

# PFITAXEL – exercices en ligne pour PFITA

## Bureau d'étude L3 Info

Érik Martin-Dorel ([erik.martin-dorel@irit.fr](mailto:erik.martin-dorel@irit.fr))  
& Frédéric Migeon ([frederic.migeon@irit.fr](mailto:frederic.migeon@irit.fr))

UPS/IRIT – Février 2018

### 1 Nature du travail : développement logiciel/web

### 2 Description du contexte du sujet

Depuis l'année 2016-2017, l'UE PFITA (Programmation Fonctionnelle et Introduction aux Types Abstraits) de la Licence 3 Informatique de l'UPS utilise un outil d'évaluation automatique à base de tests fonctionnels et d'analyse statique. Cet outil génère un rapport HTML pouvant être envoyé aux étudiants pour leur indiquer leurs erreurs ou les motifs de code à éviter.

Cet outil est composé d'un script, d'une version modifiée du moteur de MOOC `learn-ocaml`<sup>1,2</sup> développé par OCamlPro, et enfin de codes OCaml dédiés pour corriger chaque exercice.

Le but de ce projet est de développer des évolutions de l'outil

- pour faciliter le travail des enseignants
- et pour pouvoir généraliser l'utilisation de cette plateforme en TP PFITA pour accroître l'autonomie et l'engagement des étudiants.

### 3 Description du travail demandé

1. Prise en main de l'environnement technologique et identification de motifs de code typiques dans les scénarios de tests visant à évaluer automatiquement les exercices OCaml.
2. Développement de fonctionnalités pour faciliter la création de nouveaux sujets (éditeur intégré dans l'application en ligne `learn-ocaml` ET/OU ensemble de macros pour l'éditeur `emacs/tuareg-mode`). Il pourrait s'agir d'un éditeur de questions simplifié permettant de générer semi-automatiquement le code OCaml de l'évaluateur.
3. Expérimentation et utilisation de cette fonctionnalité en mettant en place des scénarios de tests automatisés pour la plupart des exercices existants dans le module PFITA.
4. Développement de macros pour `emacs` pour tirer profit en TP à la fois des fonctionnalités d'édition d'`emacs/tuareg-mode`/`merlin`, et du feedback fourni via la plateforme `learn-ocaml`.

Des revues pourront être organisées avec les différentes parties prenantes.

---

1. <https://try.ocamlpro.com/learn-ocaml-demo/>  
2. <https://github.com/ocaml-sf/learn-ocaml>

## 4 Environnement technologique

Environnement de développement : [Git](#), [Emacs](#), [Tuareg](#), [Merlin](#)

Langages : [OCaml](#), [Emacs Lisp](#)

Framework : [js\\_of\\_ocaml](#)

## 5 Taille de l'équipe recommandée

4 étudiants (au moins)