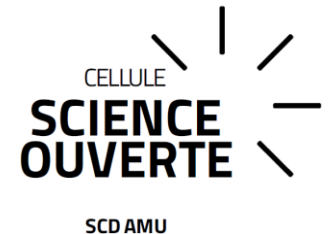




# Initiation à Python

**13/06/2024**

Julien Caugant  
SCD Aix Marseille Université



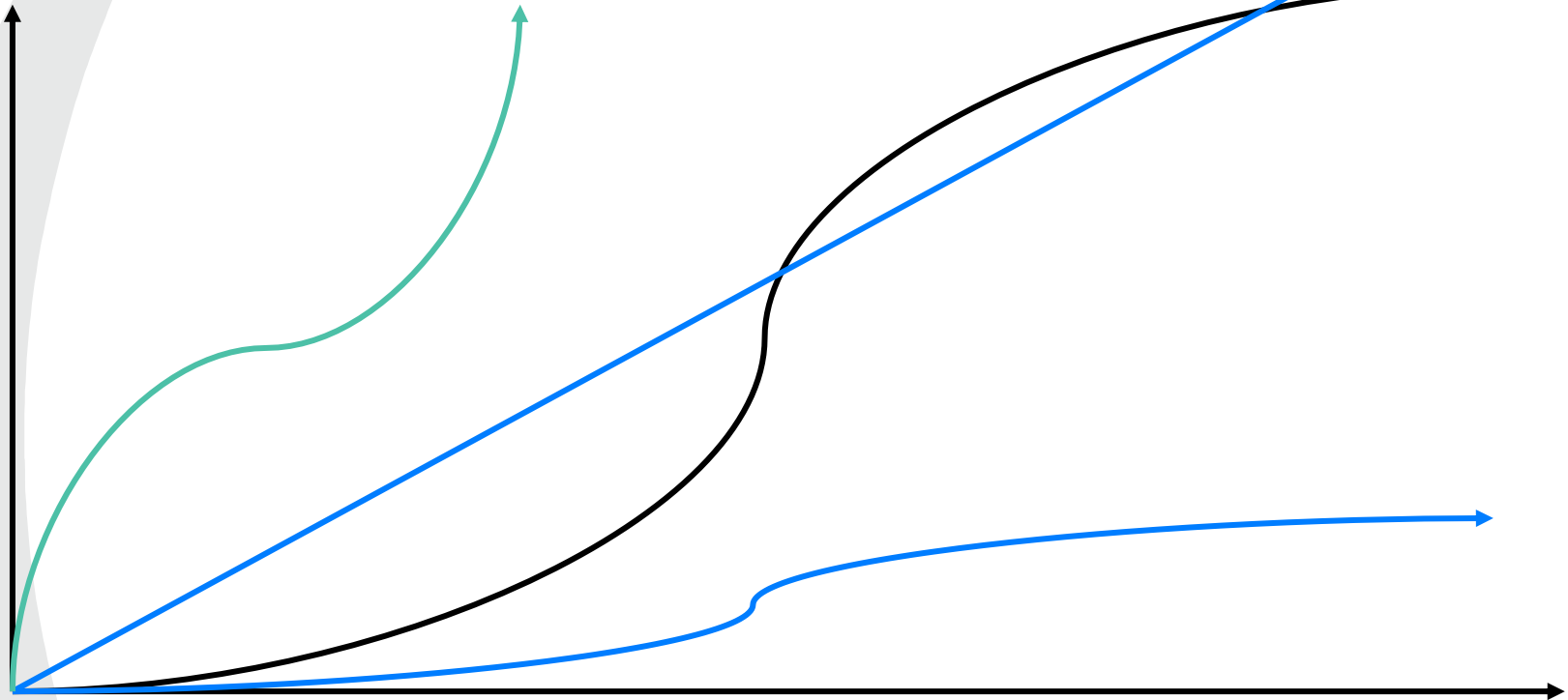


# Python





Temps passé à coder



Quel codeur suis-je?

