

# Doc - Ocaml : String et Char

## Introduction

Les modules `String` et `Char` en OCaml fournissent des fonctionnalités pour manipuler les chaînes de caractères et les caractères. Les chaînes sont indexées à partir de 0.

## Type principal

`type string`

Il s'agit d'une chaîne de caractères. Les caractères sont de type `char`.

## Notations

À connaître sans rappel.

- Une chaîne de caractères se note entre guillemets doubles : `"hello"`.
- Un caractère se note entre guillemets simples : `'a'`.
- L'opérateur de concaténation de chaînes est `^`. Ainsi `"hello" ^ " " ^ "world"` renvoie `"hello world"`.
- L'accès au *i*-ème caractère d'une chaîne `s` se fait avec `s.[i]`. Attention, l'indexation commence à 0.

## Module Char

- **Conversion caractère vers code ASCII :**

`Char.code` : `char` -> `int`

Renvoie le code ASCII d'un caractère. Par exemple, `Char.code 'A'` renvoie `65`.

- **Conversion code ASCII vers caractère :**

`Char.chr` : `int` -> `char`

Renvoie le caractère correspondant à un code ASCII. Par exemple, `Char.chr 65` renvoie `'A'`. Lève une exception `Invalid_argument` si le code n'est pas valide (0-255).

## Module String - Fonctions principales

- **Longueur d'une chaîne :**

`String.length` : `string` -> `int`

Renvoie le nombre de caractères dans la chaîne. Par exemple, `String.length "hello"` renvoie `5`.

- **Création d'une chaîne :**

`String.make` : `int` -> `char` -> `string`

Crée une chaîne de longueur donnée remplie avec un caractère. Par exemple, `String.make 5 'a'` renvoie `"aaaaa"`. Cas particulier : `String.make 1 c` permet de convertir un caractère `c` en chaîne.

- **Extraction d'une sous-chaîne :**

`String.sub` : `string` -> `int` -> `int` -> `string`

`String.sub s start len` extrait une sous-chaîne de `s` commençant à l'indice `start` et de longueur `len`. Par exemple, `String.sub "hello" 1 3` renvoie `"ell"`.

- **Concaténation de chaînes :**

`( ^ ) : string -> string -> string`

L'opérateur `^` concatène deux chaînes. Par exemple, `"bon" ^ "jour"` renvoie `"bonjour"`.

- **Accès à un caractère :**

`s.[i] : char`

Accède au caractère à l'indice `i` de la chaîne `s`. Par exemple, si `s = "hello"`, alors `s.[1]` renvoie `'e'`. Lève une exception si l'indice est hors limites.

## Fonctions hors programme mais bien utiles

À savoir utiliser en TP.

- `String.contains : string -> char -> bool` renvoie `true` si la chaîne contient le caractère donné.
- `String.split_on_char : char -> string -> string list` découpe une chaîne en une liste de sous-chaînes selon un séparateur.
- `String.concat : string -> string list -> string` concatène une liste de chaînes avec un séparateur.
- `String.trim : string -> string` supprime les espaces en début et fin de chaîne.