

Overview of the PFITAXEL project Using the Learn-OCaml platform

Érik Martin-Dorel

Dept. *Fiabilité des Systèmes et des Logiciels*, Lab. IRIT
Univ. Toulouse III - Paul Sabatier

Friday 20 Mai 2022

Agenda

- 1 Introduction
- 2 Overview of Learn-OCaml's features and architecture
- 3 Upcoming features

Agenda

- 1 Introduction
- 2 Overview of Learn-OCaml's features and architecture
- 3 Upcoming features

Meaning of the PFITAXEL acronym

*Programmation Fonctionnelle et Introduction aux Types Abstraits :
Exercices en Ligne.*

Meaning of the PFITAXEL acronym

*Programmation Fonctionnelle et Introduction aux Types Abstraits :
Exercices en Ligne.*

Démo temporaire (sur une VM louée ce mois-ci) :

<https://pfitaxel-dev.martin-dorel.org>

Secret : DAS

Contexte et historique du projet

- Enseignements de programmation fonctionnelle en L3 Informatique (OCaml)
- Étudiants concernés par semestre : ~180 étudiants
- 2015-2016 : caml-autograder (Perl+OUnit) : éval. DM / tests unitaires
- 2016-2017 : fork de **learn-ocaml** (OCamlPro) : éval. DM / **tests + expressifs**
- 2017-2018 : **learn-ocaml-editor** (5 stages L3) : conception des sujets facilitée
- 2018-2019 : transformation des **TP de l'UE PFITA**, pas seulement les nouveaux DM, puis déploiement d'un serveur **pfitaxel** (location d'une VM pour y déployer l'application web **learn-ocaml**)
Premier questionnaire d'utilisabilité.
- 2019-2020 : amélioration de **learn-ocaml-editor** (export Zip et question templates), conception et développement de **learn-ocaml.el**, support du déploiement statique pour **learn-ocaml**, conception et développement d'une authentification **Moodle/LTI/SMTP pour learn-ocaml**, transformation des **TP de l'UE TAPFA** (3 feuilles de TP OCaml)
- ...

Wrap-up of Learn-OCaml's history

- Originated in the OCaml MOOC (2015 then 2017, 2018, 2019, 2020)
 - Course contents created by Univ. Paris Diderot (Roberto Di Cosmo, Yann Régis-Gianas, Ralf Treinen...)
 - Platform developed by OCamlPro (Benjamin Canou, Çağdaş Bozman, Grégoire Henry, Louis Gesbert, Pierrick Couderc...)
 - Based on openEDX (running everything from the browser)
- 2016: OCamlPro made Learn-OCaml indep./openEDX, AGPL license
- 2018: © transfer to the OCaml Software Foundation, MIT license
- As of 2022-05-20: 280 PRs integrated by Erik MD/Yann Régis-Gianas from 30 contributors
- Used at least in France, Portugal, England, Canada, USA:
<https://github.com/ocaml-sf/learn-ocaml-places>

Questionnaire d'utilisabilité

Premier sondage en 2019 : 56 étudiants y avaient répondu. 3 parties :

- 1 Questionnaire d'utilisabilité standard en 10 questions, obtenant un score de 77. En pratique, un score > 68 correspond à un système *plus utilisable que la moyenne*.
- 2 Question plus ciblée : «Je trouve que cet environnement de TP est utile pour s'initier à un langage de programmation tel qu'OCaml ?» (échelle de 1 à 5) : score de 86 %.
- 3 Points positifs et suggestions d'améliorations :
 - + portabilité
 - + feedback (erreurs de compilation, tests automatisés)
 - + animations, ergonomie de l'application web
 - - lenteur sur certains PCs (OCaml étant émulé en JavaScript)

Agenda

- 1 Introduction
- 2 Overview of Learn-OCaml's features and architecture
- 3 Upcoming features

Learn-OCaml home page: summary of its main features

The screenshot shows the Learn-OCaml home page interface. The top bar displays the user's connection status as 'Teacher'. Below this are several utility buttons: 'Show token', 'Sync workspace', 'Export to file', 'Import', 'Download all source files', and 'Logout'. The 'Activities' section lists several options: 'Tutorials', 'Lessons', 'Exercises', 'Toplevel', 'Playground', and 'Teach'. Red lines connect these interface elements to explanatory text on the right.

Connected as **Teacher** — Nickname (⇒ BACKEND): optional name to help identifying the user. Can be for example: FirstLast@Group1.

Show token — Token (⇒ BACKEND): unique string (e.g. GX9-HBS-1KS-A1J) serving as login+password (can't be changed!)

Download all source files — Download a .zip archive (⇒ BACKEND) with all submitted exercises (i.e., with the Grade button, not just Sync)

Choose an activity.

Tutorials — "Interactive tutorials": sequence of several topics with snippets that can be automatically copied to a toplevel.

