

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

Installez VS Code à partir de la page officielle :

<https://code.visualstudio.com/Downloads>

## 2 Installation d'OCaml

### 2.1 Sous Windows

Pour installer OCaml, trois méthodes sont possibles. Si l'une d'elles ne fonctionne pas sur votre PC, utilisez en une autre.

**Méthode 1.** Installez OCaml à partir du « Graphical Installer - 64-bits » de :

<https://fdopen.github.io/opam-repository-mingw/installation/>.

Lors de l'installation, pensez à noter le dossier d'installation qui devrait être de la forme « C:/OCaml64 ».

**Méthode 2.** Lancez le programme d'installation DKML disponible dans l'onglet « Windows » de la page :

<https://ocaml.org/install>

Quelques remarques :

- ★ L'installation est assez longue (environ 1h30).
- ★ Si le message « The machine needs rebooting » s'affiche, redémarrez votre PC et relancez le programme d'installation.
- ★ Si besoin, les instructions détaillées sont disponibles ici :

<https://ocaml.org/docs/installing-ocaml#1-use-the-dkml-installer>

**Méthode 3.** Si les méthodes précédentes ne fonctionnent pas, vous pouvez utiliser :

[https://informatique-lhp.fr/opt-mpsi/installation\\_ocaml.exe](https://informatique-lhp.fr/opt-mpsi/installation_ocaml.exe).

N'utilisez cette méthode qu'en dernier recours car il s'agit d'une ancienne version d'OCaml qui ne peut pas être mise à jour et avec laquelle on ne peut pas installer de modules.

### 2.2 Sous Mac

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 (ici, on l'utilisera pour installer OCaml). Pour télécharger homebrew :

- ★ Depuis le launchpad, ouvrez l'application « terminal ».
- ★ Dans le terminal exécutez la commande :

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

- ★ Suivez les instructions pour installer homebrew.

Il faut maintenant demander à homebrew d'installer OCaml :

- ★ Dans un terminal, exécutez :

```
brew install opam ocaml
```

- ★ Suivez les instructions pour installer OCaml

## 2.3 Sous Linux

Dans un terminal :

```
apt-get install opam ocaml
```

## 3 Un premier programme en OCaml

Pour tester si les installations ont fonctionné :

- ★ Ouvrez VS Code, cliquez sur **File > New File...** et appelez le fichier « `test.ml` ».
- ★ Cliquer sur **Terminal > New Terminal**. Un terminal s’ouvre alors en bas de la fenêtre.
- ★ En fonction de votre système d’exploitation :
  - Si vous êtes sous Windows et que vous avez installé OCaml avec la méthode 1. Dans le terminal, exécutez la commande :

```
C:/OCaml64/usr/local/bin/ocaml-env.exe exec -- ocaml
```

en remplaçant `C:/OCaml64` par le dossier où est installé OCaml (ce dossier a été choisi lors de l’installation). Attendre quelques secondes qu’OCaml se lance.

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

La partie 4.3 de ce document explique comment configurer VS Code pour lancer OCaml directement avec la commande « `ocaml` ».

- Dans tous les autres cas : dans le terminal, tapez `ocaml` puis sur la touche **Entrée**.

- ★ Écrivez le programme ci-contre dans le fichier source.

- ★ Avec la souris, sélectionnez ce programme et cliquez sur **Terminal > Run Selected Text**.

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

Si tout s’est bien passé, le terminal doit afficher les résultats 25 et 36.

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

## 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 Coloration syntaxique

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.

Lorsque vous enregistrez un fichier source avec un nom finissant par l’extension `.ml`, le texte du fichier doit se colorer.

### 4.2 Raccourcis claviers pour exécuter les programmes

**Raccourci clavier 1.**


- ★ En bas à gauche, cliquez sur l’icône des paramètres, puis sur « **Keyboard Shortcuts** ».
- ★ Recherchez « **Terminal: Run Selected Text In Active Terminal** » et définissez le raccourci clavier « **Ctrl + Entrée** ».

