

Le langage de programmation utilisé en option informatique s'appelle OCaml. Ce document vous explique comment installer OCaml ainsi que l'environnement de développement *Visual Studio Code (VS code)*. Si vous le souhaitez, vous pouvez utiliser un autre environnement de développement.

### 1 Installation de VS Code

**Étape 1.** Installez VS Code à partir de la page officielle :

`https://code.visualstudio.com/Downloads`

**Étape 2.** Téléchargez :

`https://informatique-lhp.fr/ocamlterminal.vsix`

Ce fichier contient une extension pour VS Code permettant d'exécuter facilement les programmes OCaml. Pour installer cette extension :

- Ouvrir VS Code et cliquer sur « **View > Command Palette...** ». Une barre de recherche doit s'ouvrir.
- Rechercher et exécuter la commande « **Extensions: Install from VSIX...** ».
- Sélectionner le fichier `ocamlterminal.vsix` que vous venez de télécharger.

Si tout s'est bien passé, lorsque vous cliquez sur l'icône **Extensions** à gauche de VS Code, l'extension « **Ocaml Terminal** » doit être installée.

**Étape 3.** Pour activer la coloration syntaxique des programmes OCaml, il faut ajouter une extension à VS Code :

- À gauche, cliquez sur l'icône **Extensions**.
- Rechercher "OCaml Platform" et installez cette extension.

### 2 Installation d'OCaml

Les explications de cette partie sont principalement issues du site :

`https://ocaml.org/install`

Attention :

- Cette procédure ne fonctionne pas avec le réseau du lycée. Connectez vous à internet par un autre moyen.
- Pour éviter tout problème lors de l'installation, désactivez la mise en veille automatique de votre ordinateur.

## 2.1 Sous Windows

**Étape 1.** Ouvrir le logiciel « Windows PowerShell » et lancer la commande :

```
winget install Git.Git OCaml.opam ; echo "Terminé"
```

Lorsque le message « Terminé » s'affiche, vous pouvez passer à l'étape suivante.

**Étape 2.** Ouvrir un nouveau PowerShell et lancer la commande :

```
opam init ; echo "Terminé"
```

Lorsque le terminal vous pose la question « How should opam obtain Unix tools? », choisissez l'option :

```
Automatically create an internal Cygwin installation  
that will be managed by opam (recommended)
```

Lorsqu'il vous pose la question « Alternatively, would you like to select a different shell? », répondez « n » (pour no).

L'installation dure plusieurs dizaines de minutes, attendez que le message « Terminé » s'affiche pour passer à l'étape suivante.

**Étape 3.** Exécuter la commande :

```
(& opam env) -replace "$([char]0x251C)$([char]0x00BF)",'è' -split '\r?\n' | ForEach-Object { Invoke-Expression $_ }
```

puis :

```
opam install utop ; echo "Terminé"
```

Lorsque le message « Terminé » s'affiche, vous pouvez passer à la partie suivante.

## 2.2 Sous Mac

**Étape 1.** La première étape consiste à télécharger *homebrew*. C'est un gestionnaire de paquets, c'est à dire un programme qui permet d'installer d'autres programmes. Pour télécharger homebrew, ouvrir un terminal et lancer la commande :

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

puis suivre les instructions affichées dans le terminal.

**Étape 2.** Il faut maintenant demander à homebrew d'installer les logiciels dont on aura besoin en exécutant dans le terminal :

```
brew install gcc bubblewrap
```

**Étape 3.** Télécharger ensuite OCaml avec la commande :

```
bash -c "sh <(curl -fsSL https://opam.ocaml.org/install.sh)"
```

**Étape 4.** Exécuter la commande :

```
opam init
```

Lorsque le terminal vous pose la question « Do you want to configure zsh? », choisissez l'option :

```
Yes, update ~/.zshrc
```

**Étape 5.** Exécuter la commande :

```
opam install utop
```

## 2.3 Sous Linux

**Étape 1.** Dans un terminal administrateur :

```
apt-get install gcc build-essential unzip bubblewrap
```

**Étape 2.** Dans un terminal (non administrateur) :

```
bash -c "sh <(curl -fsSL https://opam.ocaml.org/install.sh)"
```

puis :

```
opam init          et          opam install utop
```

### 3 Un premier programme en OCaml

Pour tester si les installations ont fonctionné :

**Étape 1.** Ouvrir VS Code, cliquer sur `File > New File...` et nommer le fichier « `test.ml` ». Cliquer sur `Terminal > New Terminal`, un terminal s’ouvre alors en bas de la fenêtre.

