

# Algorithmique et programmation fonctionnelle 2



## En bref

- > **Langues d'enseignement:** Français
- > **Méthodes d'enseignement:** Hybride
- > **Ouvert aux étudiants en échange:** Oui

## Présentation

---

### Description

Cet enseignement est une introduction à l'algorithmique et à la programmation logicielle.

Il présente les fondements de l'algorithmique et montre comment ils peuvent être utilisés dans une démarche de résolution de problème. Il repart des notions de fonction mathématique, variable, ensemble et éléments appartenant à des ensembles. Cet enseignement présente les bases de la programmation fonctionnelle au travers des langages *Racket* et *Scheme*, inspirés du langage LISP. Cet enseignement fournit les bases nécessaires pour construire des algorithmes récursifs et itératifs simples au cours desquels les principales structures de contrôle seront abordées.

---

### Objectifs

L'objectif principal de cet enseignement est d'une part l'acquisition des principes guidant la compréhension et l'écriture d'algorithmes dans une démarche de résolution de problèmes, d'autre part la découverte de la programmation fonctionnelle comme moyen simple d'expression, de formalisation d'algorithmes. Ce travail permettra l'apprentissage d'un langage de programmation fonctionnelle.

---

## Heures d'enseignement

|    |                 |       |
|----|-----------------|-------|
| CM | Cours Magistral | 13,5h |
| TD | Travaux Dirigés | 12h   |

---

## Pré-requis obligatoires

Notions de bases en mathématiques

---

## Plan du cours

- \* Introduction à l'algorithmique et à la notion de calculabilité
  - \* Introduction à la programmation fonctionnelle avec les langages *Racket*, *Scheme*
  - \* Principes de conception de programmes
  - \* Ecriture d'algorithmes en utilisant les langages *Racket*, *Scheme*
  - \* Algorithmique et récursivité
- 

## Compétences visées

- \* Etre capable de mettre en place une démarche de résolution de problème
  - \* Savoir lire et comprendre des algorithmes simples
  - \* Etre en capacité d'écrire des algorithmes simples
  - \* Comprendre différentes formes algorithmes
  - \* Etre en capacité d'utiliser les principales structures de contrôle
  - \* Savoir utiliser de façon simple la programmation fonctionnelle avec les langages *Racket*, *Scheme*
- 

## Bibliographie

- \* T. Cormen, C. Leiserson, R. Rivest, Introduction à l'algorithmique, Dunod
- \* L. Arditi, S. Ducasse, La Programmation une approche fonctionnelle et récursive avec Scheme, Eyrolles
- \* J. Charazin, Programmer avec Scheme, Thomson Publishing
- \* R. K. Dybvig, The Scheme Programming Language, The MIT Press
- \* H. Abelson, J. Sussman, Structure and Interpretation of Computer Programs, The MIT Press
- \* J.P. Roy, Premiers cours de programmation avec Scheme – Du fonctionnel pur aux objets avec DrRacket, Ellipses

## Infos pratiques

## Contacts

Responsable du cours

Herve Verjus

☎ +33 4 50 09 65 94

✉ Herve.Verjus@univ-savoie.fr

---

## Lieux

➤ Anancy-le-Vieux (74)

---

## Campus

➤ Anancy / campus d'Anancy-le-Vieux