**Ctrl + Entrée** est maintenant équivalent à utiliser **Terminal > Run Selected Text**.

**Raccourci clavier 2.** Pour exécuter l'intégralité d'un fichier source en OCaml, il faut écrire dans le terminal :

```
#use "<chemin vers le fichier source>;";
```

en remplaçant <chemin vers le fichier source> par le chemin du fichier source. Pour se faciliter la tâche, nous allons créer un raccourci clavier qui exécute automatiquement cette commande :

- ★ À gauche, cliquez sur l'icône **Extensions**, recherchez « **Copy Relative Path Posix** » et installez cette extension.
- ★ Ouvrez le fichier `keybindings.json`. Pour cela, allez dans les paramètres (icône en bas à gauche), ouvrez **Keyboard Shortcuts**, puis cliquez sur l'icône  en haut à droite.
- ★ Dans le fichier `keybindings.json`, ajoutez les lignes suivantes (sauf la première et les deux dernières) :

```
[ // Ce crochet est déjà présent dans le fichier
  {
    "command": "runCommands",
    "key": "ctrl+shift+enter",
    "args": {
      "commands": [
        "saveAll",
        {
          "command": "workbench.action.terminal.sendSequence",
          "args": { "text": "#use \"\""}
        },
        "copyRelativePathPosix",
        "workbench.action.terminal.paste",
        {
          "command": "workbench.action.terminal.sendSequence",
          "args": { "text": "\"";\u000D"}
        },
        "workbench.action.focusActiveEditorGroup",
      ]
    },
  },
  // Si le fichier contenait d'autres lignes, ne les effacez pas
] // Ce crochet est déjà présent dans le fichier
```

puis enregistrez le fichier.

Si tout s'est bien passé, **Ctrl + Maj + Entrée** doit maintenant lancer la commande « `#use "<chemin vers le fichier source>;";` » dans le terminal. Attention :

- ★ Sous windows, le chemin absolu du fichier `.ml` ne doit pas contenir d'accent. Par exemple, si votre nom d'utilisateur est "Élève" (c'est le cas sur les PC Grand-Est), vous ne pouvez pas enregistrer votre fichier dans vos documents. Pour contourner ce problème, vous pouvez mettre votre fichier dans un dossier de la forme `C:/info/`.
- ★ Ce raccourci clavier remplace le contenu du presse-papier (utilisé lors d'un copier/coller) par le chemin absolu du fichier source.

**Résumé.** Vous avez maintenant deux possibilités :

- Pour exécuter partiellement le fichier, sélectionnez la portion de code qui vous intéresse et utilisez **Ctrl + Entrée**. Le programme doit nécessairement se terminer par un double point-virgule (`;;`) pour être exécuté.
- Pour exécuter l'intégralité du fichier source, utilisez **Ctrl + Maj + Entrée**. OCaml affichera un résultat à chaque fois qu'il rencontrera un double point-virgule.

Bien sûr, vous pouvez choisir d'autres raccourcis clavier.

### 4.3 Simplification de l'utilisation sous Windows

Cette partie concerne uniquement les personnes sous Windows ayant installé OCaml avec la méthode 1.

**Étape 1.** Le but de cette étape est de se passer de la commande :

```
C:/OCaml64/usr/local/bin/ocaml-env.exe exec -- ocaml
```

Créez un nouveau raccourci Windows (sur votre bureau ou autre) ayant pour emplacement :

```
C:/OCaml64/usr/local/bin/ocaml-env.exe exec -- code.cmd
```

en remplaçant « C:/OCaml64 » par le dossier où est installé OCaml (ce dossier a été choisi lors de l'installation). Lorsque vous cliquez sur ce raccourci, VS code s'ouvre et vous devriez pouvoir lancer OCaml dans un nouveau terminal directement avec la commande « ocaml ».

**Étape 2.** Le but est d'indiquer à OCaml où télécharger d'éventuels nouveaux modules et mises à jour (cette étape est donc nécessaire avant les parties suivantes). Dans un nouveau terminal, exécutez les commandes :

```
★ opam repository add OpamOcamlOrg https://opam.ocaml.org/  
★ opam repository remove default
```

### 4.4 Module Quick Print

Cette partie ne fonctionnera pas si vous êtes sous Windows et que vous avez installé OCaml avec la méthode 2 ou 3.

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 ocamlfind quick_print
```

Le paquet `ocamlfind` sert à importer des modules. 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 :

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

### 4.5 Mise à jour d'OCaml

Cette partie ne fonctionnera pas si vous êtes sous Windows et que vous avez installé OCaml avec la méthode 3.

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
```

### 4.6 Paquets supplémentaires (facultatif)

Cette partie ne fonctionnera pas si vous êtes sous Windows et que vous avez installé OCaml avec la méthode 2 ou 3.

L'extension `OCaml Platform` conseille d'installer quelques paquets supplémentaires. Ces paquets ajoutent certaines fonctionnalités à VS Code (analyse statique, complétion automatique ...), mais sont loin d'être indispensables pour le cours d'option informatique de MPSI. Si vous le souhaitez, vous pouvez les installer avec :

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