

# Programmation

## Titulaire

Thierry MASSART (Coordonnateur)

## Mnémonique du cours

INFO-F101

## Crédits ECTS

10 crédits

## Langue(s) d'enseignement

Français

## Période du cours

Premier quadrimestre

## Campus

Plaine

## Contenu du cours

- 1 Initiation à la programmation de base avec le langage *Python* :
  - Notions de valeur, variable, expression, instruction, fonction, exception, classes et objet
- 2 Analyse du fonctionnement et de propriétés de codes *Python* :
  - Notion de diagramme d'état
  - Mécanisme de fonctionnement de codes *Python*
  - Évaluation de l'efficacité d'un programme (complexité et notion de grand O)
  - Formalisation de l'état d'un programme sous forme logique
- 3 Algorithmique de base
  - Algorithmes de recherche et de tri
  - Algorithmes récursifs
- 4 Manipulation simple de structures de données :
  - Vecteur et tableaux à plusieurs dimensions, listes, dictionnaires, ensembles, fichiers, arbres, graphes, ...
- 5 Résolution de problèmes via des programmes *Python*
  - Petits problèmes mathématiques
  - Liés à des structures de données
  - Jeux,
  - ...
- 6 Introduction aux langages de programmation
  - Notions de syntaxe, sémantique, typage, interpréteur, compilateur, runtime, paradigmes
  - Bref historique

## Objectifs (et/ou acquis d'apprentissages spécifiques)

À l'issue de l'enseignement l'étudiant doit être capable de :

- 1 démontrer une bonne compréhension des concepts de base de *Python* ainsi que lire, comprendre, expliquer le fonctionnement et analyser l'efficacité de programmes existants
- 2 analyser un problème simple et proposer une solution informatique pour le résoudre et la mettre en œuvre en *Python*
- 3 réaliser des programmes *Python* corrects et bien structurés
- 4 identifier les cas de test pour valider ces programmes
- 5 exprimer formellement, en formalisme logique, les fonctionnalités attendues d'un programme informatique
- 6 utiliser des outils informatiques de support à la programmation; exploiter la documentation technique
- 7 décrire les notions de bases liées au langage (syntaxe, sémantique, type, interpréteur, compilateur, runtime, paradigmes, ...)

Pour les six premiers objectifs de la liste précédente, il est demandé un apprentissage en profondeur, en étant en particulier capable de concevoir et d'analyser des programmes qui résolvent des problèmes originaux. Le dernier objectif requiert une compréhension et une capacité d'expliquer les concepts.

## Pré-requis et co-requis

### Cours ayant celui-ci comme pré-requis

INFO-F202 | Langages de programmation 2 | 5 crédits, INFO-F203 | Algorithmique 2 | 5 crédits, INFO-F204 | Analyse et méthodes | 5 crédits, INFO-F205 | Calcul formel et numérique | 5 crédits et INFO-H303 | Bases de données | 5 crédits

### Cours ayant celui-ci comme co-requis

INFO-F103 | Algorithmique 1 | 10 crédits et INFO-F106 | Projets d'informatique 1 | 5 crédits

## Méthodes d'enseignement et activités d'apprentissages

L'apprentissage de la programmation Python de base, qui constitue la première partie des quatre premiers objectifs, se fait par apprentissage *hybride*. L'élève doit s'inscrire et réaliser le MOOC (Massive Open Online Course) « *Apprendre à coder avec Python* » sur la plateforme FUN-MOOC. Ce MOOC est un enseignement complet en ligne d'initiation à la programmation *Python*. Chaque matière est abordée par l'exemple, formalisée, son application est illustrée et ensuite l'écriture individuelle de codes *Python* résolvant de petits problèmes, et auto-évalués par notre environnement, est demandée. Un suivi de l'apprentissage de chaque étudiant est réalisé par prise de rapports d'avancement périodiques, demandes de réalisation de projets de programmation, ainsi qu'au cours ex-cathedra, séances de séminaires et de travaux pratiques.

