

# **Préambule**

**Tristan Crolard**

Laboratoire CEDRIC  
Equipe « Systèmes Sûrs »

**`tristan.crolard@cnam.fr`**

**`cedric.cnam.fr/sys/crolard`**

OCaml

Learn Packages Community Blog Playground

Search OCaml packages

Get Started

# OCaml

An industrial-strength functional programming language with an emphasis on expressiveness and safety

Install

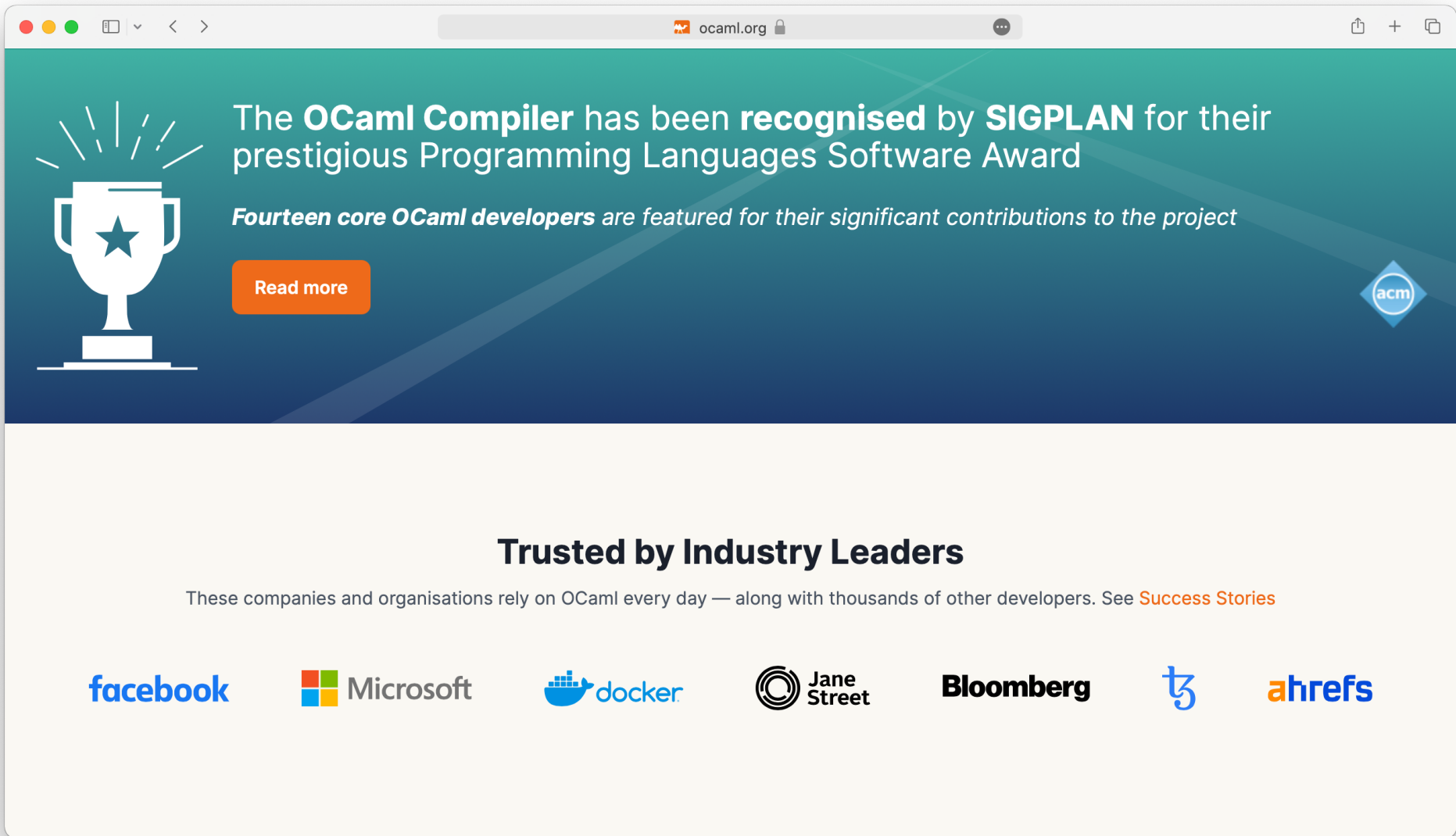
About OCaml

Latest release: 5.1.0

Long Term Support release: 4.14.1

```
# let square x = x * x
val square : int -> int = < fun >
# square 3
- : int = 9
# let rec fac x =
  if x <= 1 then 1 else x * fac (x - 1)
val fac : int -> int = < fun >
# fac 5
- : int = 120
# square 120
- : int = 14400
```

Try the Playground



ocaml.org

OCaml Learn Packages Community Blog Playground

Search OCaml packages **Get Started**

Linux, macOS or \*BSD Windows

## Install OCaml on Linux, macOS, or \*BSD

- Install the opam package manager**

OCaml's package manager **opam** installs both the compiler, as well as any additional packages. Ensure **gcc**, **build-essential**, **curl**, **unzip**, and **bubblewrap** are installed on your system, then run the following in your terminal to download and install the newest version of **opam**:

```
bash -c "sh <(curl -fsSL https://raw.githubusercontent.com
```

[view script on GitHub](#)

Other installation methods ▾
- Initialise opam**

**opam** needs to be initialised, which will create a default **opam switch**. An **opam** switch is an isolated environment for the OCaml compiler and any packages you install.

## Set Up an OCaml Development Environment

The **OCaml Platform Tools**, which include the **build system dune**, complete your OCaml development environment. To install them in your current **opam** switch, run this command:

```
opam install dune merlin ocaml-lsp-server odoc ocamlformat
```

Now you are ready to write some OCaml code!

[First Steps with OCaml](#) →

If the instructions on this page did not work for you, or you want to know more, you can find [a more detailed explanation of the installation process here](#).

