1 Installation d'OCaml et de VSCode

Pour installer OCaml et VSCode sur votre ordinateur, suivez les instructions disponibles sur cette page : https://anthonylick.com/installer-ocaml-avec-vscode/

N'hésitez pas à m'appeler si vous avez des questions.

2 Prise en main d'OCaml et de VSCode

2.1 Utilisation de l'interpréteur

Il y a plusieurs manières d'exécuter du code OCaml. La première que nous allons voir dans ce TP consiste à utiliser un **interpréteur** : on envoie un bout de code dans une console OCaml, et il est évalué directement. Dans ce cas, l'instruction à envoyer doit se terminer par ; ; pour que l'interpréteur comprenne que c'est la fin du code à évaluer. Si vous avez suivi le tutoriel d'installation de VSCode, vous devez disposez d'un raccourci clavier pour sélectionner automatiquement l'expression qui se trouve sous votre curseur et l'envoyer à l'interpréteur (CTRL+E si vous avez suivi le tutoriel).

Exercice 1 (utilisation de l'interpréteur) : Prévoir la réponse de **OCaml** aux instructions suivantes, puis tester dans VSCode si votre prédiction est correcte :

let a = 42 ;; let f x = 3 * x + 1 ;; f 1 ;; let g x = 3. * x + 1 ;;

```
let b = max_int ;;
let c = max_int + 1 ;;
let d = float_of_int b ;;
let e = int_of_float d ;;
```

Exercice 2 (déclaration de fonctions) :

1. Écrire une fonction hmin_to_minutes : int -> int -> int prenant en arguments un nombre d'heures et un nombre de minutes et renvoyant le nombre de minutes total correspondant.

```
# hmin_to_minutes 2 15 ;;
- : int = 135
```

2. Écrire une fonction minutes_to_hmin : int \rightarrow int * int prenant en argument un nombre de minutes et renvoyant un couple (h, m) où h est le nombre d'heures correspondant et m est le nombre de minutes restantes.

```
# minutes_to_hmin 135 ;;
- : int * int = (2, 15)
```

3. Écrire une fonction est_pair : int -> bool prenant en argument un entier et renvoyant true s'il est pair (et false sinon).

```
# est_pair 11 ;;
- : bool = false
# est_pair 12 ;;
- : bool = true
```

4. Écrire une fonction implication : bool -> bool -> bool implémentant l'implication logique.

```
# implication true false ;; (* tous les autres cas renvoient true *)
- : bool = false
```

5. Écrire une fonction carre : int -> int qui calcule le carré d'un entier.

carre 5 ;; - : int = 25

2.2 Compilation et exécution depuis un terminal

Une autre manière d'exécuter du code OCaml est de le sauvegarder dans un fichier dont l'extension est .ml et d'interpréter tout le fichier via la commande suivante à saisir dans un terminal :

\$ ocaml nom_du_fichier.ml

On peut aussi le compiler pour produire un fichier exécutable, puis exécuter le fichier obtenu (cela génère également des fichiers .cmi et .cmo qui sont résiduels à la compilation) :

\$ ocamlc nom_du_fichier.ml -o executable
\$./executable

Exercice 3 (exécution d'un fichier) :

1. Écrire le code suivant, et le sauvegarder dans un fichier nommé hello.ml :

let () = Printf.printf "Hello World!\n"

2. Ouvrir un terminal, et utiliser la commande cd pour vous déplacer dans le dossier où vous avez sauvegardé votre fichier. Vous pouvez utiliser la commande 1s pour lister les fichiers présents dans le dossier courant, et vérifier que vous êtes bien dans le bon dossier.

```
$ cd chemin/vers/le/dossier/
$ ls
hello.ml
```

3. Exécuter la commande suivante :

```
$ ocaml hello.ml
Hello World!
```

4. Compiler puis exécuter le fichier obtenu :

```
$ ocamlc hello.ml -o hello
$ ls
hello hello.cmi hello.cmo hello.ml
$ ./hello
Hello World!
```

Exercice 4 (programme avec entrée/sortie) : Écrire un programme demandant à l'utilisateur de saisir un nombre de minutes et affichant sa conversion en nombre d'heures + minutes restantes. N'oubliez pas de la compiler et de le tester.

```
$ ocamlc conversion.ml -o conv
$ ./conv
Veuillez saisir un nombre de minutes : 135
135 minutes correspondent à 2h15.
```

Exercice 5 (quine) : Une **quine** est un programme dont l'exécution affiche à l'écran exactement le code dont il est constitué; autrement dit, les instructions **cat quine.ml** et **ocaml quine.ml**; echo "" doivent produire exactement le même texte dans le terminal. Écrire une quine OCaml.

3 Le projet Euler

Si vous avez fini cette feuille d'exercices, vous pouvez vous rendre sur https://projecteuler.net/ pour trouver d'autres problèmes à résoudre.