

Concepts et Paradigmes de Programmation par la Pratique

C3P

Vincent Aranega
vincent.aranega@univ-lille.fr

Université de Lille

*“A language that doesn't affect the way you think about
programming, is not worth knowing”*

– Alan Perlis

Table of contents

Plan des cours

Plan des TPs

Ressources

Une autre pour la route

“Every programmer should acquire a deep understanding of functional programming.

Every programmer should acquire a deep understanding of object-oriented programming.

Every programmer should acquire a deep understanding of structured programming.”

– Uncle Bob (Robert Martin)

Objectifs

Aborder et comprendre trois des principaux paradigmes de programmation ainsi que les outils de développement associés.

- ▶ Comprendre les différences sémantiques entre langages
- ▶ Comprendre les points clés de la programmation procédurale, orienté-objet et fonctionnelle
- ▶ Produire des architectures et solutions dédiées à un paradigme
- ▶ Mélanger les paradigmes lorsque c'est possible
- ▶ Connaître les différents points clés nécessaire à l'apprentissage du développement dans un langage donné

Organisation

Intervenants

- ▶ Vincent Aranega (`vincent.aranega@univ-lille.fr`)
- ▶ Clément Ballabriga (`clement.ballabriga@univ-lille.fr`)

Découpage des cours

- ▶ 1 séance de cours magistral
- ▶ 2 séances de Travaux Pratique

Évaluation

- ▶ Note examen sur table (probablement un QCM)
- ▶ Note de Travaux Pratique
- ▶ Note sur rendu/recette pour une suite de TP en particuliers

Table of Contents

Plan des cours

Plan des TPs

Ressources

Thématiques

1. Aborder le développement dans un nouveau langage
2. Fonctionnel avec OCaml
3. Procédural avec C
4. Hybride avec Python (OO, Procédural, Fonctionnel) et métaprogrammation
5. Versioning the fichiers avec Git

Table of Contents

Plan des cours

Plan des TPs

Ressources

Plan des TPs

Thématiques

1. TPs OCaml (prise en main, programmes avancés)
2. TPs en C
3. TPs en Python
4. “Projet” et rendu

Attention

Utilisation de git systématique (fait partie de la notation)

Dernières séances

Les dernières séances seront dédiées à l'apprentissage d'un nouveau langage et à la réalisation d'un mini-projet parmi 3 propositions. À la fin, par binômes, présentation du projet réalisé, de l'architecture de la solution et des spécificités du langage.

Langages

Haskell, Typescript, Kotlin, Rust, Elm, Pharo, Lua, Go (si vous en avez un autre en tête, venez me le dire)

Table of Contents

Plan des cours

Plan des TPs

Ressources

Ressources

Langages

- ▶ OCaml
 - ▶ Documentation (<https://ocaml.org/docs/index.fr.html>)
- ▶ C
 - ▶ Abordés en L2 (PDC) et L3 (PDS)
- ▶ Python 3
 - ▶ Abordé en L1 et L2
 - ▶ Orienté objet (<https://docs.python.org/3/tutorial/classes.html>)

Outils

- ▶ git
 - ▶ Documentation (<https://git-scm.com/docs>)
 - ▶ Cours (<https://github.com/rudametw/Learning-Git/raw/master/PDFs/slides.pdf>)

Frameworks

Langages

- ▶ OCaml
 - ▶ Test unitaire avec OUnit
(<http://ounit.forge.ocamlcore.org/>)
- ▶ C
 - ▶ Test unitaires avec Check
(<https://libcheck.github.io/check/>)
- ▶ Python
 - ▶ Pipenv
(<https://pipenv-fork.readthedocs.io>)
 - ▶ Pytest
(<https://github.com/ralscha/wamp2spring>)