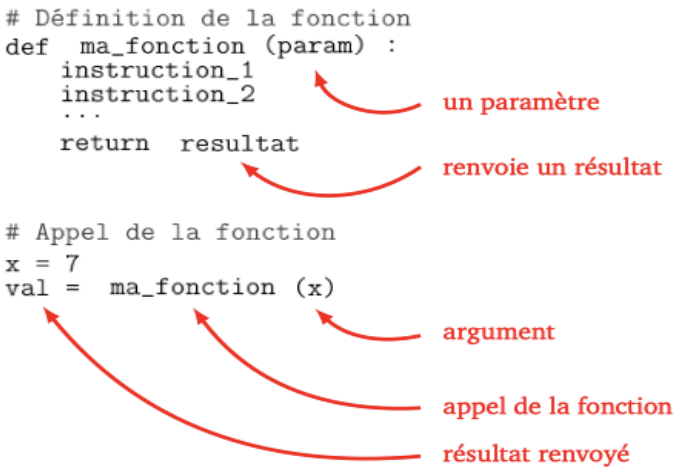
1. Que fait la commande $diag(array([1,2,3]))$?
2. Donner le code permettant d'obtenir $X = \begin{pmatrix}1 & 2 & 3 & 4\end{pmatrix}$ en utilisant *linspace()*.
3. Donner les lignes de code permettant d'obtenir la matrice suivante (utilisez *diag()* et *ones()* !)

$$A = \begin{pmatrix}2 & 1 & 1 & 1\\ 1 & 3 & 1 & 1\\ 1 & 1 & 4 & 1\\ 1 & 1 & 1 & 5\end{pmatrix}$$

4. Construire une matrice de taille 10×10 ne contenant que des 3 sur la diagonale et des 5 partout ailleurs.

2 Création de fonctions

La syntaxe pour créer une fonction est :



- **resultat** est la variable de sortie de la fonction
- **param** est la variable d'entrée de la fonction (il peut y en avoir plusieurs)

On remplace **ma_fonction** par le nom que l'on souhaite donner à la fonction.

⚠ ⚠ Les fonctions Python n'ont pas de *end* explicite, ni d'accolades qui pourraient marquer là où se termine le code de la fonction. Les seuls délimiteurs sont les deux points " :" et l'indentation du code lui-même.

L'indentation démarre le bloc et la désindendation le termine. Cela signifie que les espaces blancs sont significatifs et qu'ils doivent être cohérents.

⚠ Une fonction peut ne pas contenir de variable d'entrée et/ou de variable de sortie.

Les fichiers de fonctions permettent à l'utilisateur de définir des fonctions qui ne figurent pas parmi les fonctions prédéfinies et de les utiliser de la même manière que ces dernières (ces fonctions sont appelées fonctions utilisateur).

Exemple 1. *Implémentons la fonction $f(x) = x^2 + 1$. Pour cela, on entre dans un script la séquence d'instruction suivante :*

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

Pour tester la fonction, il suffit de rentrer f(1) dans un script ou dans la fenêtre de commande

Exemple 2. *Pour implémenter une fonction qui prend en entrée deux paramètres a et b et qui renvoie une matrice carrée de taille 2 avec des a sur la première ligne et des b sur la deuxième ligne, on entre dans un script la séquence suivante :*

```
def matrice(a,b) :
    A=a*ones([1,2])
    B=b*ones([1,2])
    C=concatenate((A,B))
    return C
```

Pour tester la fonction, il suffit de rentrer matrice$(3, 17)$ *dans un script ou dans la fenêtre de commande*

Exercice 2 On considère la fonction $g(x) = x^2 + 3x - 1$.

1. Dans un script tapez le code suivant :
```
def g(x) :
    y=x**2+3*x-1
    return y
```
2. Tester la fonction en l'utilisant dans la fenêtre de commandes pour calculer $g(1)$ et $g(2)$.
3. Taper dans la fenêtre de commande :
```
X=array([[1,2],[3,4]])
Y = g(X)
```
Le résultat de $g(X)$ est-il celui d'un calcul matriciel ou est-il le résultat terme à terme ?

Exercice 3 : L'UE23 est composée de 3 notes :

1. une note d'anglais, avec un coefficient 2
2. une note de mathématiques, avec un coefficient 3
3. une note de culture et communication, avec un coefficient 2

Ecrire une fonction, appelé moyenneUE23, prenant en paramètres d'entrées ces trois notes et qui renvoie la moyenne de l'UE23.

