

CS-119(h)

**Information, calcul, communication**

Lévêque Olivier, Stojilovic Mirjana

Cursus	Sem.	Type
Génie civil	BA1	Obl.
Science et génie des matériaux	BA1	Obl.

Langue d'enseignement	français
Coefficient	6
Session	Hiver
Semestre	Automne
Examen	Pendant le semestre
Charge	180h
Semaines	14
<b>Heures</b>	<b>6 hebdo</b>
Cours	3 hebdo
Exercices	3 hebdo
<b>Nombre de places</b>	

**Résumé**

L'objectif de ce cours est d'initier les étudiants à la pensée algorithmique, de les familiariser avec les fondamentaux de l'informatique et des communications et de développer une première compétence en programmation avec le langage Python.

**Contenu**

La partie théorique est organisée en trois modules : calcul (algorithmes, récursion, complexité, représentation des nombres), information (échantillonnage, reconstruction, th. de Nyquist-Shannon, compression, 1er th. de Shannon), systèmes et sécurité (ordinateur de von Neumann, hiérarchies de mémoire, réseaux, menaces et défenses, cryptographie à clé secrète, RSA).

La partie pratique présente une introduction à la programmation impérative et à ses concepts fondamentaux avec le langage Python. Elle aborde les notions suivantes: variables, expressions, structures de contrôle (conditions, boucles), fonctions (déclaration, appel, arguments), modules, entrées-sorties, structures de données (*list*, *set*, *tuple*, *dictionary*), interfaces graphiques simples et utilisation de bibliothèques (*numpy*, *matplotlib*) et d'environnement différents (IDE vs. Jupyter notebooks).

**Mots-clés**

Informatique, Ordinateurs, Algorithmes, Communication, Programmation

**Acquis de formation**

A la fin de ce cours l'étudiant doit être capable de:

- Analyser un problème complexe pour le décomposer en sous-problèmes
- Concevoir et exprimer un algorithme
- Modéliser en langage Python une situation simple du monde réel décrite en français
- Transcrire un algorithme en son programme équivalent en Python
- Calculer la complexité d'un algorithme simple
- Expliquer comment représenter des nombres et des symboles dans un ordinateur
- Exposer comment mesurer la quantité d'information présente dans des données
- Réaliser de façon autonome une application de petite taille au moyen du langage Python
- Analyser du code Python simple pour en décrire le résultat ou le corriger s'il est erroné
- Tester l'adéquation du résultat d'un programme simple par rapport à la tâche demandée
- Exposer comment capter la réalité physique avec des (suites de) nombres
- Exposer comment reconstruire les grandeurs physiques à partir de suites de nombres les représentant
- Exposer quel problème peut être résolu avec un algorithme

- Utiliser la ligne de commande Linux pour exécuter un programme
- Structurer un programme
- Comparer plusieurs approches (bouts de code)

### Compétences transversales

- Utiliser les outils informatiques courants ainsi que ceux spécifiques à leur discipline.
- Accéder aux sources d'informations appropriées et les évaluer.

### Méthode d'enseignement

Partie théorique: séances ex cathedra avec exercices

Partie pratique (programmation): séances ex cathedra, classe inversée, travaux pratiques sur ordinateur

### Travail attendu

Participation au cours (en salle), résolutions d'exercices, travail personnel à la maison

### Méthode d'évaluation

Contrôle continu: deux examens intermédiaires pendant le semestre (valant pour respectivement 33.25% et 47.5% de la note finale), un projet de programmation (valant pour 14.25% de la note finale) et un quiz sur la sécurité informatique (valant pour 5% de la note finale)

### Encadrement

Office hours	Non
Assistants	Oui
Forum électronique	Oui

### Ressources

#### Service de cours virtuels (VDI)

Oui

#### Bibliographie

- Découvrir le numérique, André Schiper et al., EPFL Press, 2016
- Python Crash Course: A Hands-On, Project-Based Introduction to Programming, Eric Matthes, No Starch Press, 3rd edition
- Automate the Boring Stuff with Python, 2nd Edition: Practical Programming for Total Beginners, Al Sweigart, No Starch Press
- The Big Book of Small Python Projects, Al Sweigart, No Starch Press

#### Ressources en bibliothèque

- [Découvrir le numérique / Schiper](#)
- [The Big Book of Small Python Projects / Sweigart](#)
- [Automate the Boring Stuff with Python / Sweigart](#)
- [Python Crash Course / Matthes](#)

#### Liens Moodle

- [https://go.epfl.ch/CS-119\\_h](https://go.epfl.ch/CS-119_h)

#### Vidéos

- <https://courseware.epfl.ch/courses/course-v1:EPFL+PENSEE-INFORMATIQUE+2020/course/>