

PROGRAMMATION AVANCÉE EN PYTHON (PAPY)

2022-2023

| | | | |
|--------------------|--|----------------|------|
| Enseignant: | Lucas Lestandi | Volume: | 32h |
| Email: | lucas.lestandi@ec-nantes.fr | Bureau: | T123 |

Répartition horaire:

1. 8h CM
2. 22h TP
3. 2h DS

Mots-clés: environnement, objets, introspection, modules, architecture, packaging

Objectifs: À la fin de ce cours, les étudiants et étudiantes sauront :

1. Mettre en place un environnement python robuste ainsi qu'un environnement de développement adapté.
2. Écrire des programmes python efficaces utilisant les bibliothèques et abstractions modernes.
3. Assurer la lisibilité et la maintenance en suivant les principes de structuration modulaire et de style cohérents avec celles de la communauté.
4. Créer un paquet pour la distribution sur d'autres machines y compris des dépendances.

Plan du cours:

1. Bien travailler avec python
 - choisir un environnement de travail: IDE, notebooks jupyter, environnement python (conda,...)
 - philosophie et syntaxe de python
 - l'interpréteur python
 - bonnes pratiques en programmation python (PEP8,...)
2. Structuration et types de données
 - Variables, références et gestion de la mémoire
 - Types de données et structures
 - Programmation orientée objet (OOP) : les classes
 - Écrire du code robuste : architecture, introspection, exceptions, etc.
3. Programmer avec des modules
 - les modules natifs : os, sys, subprocess,...
 - les bibliothèques externes : avec pip ou conda
 - quelques exemples communs : numpy, scipy, matplotlib
 - créer ses propres modules

4. Pour aller plus loin

- les décorateurs
- communication avec d'autres langages (C++,...)
- tester et debugger
- distribution and portabilité des paquets

Modalités d'évaluation:

- TPs notés (coefficient 1/2)
- DS (coefficient 1/2)

English

Keywords: environment, OOP, introspection, modules, architecture, packaging

Objectives: At the end of this course, student will have the skills to

1. Set up a robust python environment and a suitable development environment.
2. Write efficient python programs using libraries and modern abstractions.
3. Ensure readability and maintenance by following modular architecture and community style guidelines.
4. Package their code for distribution on other machines including dependencies.

Course outline:

1. The right way to work with python
 - setting up the right environment : IDE, jupyter notebooks, python environments (conda,...)
 - language philosophy, syntax
 - python interpreter
 - good practices for programming in python (PEP8 style guidelines,...)
2. Code structure and data types
 - variables, memory and references
 - data types and structures
 - object oriented programming (OOP) : classes
 - writing robust code : architecture, introspection, exceptions,...
3. Using modules
 - native modules : os, sys, subprocess
 - external libraries : using pip and conda
 - useful examples : numpy, scipy, matplotlib
 - create your own modules
4. Towards production code
 - decorators

- integration with other languages (C++,...)
- testing and debugger
- distribution and portability

Grading Policy:

- Labs report (1/2)
- Final exam (1/2)