Nb lignes  
De  
code



Langage de programmation libre interprété, impératif, fonctionnel, orienté-objet avec un typage dynamique fort

Créé à la fin des années 80 par Guido Van Rossum, développeur néerlandais

Très utilisé aujourd'hui dans le domaine des sciences pour ses multiples usages (datascience, automatisation, webscrapping, calcul scientifique, datavisualisation, TAL, robotique)

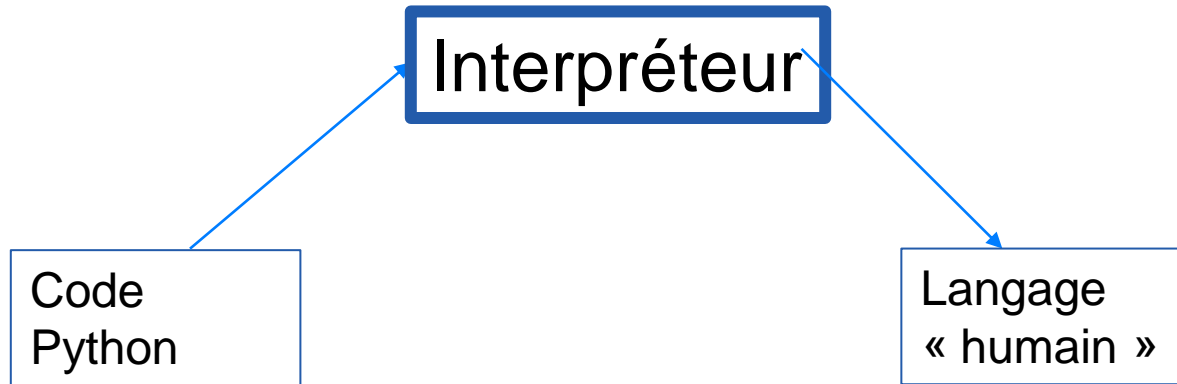
Aucun prérequis nécessaire, convient aussi bien pour les SHS que les ST

Syntaxe relativement simple à utiliser

Traitement numérique des données est aujourd'hui indispensable



Interprétation :



Impératif : suite de fonctions

Ex : a = 4

b = 6

C = a+b

Print(C)

≠ programmation déclarative  
(HTML)



## Quel(s) outil(s) pour programmer?

Anaconda : distribution Python open-source permettant de créer des notebooks en différents langages (Python, R etc...). Peut être déployée sur serveur, permettant ainsi la création de notebooks en ligne

PyCharm : IDE avec de nombreuses fonctionnalités avec une version libre et communautaire, et une version payante pour entreprise

Visual Studio Code : éditeur développé par Microsoft, open-source et gratuit

Notepad++

Google Collab : plateforme de Google permettant de développer en différents langages avec stockage cloud. Accepte les notebooks.

# Python : les bases

Une variable comprend 3 caractéristiques : son nom, son type et sa valeur

Les variables peuvent être des entiers (int), nombres à virgules (float), des chaînes de caractères (string) ou encore des booléens (bool)

Une variable peut contenir un élément unique, ou bien plusieurs éléments, sur lesquels on peut commuter (liste, dictionnaire) ou non (tuple)

Il existe des fonctions prédéfinies en Python (print, len, type...) qui interagissent avec nos variables

Une documentation Python existe : <https://docs.python.org/fr/3/>



## Variables, conditions et boucles

On peut utiliser les conditions pour contrôler le résultat du programme en fonction de nos variables (if, elif) et à l'aide d'opérateurs (<, >, =, != )

On peut également créer des boucles de différentes formes (for, while)

Attention à l'indentation quand on met en place une condition ou une boucle!





## Fonctions et classes

Une fonction est un ensemble d'instructions qui va récupérer des éléments et les manipuler

Les fonctions évitent d'avoir à répéter son code et permettent de l'alléger. Si trop de répétitions de lignes de code, utilisez une fonction

Les classes permettent de simplifier son code et de réutiliser des fonctions attribuées à une classe précédemment déclarée

On crée une fonction avec le mot-clé « def », et une classe avec « class »

Comme pour les boucles, attention à l'indentation!



## Modules et librairies

Un module est un ensemble de classes et fonctions prédéfinies

Une librairie est donc un ensemble de modules

On importe nos librairies et nos modules en premier lieu dans notre code, et si besoin, les classes et fonctions qui y sont liées

Il est conseillé, avant de tester une librairie, de lire sa documentation

# Résumé

**Déclarer un variable :**

**Nom\_de\_la\_variable = « Valeur de la variable »**

**Pour une liste : nom\_de\_la\_liste = ['Valeur1','Valeur2', 3,4, 'Valeur5']**

**Ajouter un élément : nom\_de\_la\_liste.append(nom\_de\_la\_variable)**

**Condition :**

**If variable == valeur :**

**action1**

**action 2**

**Pour les conditions, on peut utiliser les comparateurs '==', '!=', '>', '<', '<=', '>='**

**Les boucles :**

**while variable > valeur :**

**action**

**For i in range(5) :**

**action**

**Fonction :**

**Def nom\_de\_ma\_fonction(variable\_à\_passer) :**

**action(variable\_à\_passer)**

**Librairies :**

**Installation : pip install nom\_de\_la\_librairie**

**Utilisation : import nom\_de\_la\_librairie**