Exercice 4 : Fonctions dont la sortie est une matrice :

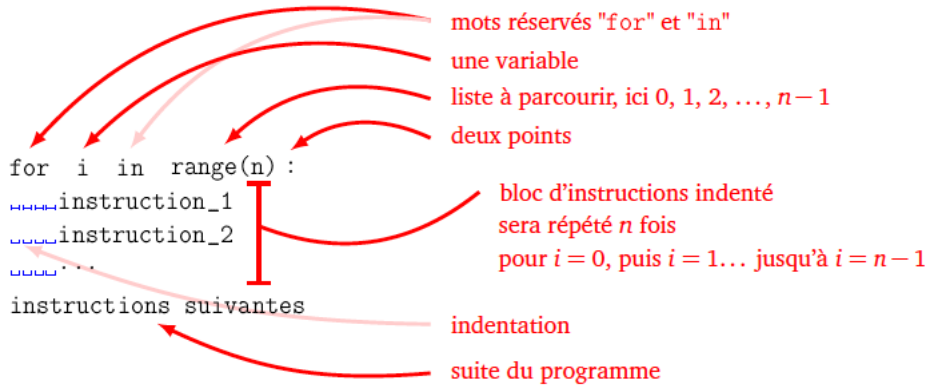
1. Écrire une fonction, nommée twos, qui prend pour variables d'entrées deux entiers n et p et qui renvoie une matrice de taille $n \times p$ et dont tous les coefficients valent 2.
2. Écrire une fonction, nommée identite, qui prend pour variable d'entrée un entier n et qui renvoie la matrice identité de taille $n \times n$ (avec des 1 sur la diagonale et des 0 ailleurs).
3. Écrire une fonction, nommée matrice1, qui prend pour variable d'entrée un entier n et qui renvoie une matrice carrée de taille $n \times n$ telle que les $n-1$ premières lignes sont composées de 2 et la dernière ligne est composée de zéros.

Remarque : La commande $eye(n)$, de la bibliothèque pylab, renvoie la matrice identité de taille n.

3 La boucle FOR

Une possibilité pour exécuter une séquence d'instructions de manière répétée consiste à effectuer une boucle pour les valeurs d'un indice qui parcourt une liste.

C'est le principe de la boucle for. La syntaxe est la suivante :



⚠ ⚠ Les boucles for Python n'ont pas de *end* explicite, ni d'accolades qui pourraient marquer là où se termine le code de la fonction. Les seuls délimiteurs sont les deux points " :" et l'indentation du code lui-même.

L'indentation démarre le bloc et la désindendation le termine. Cela signifie que les espaces blancs sont significatifs et qu'ils doivent être cohérents.

Exercice 5

1. Dans un script, entrer les commandes suivantes :

```
a=0
for i in range(10) :
    print('a=',a)
    print('i=',i)
    a=a+i
print('a=',a)
```

2. Observer les valeurs de a et i. Expliquer ce que fait le programme.
3. Que se passe-t-il si l'on modifie le code comme suit :

```
a=0
for i in range(3,10,2) :
    print('a=',a)
    print('i=',i)
    a=a+i
print('a=',a)
```

4. Ecrire une fonction, nommée somme, qui prend en paramètre un entier N et qui calcule $\sum_{k=1}^{N} k$.

Exercice 6 : Soient n et p deux entiers naturels. On considère les matrices de tailles $n \times p$ dont les coefficients sont donnés par les formules :

$$A(i,j) = i + j \qquad \text{et} \qquad B(i,j) = 2^i \times 3^j$$

1. On suppose dans cette question que n=2 et p=3. Que valent A et B ?
2. Écrire, à l'aide de deux boucles for, la fonction prenant en entrées les entiers n et p et renvoyant la matrice A.
3. Même question pour la matrice B.

Exercice 7 : À l'aide d'une boucle for, écrire une fonction qui prend en paramètre un entier naturel n et qui retourne en sortie $n!$
On rappelle que $n! = 1 \times 2 \times 3 \times ... \times (n-1) \times n$. Par exemple : $5! = 120$.

4 Exercices complémentaires

Exercice 8
Écrire une fonction qui prend en paramètre un entier naturel n et qui retourne en sortie une matrice A de taille n telle que

- $A(i,i) = \dfrac{1}{n}$ pour $i = 1, ..., n$,
- $A(1,n) = n!$,
- et contenant des 1 partout ailleurs.

On utilisera la fonction factorielle créée dans l'exercice précédent.
Exemple : Pour $n = 4$, la fonction doit renvoyer la matrice suivante :

$$A = \begin{pmatrix} 0.25 & 1 & 1 & 24 \\ 1 & 0.25 & 1 & 1 \\ 1 & 1 & 0.25 & 1 \\ 1 & 1 & 1 & 0.25 \end{pmatrix}$$

Exercice 9
À l'aide de deux boucles for, écrire une fonction qui prend en paramètre un entier naturel n et qui retourne en sortie une matrice A de taille n telle que

$$A(i,j) = \begin{cases} i+j & \text{si} \quad i > j, \\ i-j & \text{si} \quad i \leq j. \end{cases}$$

Exemple : Pour $n = 4$, la fonction doit renvoyer la matrice suivante :

$$A = \begin{pmatrix} 0 & -1 & -2 & -3 \\ 3 & 0 & -1 & -2 \\ 4 & 5 & 0 & -1 \\ 5 & 6 & 7 & 0 \end{pmatrix}$$