

TP 1 – Mise en jambe

Cette série d'exercices couvre le premier chapitre du cours (valeurs, variables et types).

► Exercice 1 : Test en ligne

Objectifs : aider la science à avancer, contribuer à la recherche en didactique informatique

Avant de commencer les exercices à proprement parler, nous aimerions vous inviter à remplir le questionnaire suivant:

<https://nettskjema.no/a/pika2025>

Il s'agit d'un test anonymisé mis en place pour permettre d'étudier l'impact de l'enseignement au lycée sur les compétences en programmation des étudiants entrant à l'université. Merci !

► Exercice 2 : LightBot

Objectifs : se familiariser avec un environnement de type "tortue" ; optimiser un algorithme ; utiliser le serveur Discord de la L1 ; s'assurer de pouvoir contacter le chargé de TP par Discord et par mail ; s'amuser un peu.

1. Rendez vous sur <https://lightbot.lu/> et effectuez les niveaux 0, 1, 2, 3, 4, 5, 8, 9, 10.
2. Lorsque vous avez fini, envoyez une capture d'écran à votre chargé.e de TP par Discord *et* par email.

► Exercice 3 : Se familiariser avec Thonny

Objectifs : lire et faire fonctionner un algorithme ; différencier valeurs et variables ; appréhender la notion de concaténation de chaînes ; traduire un algorithme en Python ; apprendre à utiliser Thonny.

On considère l'algorithme suivant :

Algorithm 1: Un algorithme manipulant des variables

```
1 riri ← "fifi"
2 fifi ← "riri"
3 loulou ← fifi + "loulou"
4 riri ← loulou × 2
```

1. Quel est l'état des variables à la fin de l'exécution de cet algorithme ?
2. Lancez l'IDE Thonny puis assurez-vous que les zones **Console** et **Variables** sont affichées dans la fenêtre de Thonny. Vous pouvez choisir les onglets affichés dans l'onglet **Affichage**.
3. A l'aide de Thonny, écrivez puis exécutez le programme Python correspondant à l'algorithme donné.

Vérifiez l'état des variables dans la zone **Variables**.

4. Dans la zone **Console**, “appelez” successivement chacune des variables pour en vérifier à nouveau le contenu.

► Exercice 4 : Exécuter un programme dans Thonny

Objectifs : Effectuer quelques opérations sur des variables et observer le résultat ; récupérer le type d’une variable ; se familiariser avec l’utilisation de Thonny.

1. Dans Thonny, écrivez le mini-programme suivant.

```
a = 5
b = "5"
c = a/5
d = a//5
e = b*a
```

2. Dans la zone **Variables** de Thonny visualisez l’état des variables après l’exécution de ce programme.
3. La fonction Python `type(a)` permet de récupérer le type de la variable **a**.

Dans la zone **Console** et en utilisant cette fonction, vérifiez le type de chacune des variables de ce programme.

► Exercice 5 : Hello You !

Objectifs : utiliser l’IDE Thonny ; écrire et exécuter un court programme en autonomie.

1. Dans votre IDE (Thonny ou un autre), écrivez un programme qui demande à l’utilisateur d’entrer une chaîne de caractères, l’affecte à la variable **prenom** et affiche "Hello " suivi de la valeur de **prenom** suivi de "!".
2. Exécutez votre programme. Qu’affiche-t-il ?

► Exercice 6 : Tableaux de valeurs

Objectifs : affecter une valeur à une variable ; concaténer des chaînes ; utiliser `input` et `print` ; dresser un tableau de valeurs ; se familiariser avec l’utilisation de PythonTutor.

À l’aide de [PythonTutor](#), dessinez les tableaux de valeurs des programmes suivants (vus en TD) :

```
# Programme TD1 exercice 2
demi_mot_1 = "cou"
demi_mot_2 = demi_mot_1
mot = demi_mot_1 + demi_mot_2

mot_utilisateur = input("Veuillez rentrer un mot : ")
mot_utilisateur = " " + mot_utilisateur

phrase = mot + mot_utilisateur
print(phrase)
```

```
# Programme TD1 exercice 3.1
```

```
x = 6
y = x + 3
x = 3
z = 2 + 7 * y
```

```
# Programme TD1 exercice 3.2
```

```
x = 6
y = 7
z = x
z = z + y
```

► Exercice 7 : Suites récurrentes

Objectif : réactiver des notions d’algorithmique et de programmation vues en enseignement de spécialité mathématiques en terminale.

On considère la suite (u_n) définie par $u_0 = 2$ et pour tout rang $n > 0$ par la relation $u_{n+1} = 2u_n - 5$.

La fonction `terme` suivante doit retourner le terme de rang n de la suite (u_n) .

1. Complétez les lignes 2 et 4 afin que la fonction produise le résultat souhaité.

```
def terme(n) :
    u = ...
    for i in range(n) :
        u = ...
    return u
```

```
# Exemple d'appel de la fonction pour tester
terme(5)
```

2. Ecrire une instruction permettant de calculer u_{100} .

► Exercice 8 : Suites récurrentes et seuil

Objectif : réactiver des notions d’algorithmique et de programmation vues en enseignement de spécialité mathématiques en terminale.

La suite (v_n) est définie par $v_0 = 3$ et pour tout rang $n > 1$, $v_{n+1} = 2v_n - 1$.

La fonction `seuil` suivante doit retourner le plus petit rang à partir duquel un terme de la suite (v_n) prend une valeur supérieure à nb .

1. Complétez les lignes 2, 3 et 6 pour que la fonction `seuil` réponde à cet objectif.

```
def seuil(nb) :
    ...
    n = ...
    while v < nb :
        n = n + 1
    ...
    return n
```

2. Quel est le rang k à partir duquel v_k semble dépasser 10^{100} ?

► **Exercice 9 : Next!**

Si vous avez fini, vous pouvez commencer le TP “exo+”.