

Input/Output en OCAML

Afin de lire/d'écrire dans des fichiers en OCAML, on utilise les deux fonctions suivantes.

- `output_string` : `out_channel -> string -> unit` qui se comporte comme `print_string` sauf qu'elle prend en premier argument un descripteur du fichier dans lequel elle doit écrire (de type `out_channel`).
- `input_line` : `in_channel -> string` qui prend en argument un descripteur du fichier dans lequel elle doit lire (de type `in_channel`) et retournant la prochaine ligne du fichier. Dans le cas où l'on est arrivé en fin de fichier, cette fonction lève l'exception `End_of_file`.

Descripteurs de fichiers. Le descripteur d'un fichier ouvert en lecture est de type `in_channel`, tandis que celui d'un fichier ouvert en écriture est de type `out_channel`. On obtient ces descripteurs à l'aide des deux fonctions suivantes.

- `open_in` : `string -> in_channel` qui prend en argument le nom d'un fichier existant et qui renvoie un descripteur vers ce fichier en lecture (si le fichier n'existe pas une exception `Sys_error` est levée).
- `open_out` : `string -> out_channel` prend en argument le nom d'un fichier et renvoie un descripteur vers ce fichier en écriture. Si le nom donné n'est pas celui d'un fichier existant, un fichier de ce nom est créé. Si le nom donné est celui d'un fichier existant, le contenu dudit fichier est écrasé dès l'ouverture.

Après usage, on ferme les fichiers ouverts à l'aide des deux fonctions suivantes.

- `close_in` : `in_channel -> unit` pour les fichiers ouverts en lecture.
- `close_out` : `out_channel -> unit` pour les fichiers ouverts en écriture.

Exemple Le programme OCAML ci-dessous, lorsqu'il est exécuté en présence d'un fichier `a.txt` de la forme :

```
toto
3 4
```

produit un fichier `b.txt` de la forme :

```
(toto)3 4
```

```
1 let () =
2   let fp1 = open_in "a.txt" in
3   let fp2 = open_out "b.txt" in
4   let ligne1 = input_line fp1 in
5   output_string fp2 "(" ^ ligne1 ^ ")";
6   let ligne2 = input_line fp1 in
7   let tmp1 = int_of_string (String.sub ligne2 0 1) in
8   let tmp2 = int_of_string (String.sub ligne2 2 1) in
9   output_string fp2 (string_of_int tmp1);
10  output_string fp2 " ";
11  output_string fp2 (string_of_int tmp2);
12  close_in fp1;
13  close_out fp2
```