Lessons — Lectures: sequence of web slides with -- → arrows and ocaml syntax highlighting.

Exercises — (The Main Feature™) Exercises with description, prelude, template, editor, toplevel, and custom graders.

Toplevel — Client-side toplevel (less useful than Playgrounds!)

Playground — Playgrounds: editor, toplevel, and customizable preludes.

Teach — Teacher Dashboard (⇒ BACKEND): track the progress of students on the exercises (if they clicked on Grade)

Learn-OCaml exercises repositories: the "exodir" spec. I

```
the-repository
├── exercises
│   ├── index.json    → ordered list of exos
│   ├── tpl
│   │   ├── descr.md → questions
│   │   ├── meta.json → meta-data
│   │   ├── prelude.ml → given code
│   │   ├── prepare.ml → hidden given code
│   │   ├── solution.ml → complete solution
│   │   ├── template.ml → starter code
│   │   └── test.ml → ocaml grader
│   ├── tp2
│   └── ...
├── lessons
│   └── ...
├── playgrounds
│   └── ...
├── tutorials
│   └── ...
└── $ learn-ocaml build serve --repo=the-repository
```

Learn-OCaml exercises repositories: the "exodir" spec. II

```
the-repository/exercises/index.json:
{
  "learnocaml_version": "1",
  "groups":
  { "group1":
    {
      "title": "Some group of exercises",
      "exercises": [ "tp1", "tp2" ]
    }
  }
}
```

cf. https://ocaml-sf.org/learn-ocaml/exercises_format.html

Summary of the architecture of Learn-OCaml

- opam dependencies: ocaml, lwt, and js_of_ocaml
- **client-side** grading (an "exploitation" is still possible currently)
- **static** deployment:
 - a running backend is optional in learn-ocaml !
 - <https://github.com/ocaml-sf/learn-ocaml-public> (GH Pages)
- full-stack deployment (with a running backend):
 - using docker/docker-compose and an [ocaml-sf/learn-ocaml](#) image
 - or using the [learn-ocaml-essok](#) project
 - or using the [learn-ocaml](#) binaries (handy for local tests)
 - **provides** Student accounts and Teacher accounts (with a Dashboard)

learn-ocaml.el *a.k.a.* learn-ocaml-mode (1/2)

- by Manuel Cabarcos-Baulina, Louis Ayroles, Erik MD
- Separated software project:
<https://github.com/pfitaxel/learn-ocaml.el> (MIT license)
- Emacs/Tuareg front-end that is integrated **in the MELPA distribution**:
<https://melpa.org/#/learn-ocaml>
- Use `learn-ocaml-client`
- Advantages:
 - a standard OCaml **IDE UX** (using **Merlin+Eldoc**) for coding exercises
 - the students store their `.ml` files directly on their workstation
 - **grading** w.r.t. a Learn-OCaml server (TOKEN or mail/passwd)
 - **very quick feedback** (no JS is involved anymore)
- Bottlenecks:
 - Pending PR: [#458](#) (lifts a limitation, paves the way to `.cmo` JS grader)
 - Needs a few small additions to `learn-ocaml-client`

learn-ocaml.el *a.k.a.* learn-ocaml-mode (2/2)

learn-ocaml.el : dashboard avec la liste des exercices

```
emacs@presto
File Edit Options Buffers Tools LearnOCaml Exercise-Id: Help
[Search] [Save] [Undo] [Redo] [Refresh]
: Refresh List | TAB / S-TAB : Navigate | q : Close List
-----
LearnOCaml [directory] (-/) [Change & refresh]

Démo de l'environnement learn-ocaml

Démo de l'environnement learn-ocaml
Commencez par consulter cet exercice
Difficulty: 0/4 id: demo
[Browse subject] [Get template] [Open .ml] [Get last saved version]

PFITA/TP1 : Types et expressions, filtrage et n-uplets

PFITA - Thème 5
Filtrage par motifs
Difficulty: 2/4 id: pfita-theme5
[Browse subject] [Get template] [Open .ml] [Get last saved version]

PFITA - Thème 6 - TPne
Révisions CTD 1-3
Difficulty: 2/4 id: pfita-theme6
[Browse subject] [Get template] [Open .ml] [Get last saved version]

PFITA/TP2 : Filtrage et récursion sur les entiers

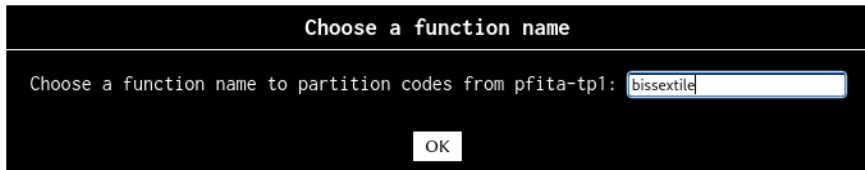
PFITA - Thème 7
Récursion sur les entiers
Difficulty: 2.5/4 id: pfita-theme7
[Browse subject] [Get template] [Open .ml] [Get last saved version]

PFITA/TP3 : Listes et récursion sur les listes

U:%*- *learn-ocaml-exercise-list* Top (1,0) (Fundamental LearnOCaml)
```