The screenshot shows a web browser window with the URL `ocaml.org`. The page header includes the OCaml logo and navigation links: [Learn](#), [Packages](#), [Community](#), [Blog](#), and [Playground](#). There is a search bar for OCaml packages and a [Get Started](#) button. Below the header, there are two buttons: [Linux, macOS or \\*BSD](#) and [Windows](#). The main heading is **Install OCaml on Windows**. The first step is **1. Use the Diskuv OCaml ("DKML") Windows installer**. The text explains that the installer sets up OCaml 4.14.0, OCaml's package manager `opam`, Git, and the Visual Studio compiler. It also mentions alternative installation instructions in the ["OCaml on Windows" guide](#). A [Download DKML Installer](#) button is provided. A warning states: "Before you run the installer: Make sure your Windows username does **not** contain a space character (e.g. for `C:\Users\Jane Smith`, OCaml will not install properly)." The page concludes with congratulations and information about the [OCaml Platform Tools](#), including the `dune` build system.

The image shows a web browser window with the URL `try.ocamlpro.com`. The page title is "Try OCaml<sup>BETA</sup>". The interface has a dark theme and includes a navigation bar with "Editor" and "Toplevel" tabs. A help icon and "OCaml PRO" logo are also present. The main content area is a terminal window displaying the following text:

```
Welcome to OCaml 4.13.1
- use the editor pane and press [Ctrl-e]
- or type your OCaml phrase in the box below and press [Enter]
- use [Shift-Enter] to break lines without triggering execution
- use [Ctrl-↑] once to reuse the previous entry
- use [Ctrl-↑] / [Ctrl-↓] to navigate through history
```

Below the terminal window is a light gray input field containing a hash symbol `#` and a cursor. At the bottom of the interface are three buttons: "Eval phrase" (with a gear icon), "Reset" (with a circular arrow icon), and "Clear" (with a trash can icon).

