

# Introduction à Python

Séance pratique #1 – 29 septembre 2023

## 1. Préambule

Résoudre un problème en Python, mais c'est valable quel que soit le langage, c'est :

- Réfléchir et comprendre le problème à résoudre (en français)
- Se documenter sur les spécificités de Python à prendre compte (syntaxe, fonctions natives, etc.)
- Prendre en main l'environnement de programmation
- Implémenter la solution en Python

Le CM que vous avez suivi couvre uniquement le deuxième point. Et jusqu'ici, la documentation est venue à vous. Les séances pratiques de cet enseignement d'introduction à Python auront pour objectif de renforcer vos compétences sur les trois points restants, mais aussi de vous apprendre à chercher dans la documentation.

La démarche ci-dessous est une proposition, et la part de temps que prendra chaque partie variera en fonction des personnes. Mais nul ne peut se passer ne serait-ce que d'une seule.

### a. Réfléchir et comprendre le problème à résoudre

Avant de programmer, réfléchissez. Assurez-vous d'avoir compris le problème en français (ou dans votre langue préférée) avant de chercher à le traduire dans un langage de programmation.

Si vous avez un doute sur *"l'utilité"* de ce que vous êtes en train de faire, dites-vous qu'en conditions réelles, un client n'attendra pas de vous que vous soyez d'accord avec ce pour quoi il vous paie, ni que vous trouviez cela utile. Soyez professionnel et investi dans votre tâche, même si de votre position d'apprenant, elle semble ne vous servir à rien.

### b. Se documenter sur Python

Les 4h de CM ont abordé les clefs de Python. Vous aurez besoin davantage de matière pour travailler. La plupart du temps, taper le nom d'une fonction sur Google suffira à avoir une réponse. Cependant, savoir naviguer dans la documentation officielle Python (<https://docs.python.org>) est indispensable.

Le site <https://www.w3schools.com> propose des tutoriels adaptés aux débutants, ainsi qu'une documentation accessible pour eux.

Le forum <https://stackoverflow.com> est incontournable pour les questions un peu plus tordues.

Enfin, si vous utilisez ChatGPT, utilisez-le à bon escient. Plutôt que de lui demander de faire le travail pour vous, demandez-lui de vous expliquer ce que vous n'avez pas compris.

### c. Prendre en main l'environnement de programmation

Pour programmer, on utilise généralement un IDE (*integrated development environment*), c'est à dire un logiciel "tout-en-un" qui vous permet de taper votre code, de le lancer, d'ajouter ou de supprimer des librairies (packages), et bien plus encore.

Quelques IDE :

- PyCharm (conçu pour les professionnels, orienté vers la productivité)
- Spyder (conçu pour les scientifiques)
- VSCode (conçu par Microsoft)

Cependant, un IDE n'est pas suffisant, il faut aussi installer "le langage", Python. Certains IDE comme PyCharm ou Spyder vous guident dans cette tâche.

Spyder est l'IDE préconisé pour les séances pratiques.

#### d. Implémenter la solution en Python

A vous de jouer !

## 2. Travail à remettre à la fin de la séance

Vous avez deux choses à remettre :

- Un fichier texte (voir explications plus bas)
- Un fichier \*.py contenant votre Python pour chaque exercice

A chaque fois que vous rencontrez un problème et que vous êtes bloqué (erreur tenace, résultat inattendu, difficulté à installer un package, etc.), vous devez en prendre note dans le fichier texte. Si vous réussissez à résoudre le problème, vous indiquerez succinctement la cause et la solution que vous avez appliquée. Sinon, vous porterez la mention « non résolu ».

Exemple :

*Ça me dit "TypeError : can only concatenate str (not "int") to str" alors que mon code est parfait*

*Cause : j'avais des entiers et des string dans mon print*

*Résolution : caster les entiers en string*

  

*Problème 2*

*...*

## 3. Exercice 0

Prenez en main l'éditeur Spyder, ou bien à défaut celui que vous avez choisi. Essayez de créer un projet et de lancer l'un des nombreux exemples vus en CM. Le TPO transmis par mail comporte également de bons exercices pour la prise en main.

## 4. Exercice 1

Ecrire un programme Python qui renvoie le nombre jours dans un mois. L'utilisateur devra saisir le numéro de mois (valeur comprise entre 1 et 12) et l'année. Attention le programme devra tenir compte année des années bissextiles.

## 5. Exercice 2

Ecrire un programme python qui demande à l'utilisateur de saisir un temps en seconde puis qui calcule et affiche ce temps en heures, minutes et secondes (au format HH:mm:ss, ex. 08:30:50). Vous n'utiliserez aucun facilitateur de type convertisseur ou formateur de date.

## 6. Exercice 3

Ecrire un programme python qui calcule l'énergie cinétique d'un objet. L'utilisateur devra saisir la masse et la vitesse de l'objet. La réponse affichée devra préciser la formule utilisée pour le calcul, ainsi que l'unité du résultat.

## 7. Exercice 4

On considère le fichier csv du répertoire TP0/ressources. Chargez ce fichier dans python, à l'aide de Numpy. Ensuite, faites une fonction pour :

- Calculer et afficher les températures maximales
- Calculer et afficher les températures minimales
- Calculer et afficher les températures moyennes
- Afficher une courbe des températures moyennes, par date
- Afficher une courbe des températures maximales et minimales, par date
- Afficher une courbe des températures maximales, minimales et moyennes, par date.

Le calcul des maximums, minimums et moyennes pourra se faire avec les fonctions de Numpy.

Pour chaque courbe, soignez la couleur, le tracé, le titre et le nommage des axes. Essayez plusieurs types de représentations (lignes, barres, moustache, etc.)

## 8. Exercice 5

Envoyez par mail à l'adresse indiquée vos fichiers.

En objet du mail vous mettrez « **[Intro à Python] – Remise TP1** ». Si votre adresse e-mail ne permet pas de vous identifier, vous préciserez vos nom et prénom dans le corps du message.

Vous nommerez le fichier texte « nom\_prenom\_m2physique\_tp1\_log.txt », où nom et prenom sont à remplacer par vos nom et prénom.

Vous nommerez chacun des fichiers d'exercice « **exon\_nom\_prenom\_m2physique\_tp1.py** », où n est le numéro de l'exercice, par exemp9.10.le exo3\_xavier\_alfapac\_m2physique\_tp1.py.