[PF5] Cours 1 22/9/14

“

michele.pagani@pps.univ-paris-diderot.fr

Char

1. Valeurs

Caractères ASCII `'a', 'z', ' ', 'W'`

2. Échapement

- `\` : antislash ()
- `\n` : Saut de ligne
- `\r` : retour chariot
- `\t` : tabulation
- `\ddd` : char avec le code ASCII `ddd` en décimal
- `\'` : apostrophe

3. Fonction de conversion

- `Char.code` : `char -> int`
- `Char.chr` : `int -> char`
- `Char.lowercase` : `char -> char`
- `Char.uppercase` : `char -> char`

Voir le module `Char` pour une liste plus complète.

3.1. Exemples

```
# 'a';;
- : char = 'a'

# Char.code 'a';;
- : int = 97

# '\097';;
- : char = 'a'

# '\97';;
    ^^
Error: Illegal backslash escape in string or character (\9)

# Char.uppercase 'a';;
- : char = 'A'

# Char.uppercase '[';;
- : char = '['
```

Module (Intro)

- `Char.code` appelle la fonction code du module **Char**
- La bibliothèque standard de `OCaml` contient pusieurs modules qu'on utilisera par la suite: `Char, String, List, Array, ...`
- Pour appeler une fonction d'un module :
    - Soit on ecrit le nom du module suivi du nom de la fonction:

```
# Char.code;;
- : char -> int = <fun>
```

- Soit ont ouvre le module avec `open nom_Module` puis on appelle les fonctions librement

```
# code;;
  ^^^
Error: Unbound value Code
```

```
# open Char;;
# code;;
- : char -> int = <fun>
```

String

1. valeurs

Chaîne de caractères(entre guillemets ") `"Hello", "a", " ", "\097 est a"`

2. String $\neq$ char