Dissimilarity analysis (by Alexandre Moine & Yann R.-G.) I

- so-called Learn-OCaml's **partition-view** feature
Teacher Dashboard → **middle-click** on an exercise (e.g. pfita-tp1)
- example input:



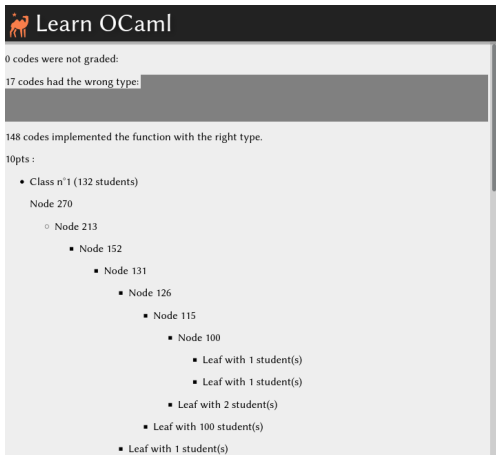
Choose a function name

Choose a function name to partition codes from pfita-tp1:

OK

Dissimilarity analysis (by Alexandre Moine & Yann R.-G.) II

- example output:



Learn OCaml

0 codes were not graded:

17 codes had the wrong type:

148 codes implemented the function with the right type.

10pts :

- Class n°1 (132 students)
 - Node 270
 - Node 213
 - Node 152
 - Node 131
 - Node 126
 - Node 115
 - Node 100
 - Leaf with 1 student(s)
 - Leaf with 1 student(s)
 - Leaf with 2 student(s)
 - Leaf with 100 student(s)
 - Leaf with 1 student(s)

Static binaries as release assets (by Louis Gesbert & EMD)

- For each release, static binaries are available for {Linux, macOS}:
<https://github.com/ocaml-sf/learn-ocaml/releases>

▼ Assets 9

 learn-ocaml-client-darwin-x86_64
 learn-ocaml-client-linux-x86_64
 learn-ocaml-darwin-x86_64
 learn-ocaml-linux-x86_64
 learn-ocaml-server-darwin-x86_64
 learn-ocaml-server-linux-x86_64
 learn-ocaml-www.zip

- `learn-ocaml-client`: CLI grader for **Tuareg+Merlin**, faster than JS
- `learn-ocaml`: CLI teacher tool to statically build and serve an instance
- `learn-ocaml-server`: server binary, faster than `learn-ocaml serve`

Agenda

- 1 Introduction
- 2 Overview of Learn-OCaml's features and architecture
- 3 Upcoming features

use_{passwd}/use_{moodle} (by IRIT interns & Erik MD)

- Strong authentication (by e-mail/password, that can be changed)
 - Add `"use_passwd":true` in `server_config.json`
 - SMTP protocol: send automatic mails for e-mail/password change
 - support in Tuareg/Merlin/learn-ocaml-mode : OK
- Moodle auth (Moodle admin rights unneeded, just teacher ones) :
 - Add `"use_moodle":true` in `server_config.json`
 - OAuth & LTI protocols: students authenticate via a click in Moodle
- Remaining tasks:
 - Migrate the `oauth-moodle` branch : OCaml 4.05 → 4.12
 - Add the `irmin` dependency (requiring OCaml > 4.08)
 - Extend the documentation
 - Release `learn-ocaml.el` accordingly

Supporting Vg-based exercises in Learn-OCaml

- <https://opam.ocaml.org/packages/vg/>
- "Declarative 2D vector graphics in OCaml"
- Learn-OCaml support: on-going implementation by Étienne Marais, Émile Rolley, Yann R.-G.
- Bottlenecks:
 - Use the rendering engine of browsers?
Reimplement a rendering engine within OCaml?
 - Understand how to generate a test report that is as useful as possible for students?

Multi-part exercises

- (issue #331) Feature wish: Organize an exercise into sub-parts, where one can navigate using left and right arrows. . .
- (issue #395) Feature wish: Handle several different graders for the same exercise. . .
- Objectives:
 - Support these two feature wishes in one go
 - Make design choices with a special focus on **compatibility** (the implementation code of a single-part exo should be kept as is; a multi-part exo should be partly rendered by a non-multi-part-aware server)
- Milestone:
 - The design has been completed and is satisfactory: PoC devised by Yoan Mollet (former L3 intern at IRIT), EMD, YRG.
 - To do: full-stack implementation (JS, backend, learn-ocaml-mode): Jossy Yila (M1 intern at IRIT): **May-July 2022**