

Utilisation de fichiers

C/OCAML

	En C...	En OCAML...
Ouvrir un fichier (lecture)	<code>FILE* fichier = fopen("./data.txt", "r");</code>	<code>let fichier = open_in "data.txt"</code>
Ouvrir un fichier (écriture)	<code>FILE* fichier = fopen("./data.txt", "w");</code>	<code>let fichier = open_out "data.txt"</code>
Lire un fichier	<code>fscanf(fichier, "%d", &n);</code>	<code>int_of_string (input_line fichier)</code>
Écrire dans un fichier	<code>fprintf(fichier, "%d\n", n);</code>	<code>output_string fichier (string_of_int n)</code>
Fermer un fichier (lecture)	<code>fclose(fichier)</code>	<code>close_in fichier</code>
Fermer un fichier (écriture)	<code>fclose(fichier)</code>	<code>close_out fichier</code>

En C, le `fprintf` s'utilise comme le `printf` : la seule différence est qu'en premier argument, le `fprintf` prend un *flux* (le fichier ouvert). De même avec le `fscanf`, il s'utilise comme le `scanf` et prend en argument le flux, puis le « format » de l'entrée et enfin des pointeurs sur les variables.

Dans l'exemple ci-dessous, on prend un entier dans un fichier `entree.txt` et on le place dans le fichier `sortie.txt`.

En C :

```
int n;

FILE* entree = fopen("entree.txt", "r");
fscanf(entree, "%d", &n);

FILE* sortie = fopen("sortie.txt", "w");
fprintf(sortie, "%d\n", 3);

fclose(entree);
fclose(sortie);
```

En OCAML :

```
let entree = open_in "entree.txt" in
let n = int_of_string (input_line entree) in

let sortie = open_out "sortie.txt" in
output_string sortie (string_of_int n ^ "\n");

close_in entree;
close_out sortie
```