

🔑 Créer un exécutable qui prend des arguments en ligne de commande en OCAML

Il est possible de passer des paramètres à un programme OCAML compilé `prgm` au moment où on lance son exécution en ligne de commande. On rappelle que la commande ci-dessous compile le fichier `code.ml` en un exécutable `prgm`.

```
ocamlc code.ml -o prgm
```

Contrairement à ce qui est fait en C, le point d'entrée du programme n'est pas une fonction particulière. En effet l'exécution de `prgm` a pour effet l'évaluation de toutes les valeurs définies dans `code.ml`, en particulier celles de type `unit` dont on peut observer les effets de bord. Ainsi, de même qu'on peut définir un jeu de tests par `let test_f:unit = ...`, on peut définir la suite d'instructions principales par `let main:unit = ...`.

On accède aux arguments passés au programme à travers le tableau `Sys.argv`.

Comme en C, il s'agit d'un tableau de chaînes de caractères (i.e. de type `string array`) initialisé selon les mots de la ligne de commande. En particulier `Sys.argv(0)` est le nom de la commande (rarement utile), et `Array.length Sys.argv` est le nombre de paramètres + 1.

Deux exemples

```
1 let main : unit =
2   print_string ("ici on a lancé la commande \" ^Sys.argv.(0)^\n");
3   let n = Array.length Sys.argv in
4   for i=1 to n-1 do
5     print_string ("le paramètre " ^string_of_int i^" est " ^Sys.argv.(i)^\n")
6   done;
```

Si `affiche_param` est le programme obtenu en compilant le code ci-dessus, `./affiche_param 12 bjr` produit l'affichage suivant.

```
ici on a lancé la commande " ./affiche_param"
le paramètre 1 est 12
le paramètre 2 est bjr
```

```
1 let main : unit =
2   let n = Array.length Sys.argv in
3   let s = ref 0 in
4   for i=1 to n-1 do
5     s := !s + int_of_string Sys.argv.(i)
6   done;
7   print_string("la somme des paramètres vaut " ^string_of_int !s^\n")
```

Si `add` est le programme obtenu en compilant le code ci-dessus, alors

- `./add` affiche la somme des paramètres vaut 0
- `./add 1 2 6` affiche la somme des paramètres vaut 9
- `./add 1 2 salut` affiche l'erreur Fatal error: exception Failure("int_of_string")

NB : lors de l'import avec `#use` sous `utop`, il n'est pas possible de passer des paramètres, seul `Sys.argv.(0)` est rempli avec le chemin jusqu'à l'exécutable `utop`.