**Étape 2 (si vous êtes sous Windows).** Si vous êtes sous Windows, dans le terminal exécutez la commande :

```
(& opam env) -replace "$([char]0x251C)$([char]0x00BF)", 'è' -split '\r?\n' | ForEach-Object { Invoke-Expression $_ }
```

puis :

```
utop
```

Vous devez exécuter les deux commandes ci-dessus à chaque fois que vous relancez OCaml. Notez qu’il n’est pas nécessaire de les réécrire à chaque fois, puisque vous pouvez les retrouver dans l’historique en cliquant dans le terminal et en utilisant la flèche vers le haut du clavier.

La partie 4.1 de ce document explique comment configurer VS Code pour se passer de la première commande et lancer OCaml directement avec la commande « `utop` ».

**Étape 2 (Si vous êtes sous Mac ou linux).** Dans le terminal, lancer la commande « `utop` »

**Étape 3.** Dans le fichier source, écrire le programme :

```
let f n =
  let a = n + 1 in
  let b = a * n in
  a+b;;

(* Tests *)
f 4;;
f 5;;
```

Enregistrez votre fichier, puis cliquez sur l’icône  en haut à droite (ou utilisez le raccourci clavier « `Ctrl + Maj + Entrée` »). Si tout s’est bien passé, le terminal doit afficher les résultats 25 et 36.

**Remarque.** Si le raccourci « `Ctrl + Maj + Entrée` » saute une ligne au lieu de lancer l’exécution, c’est peut-être que VS code s’est mis en “Restricted mode”. Dans ce cas, cliquer sur “Restricted mode” en bas à gauche, puis sur “Trust”.

## 4 À lire maintenant et/ou plus tard

À partir de maintenant, le but est de configurer VS Code pour pouvoir utiliser OCaml plus simplement. Il n'est pas nécessaire de faire tout ce qui suit dès maintenant, vous pourrez y revenir plus tard.

### 4.1 Simplification de l'utilisation sous Windows

Cette partie concerne uniquement les personnes sous Windows. Le but est de se passer de la commande :

```
(& opam env) -replace "$([char]0x251C)$([char]0x00BF)", 'è' -split '\r?\n' | ForEach-Object { Invoke-Expression $_ }
```

Sous VS Code :

- Dans les paramètres (icône en bas à gauche), ouvrir « Settings » puis cliquer sur l'icône  en haut à droite.
- Dans le fichier `settings.json` qui vient de s'ouvrir, ajoutez les lignes suivantes (sauf la première et les deux dernières) :

```
{ // Cette accolade est déjà présente au début du fichier
  "terminal.integrated.profiles.windows": {
    "PowerShell with OCaml": {
      "source": "PowerShell",
      "args": [
        "-NoExit",
        "-Command",
        "(& opam env) -replace \"$([char]0x251C)$([char]0x00BF)\", 'è' -split '\r?\n' | ForEach-Object { Invoke-Expression $_ }"
      ]
    }
  },
  "terminal.integrated.defaultProfile.windows": "PowerShell with OCaml",
  // Si le fichier contenait d'autres lignes, ne les effacez pas
} // Cette accolade est déjà présente à la fin du fichier
```

- Enregistrez le fichier `settings.json` .

Vous devriez maintenant pouvoir lancer OCaml dans un nouveau terminal directement avec la commande « `utop` ».

### 4.2 Module Quick Print

Pour installer des modules sous OCaml, il faut utiliser le logiciel `opam` (c'est l'équivalent du logiciel `pip` associé à Python). Dans un terminal, lancez la commande :

```
opam install quick_print
```

Le module `quick_print` permet d'afficher des objets de différents types. Par exemple, le programme suivant doit afficher la liste `[1;2;3;4]` dans le terminal :

```
#require "quick_print";;
Quick_print.int_list [1;2;3;4];;
```

### 4.3 Mise à jour d'OCaml

Pour mettre OCaml à jour (à faire régulièrement si vous le souhaitez), exécutez les deux commandes suivantes dans un terminal :

```
opam update      puis      opam upgrade
```