L'apprentissage plus en profondeur des premiers objectifs ainsi que des autres objectifs se fait lors des séances de cours ex-cathedra, des séminaires et travaux pratiques.

L'accent est mis sur le travail autonome .

## Contribution au profil d'enseignement

Si vous avez suivi avec fruit le cours, vous aurez amélioré les compétences suivantes :

- > L'apprentissage autonome : vous serez dans une dynamique d'apprentissage autonome et permanent dans ce domaine en constante et rapide évolution qu'est l'informatique ; vous pouvez vous adapter tout au long de votre carrière aux technologies nouvelles. Par exemple, vous apprendrez, en vous référant aux manuels.
- > La résolution de problèmes : vous aurez la capacité d'analyser des besoins, de structurer l'information, de concevoir, modéliser et implémenter des solutions pertinentes et efficaces ; de façon plus globale on vous demandera dans les cours d'informatique d'acquiescer la "pensée informatique" ("computational thinking") en étant capable de faire des abstractions adéquates pour un problème, et d'allier la théorie à la pratique avec l'ordinateur comme support;
- > La communication : vous pourrez comprendre les problèmes posés, et expliquer les solutions proposées ; vous pourrez utiliser la communication scientifique et technique (formalismes mathématiques).

## Références, bibliographie et lectures recommandées

- > MOOC "Apprendre à coder avec Python" sur fun-mooc.fr
- > Programmation - INFO-F-101, T. Massart, Presses Universitaires de Bruxelles, 2022-2023 (2 syllabi du cours)
- > Livre Apprendre à programmer avec Python 3, de Gérard Swinnen [<http://inforef.be/swi/python.htm>] (vendu aux PUB)

## Support(s) de cours

Université virtuelle et Syllabus

## Autres renseignements

### Lieu(x) d'enseignement

Plaine

### Contact(s)

M. Thierry MASSART (Campus Plaine, bâtiment NO, 2N8 113) [thierry.massart@ulb.be](mailto:thierry.massart@ulb.be)

## Méthode(s) d'évaluation

Examen écrit, Projet et Travail personnel

## Méthode(s) d'évaluation (complément)

Examen écrit avec questions ouvertes théoriques et pratiques (programmes à réaliser). Note d'année sur les projets personnels et le suivi au MOOC "Apprendre à coder avec Python"

## Construction de la note (en ce compris, la pondération des notes partielles)

- > Projets\_et\_Annee = Projet(s) d'année + rapports d'avancement au MOOC
- > Test\_Novembre = test fin octobre ou début novembre sur la matière déjà vue (0/20 si absent)
- > Examen\_Janvier = Examen de janvier
- > Examen\_Mai = Examen de mai / juin
- > Examen\_Aout = Examen d'août / septembre
- > Janvier =  $(4/20 \cdot Projets\_et\_Annee + 16/20 \cdot Interro\_Janvier)$
- > **Final\_Janvier =**
  - > si Test\_Novembre > Janvier :  $(5/20 \cdot Test\_Novembre + 15/20 \cdot Janvier)$
  - > sinon : (Janvier)
- > **Final\_Mai =**
  - > si l'étudiant est présent:  $(4/20 \cdot Projets\_et\_Annee + 16/20 \cdot Examen\_Mai)$
  - > sinon : (Final\_Janvier)
- > **Final\_Aout =**
  - > si l'étudiant est présent:  $4/20 \cdot Projets\_et\_Annee + 16/20 \cdot Examen\_Aout$
  - > sinon : il est mis Absent
- > Attention, la note d'année et les projets comptent pour toutes les sessions et aucun rattrapage n'est possible !!!!

## Langue(s) d'évaluation principale(s)

Français

## Programmes

### Programmes proposant ce cours à la faculté des Sciences

BA-INFO | **Bachelier en sciences informatiques** | bloc 1, BA-MATH | **Bachelier en sciences mathématiques** | bloc 1, MA-BINF | **Master en bioinformatique et modélisation** | finalité Approfondie/bloc 1 et MA-GEOG | **Master en sciences géographiques, orientation générale** | finalité Développement territorial/bloc 2