



# Développement logiciel



## Présentation

### Description

Ce cours est axé sur la découverte des bonnes pratiques de codage pour un niveau professionnel.

Le langage utilisé est Python, mais certains éléments de bash et de git seront également utiles.

Un accent particulier sera mis sur le traitement et la visualisation des données au cœur du cours.

Nous nous concentrerons principalement sur les concepts de base de la programmation, ainsi que sur la découverte des bibliothèques scientifiques de Python, dont "numpy, scipy, pandas, matplotlib, seaborn".

Au-delà des connaissances de ces packages fondamentaux, nous introduirons des pratiques modernes pour le code : tests (unitaires), contrôle de version (git), génération de documentation automatique, etc.

### Objectifs

Être capable de créer un package de niveau professionnel en Python incluant, versionnement du code, test unitaire et documentation.

## Pré-requis nécessaires

Les étudiantes et étudiants doivent connaître les notions de base des probabilités, de l'optimisation, de l'algèbre linéaire et des statistiques. Un minimum de connaissance des structures de base de la programmation est aussi demandé (if, then... else, boucle while/for).

Pré-requis recommandés : Un minimum de connaissance en Python serait un plus, ainsi qu'en analyse numérique (flottant, erreurs d'arrondis, etc.).

## Contrôle des connaissances

- CC
- Projet (70 % de la note finale)
- Modalité: en petit groupe, avec une contribution équilibrée de chaque membre.
- Travaux attendus : un dépôt "github" où les travaux sont disponibles, un rapport de synthèse et une soutenance.
- TP noté (15 % de la note finale)
- Quiz (15 % de la note finale)



---

## Syllabus

1. codage : algorithmes, modules, types de base, fonctions, boucles
2. codage : liste, dictionnaire, tuples, if et boucles, exceptions
3. classes (`__init__`, `__call__`, etc...), surcharge des opérateurs, gestion des fichiers, git : une première introduction
4. `numpy` : notions de base sur les matrices (arrays), le découpage, l'algèbre linéaire simple, le masquage ; `matplotlib` : premiers tracés
5. `github`, création d'une clé ssh, diverses commandes git, conflit, pull request
6. `numpy` : casting, concaténation, `imshow`, `meshgrid`, casting, copy, `scipy` : EDO, Interpolation, Optimisation
7. hands on git, intro aux bases de linux et aux outils en ligne de commande
8. `scipy` : Images/canal, FFT, Pandas : données manquantes
9. bash, regexp, grep, find, rename, environnement virtuel Python
10. Pandas : premiers pas
11. Python virtual env : Anaconda, IDE : VScode, Create a Python Module
12. Pandas : en savoir plus
13. Créer un module Python, tests unitaires
14. test unitaire

15. Matrices et graphiques éparses et mémoire

16. Numba

17. Documentation avec Sphinx

18. Statsmodels

---

## Informations complémentaires

Volumes horaires :

CM : 12h

TD : 18h

TP :

Terrain :

---

## Infos pratiques

---

### Contacts

Responsable pédagogique

Xavier Bry

+33 4 67 14 35 78

xavier.bry@umontpellier.fr

Responsable pédagogique

Elodie Brunel-piccinini

+33 4 67 14 41 64

elodie.brunel-piccinini@umontpellier.fr