

# Introduction à Python pour l'Économie Appliquée

Présentation du cours

Michal W. Urdanivia\*

\*UGA, Faculté d'Économie, GAEL,  
e-mail : [michal.wong-urdanivia@univ-grenoble-alpes.fr](mailto:michal.wong-urdanivia@univ-grenoble-alpes.fr)

28 septembre 2021

## Objectifs

# Python

## 1. Familiarité avec la programmation avec le langage Python.

- (i) Concepts fondamentaux autour desquelles s'organise le langage.
- (ii) Capacité à comprendre un programme écrit en Python.
- (iii) Modules/bibliothèques populaires : numpy, scipy, pandas, statmodels, scikit-learn, matplotlib, ...

## 2. Application en économie

- (i) Analyse numérique de certains modèles classiques de l'économie (Microéconomie, Macroéconomie)
- (ii) Analyse empirique de certaines questions en économie (e.g., offre de travail, déterminants des salaires, ...)

# Économie appliquée

- Pouvoir utiliser les outils du cours dans d'autres projets(cours en économie, économétrie, mémoires, ...)
- Remarque : bien que les applications fassent souvent référence à la théorie économique, ce cours n'est pas un cours de théorie.
- Pouvoir s'orienter dans l'ensemble très riche de références pour développer sa maîtrise du langage Python, et d'autres langages utilisés dans le calcul scientifique(R, Julia, ...)

# Méthodologie

# Apprentissage par la pratique et autoformation

- Vous avez besoin de programmer pour apprendre à programmer :
  - comprendre les programmes/exemples du cours en les exécutant,
  - comprendre les erreurs qui peuvent être rencontrées.
- Beaucoup de concepts/outils utilisés ne sont pas étudiés avec tout le détail qui leur est accordé par d'autres sources/références.
  - Vous devrez(savoir) les consulter dans certains cas où cela vous sera nécessaire.
  - Certains exercices seront repris d'autres sources.

# Matériel

- Notebooks
  - toutes les séances de cours se feront sous la forme de Notebooks Jupyter(voir indications sur Moodle),
  - disponibles ici <https://github.com/MWUrda/Cours-UGA-Python-L3-Eco>.
- Venir en séance avec un ordinateur pour suivre.

# Travail

- Du travail vous sera régulièrement demandé(toutes les deux semaines)
- Il pourra être fait par groupes de deux personnes maximum
- Les correction ne se feront pas en séance mais mises sur le site du cours
- Néanmoins il arrivera régulièrement qu'il soit demandé à certains de les corriger en séance.



## Plan(escompté) et références

# Thèmes

1. Introduction
2. Bases : objets, types, boucles, fonctions
3. Bases : optimizer, affichage, représentation graphique
4. Bases : nombres aléatoires, simulation,
5. Bases : workflow, débogage
6. Bases : résumé, et vue d'ensemble
7. Données : introduction à la bibliothèque pandas
8. Données : applications
9. Algorithmes : introduction
10. Algorithmes : résoudre des équation
11. Algorithmes : optimization numérique
12. Développements/thèmes avancés.

**Remarque** : le plan est susceptible d'être modifié au fur et à mesure de l'avancée du cours.

# Références

- Il existe beaucoup de références pour apprendre à programmer en Python,
- Aucune ne vous donnera une connaissance complète du langage
- Elles se différencient souvent selon les publics et spécialités visées (sciences de la donnée, économie, informatique, ...)
- Le cours s'appuie sur certaines d'entre elles qui vous seront indiquées dans les séances et en fonction des thèmes pour lesquelles elles sont mobilisées.

# Références

- Quelques unes qui reviendront souvent :
  1. QuantEcon : <https://quantecon.org/>
  2. Documentation spécifique de Python : <https://docs.python.org/3/library/>
  3. Les sites de documentation des principales bibliothèques(numpy, scipy, pandas, ...)
  4. Python course : [https://www.python-course.eu/python3\\_course.php](https://www.python-course.eu/python3_course.php)