The image shows a Replit IDE interface. At the top, the browser address bar displays "replit.com". The main header includes a "Run" button and an "Edit project" dropdown. On the left, a sidebar contains a "Files" section with a search bar and a list of files: ".lesson", ".ocamlformat", "dune", "dune-project", and "Main.ml". Below this is a "Tools" section with icons for Ghostwriter, Chat, Code Search, Console, Database, Docs, Extensions, Git, and Markdown. At the bottom of the sidebar are CPU, RAM, and Storage monitors, and a "Help" button.


The central editor displays a file named "Main.ml" with the following code:

```
1 print_endline "Hello World!"  
2
```

At the bottom, a "Console" window shows the execution of the code:

```
>_ Console x Shell x +  
➤ dune exec --display=quiet ./Main.bc  
Hello World!  
➤ █
```

nodejs.org



HOME | ABOUT | DOWNLOADS | DOCS | GET INVOLVED | SECURITY | CERTIFICATION | NEWS

Node.js® is an open-source, cross-platform JavaScript runtime environment.

## Download for macOS (x64)

**16.18.0 LTS**  
Recommended For Most Users

**18.11.0 Current**  
Latest Features

[Other Downloads](#) | [Changelog](#) | [API Docs](#)   [Other Downloads](#) | [Changelog](#) | [API Docs](#)


For information about supported releases, see the [release schedule](#).

Copyright OpenJS Foundation and Node.js contributors. All rights reserved. The OpenJS Foundation has registered trademarks and uses trademarks. For a list of trademarks of the OpenJS Foundation, please see our [Trademark Policy](#) and [Trademark List](#). Trademarks and logos not indicated on the list of OpenJS Foundation trademarks are trademarks™ or registered® trademarks of their respective holders. Use of them does not imply any affiliation with or endorsement by them.

[The OpenJS Foundation](#) | [Terms of Use](#) | [Privacy Policy](#) | [Bylaws](#) | [Code of Conduct](#) | [Trademark Policy](#) | [Trademark List](#) | [Cookie Policy](#) | [Edit On GitHub](#)



esy 0.6.12    Getting started    Docs    Blog    Community    Search    GitHub



# esy

Easy package management for native Reason,  
OCaml and more

[Get Started](#)    [How it works](#)

**package.json** Driven    Project Isolation    Fast, Teleporting Builds    Deterministic and Offline

**Familiar** `npm` inspired dependency management.    Develop **multiple projects** simultaneously without conflict.    All local projects automatically share **build caches** with each other and caches support **teleportation** across network.    Generate **lock** files and dependency source **snapshots** for ultra-reliable, corporate-friendly builds. Network optional.

OCaml Learn Packages Community Blog Playground Search OCaml Packages [Get Started](#)

# dune

Fast, portable, and opinionated build system

Version: 3.4.1

[Overview](#) [Documentation](#)

## README

Dune is a build system designed for OCaml/Reason projects only. It focuses on providing the user with a consistent experience and takes care of most low-level details of OCaml compilations. It's merely necessary to provide a description of your project, and Dune will do the rest.

## 2.6 Building a Hello World Program in Bytecode

In a directory of your choice, write this dune file:

```
;; This declares the hello_world executable implemented by hello_world.ml
;; to be build as native (.exe) or bytecode (.bc) version.
(executable
 (name hello_world)
 (modes byte exe))
```

This hello\_world.ml file:

```
print_endline "Hello, world!"
```

And build it with:

```
dune build hello_world.bc
```

The executable will be built as `_build/default/hello_world.bc`. The executable can be built and run in a single step with `dune exec ./hello_world.bc`. This bytecode version allows the usage of `ocamldebug`.

2.7 Setting the OCaml Compilation Flags Globally

Write this dune file at the root of your project:

```
(env
  (dev
    (flags (:standard -w +42)))
  (release
    (ocamlopt_flags (:standard -O3))))
```

*dev* and *release* correspond to build profiles. The build profile can be selected from the command line with `--profile foo` or from a *dune-workspace* file by writing:

```
(profile foo)
```