



[Announcements](#) › Final project details

Final project details

Hanneli Andreazzi Tavante posted on Mar 24, 2021 10:18 PM • Edited

Hi everyone! 🙌

The final project is released! The instructions are here.

You must submit your final project as one ml file to MyCourses, under the Assignment tab, in Final Project Submission. You can submit as many times as you want but only the last submission counts.

Please note that the deadline for the final project is hard and fixed. It is governed by McGill policies and controlled by MyCourses. Please get your project started early and submit your project by the deadline.

You will notice the size of the codebase will be much larger than the usual HW assignments. While it will be possible to develop it in LearnOCaml, we are aware of technical issues with the platform and strongly recommend backing up your progress **often** if you do so. You can also opt-in for developing it in your local machine.

About your Submission

Repeating the previous information, **submit your final project as one ml file to MyCourses, under the Assignment tab, in Final Project Submission. You can submit as many times as you want but only the last submission counts.**

You should only submit the part of the code that you worked on. This means either what was editable by you in LearnOCaml, or the section delimited by appropriate comments in the [all-in-one file on MyCourses](#) (starting around line 1846).

Working on The Project on your local machine

Before using this template file, please have it clear: ****only submit the portion that corresponds to the editable buffer on LearnOCaml****. Again, ****DO NOT**** submit the whole all-in-one file to us as it contains code that comes from us. ****ONLY**** read forward if you understand this criterion.

To be able to work on the project locally, you will need to install the OCaml toolchain.

If you use Windows, check the instructions described here <https://ocaml.org/docs/install.html#Windows> to install Opam.

For MacOS/Linux distros, the instructions below should work:

1. Install `opam`: <https://opam.ocaml.org/doc/Install.html> - Opam will let you easily install one (or multiple) versions of the OCaml compiler.
2. Install OCaml - here we suggest 4.05.0, but you can install newer versions if you want. First, we create a switch containing OCaml v. 4.05.0...

```
opam switch create 4.05.0
```

3. Then we effectively switch to it:

```
opam switch 4.05.0
```

4. Open the project files (they will be available soon) in your favourite text editor. If you use VSCode, take a look at <https://marketplace.visualstudio.com/items?itemName=ocaml-lsp> platform .

As the instructions from the previous link suggest, once you install the plugin, do not forget to install `ocaml-lsp`:

```
opam install ocaml-lsp-server
```

Check out <https://github.com/ocaml/ocaml-lsp> for more information about `ocaml-lsp`.

For IntelliJ, <https://plugins.jetbrains.com/plugin/9440-reasonml> (more recent plugin) or <https://plugins.jetbrains.com/plugin/4986-ocaml-support> should work. Eclipse and other IDEs offer plugins as well. Vim or Emacs are also candidates!

Pick your editor and download the [project all-in-one file](#). In this file, search for

```
(** Template Begins Here *)
```

What you see on the LearnOCaml buffer should come after this line. ****DO NOT**** edit any code before this line.

Compile the file:

```
ocaml -w all all_in_one_final_project.ml
```

`-w all` turns on all warnings. We ask you to make sure your code is warning-free with this option

on.

Testing Your Code

You can run REPL on your computer by running ``ocaml``. In the prompt, type

```
#use "all_in_one_final_project.ml";;
```

You can then test your code in the REPL. You don't have access to the hidden tests so you are left to your own tests.

Improving Your Productivity

Depending on your editor, you might want to install a full toolchain for code completion, etc.

The default REPL *ocaml* might not be very convenient. You might want to use an improved version, *utop*. You can install *utop* by running:

```
opam install utop
```

Then you will be able to run

```
utop
```

You will see that *utop* has much a better interactive experience.

Reminder: if you decide to implement the project in LearnOCaml, make sure to backup your solution regularly!

Happy coding! ~