

Cours 1 : Enregistrements à champs modifiables et références

1 Enregistrements à champs modifiables

Un champ précédé du mot-clé `mutable` est modifiable.

```
type t = { prenom : string; mutable age : int; }
```

La syntaxe pour initialiser des champs modifiables est la même que pour les autres champs.

```
let v = { prenom = "toto"; age = 10 }
```

De même, l'accès à un champ se fait de manière identique, qu'il soit modifiable ou non.

```
# v.prenom ^ " a " ^ (string_of_int v.age) ^ " ans";;  
- : string = "toto a 10 ans"
```

La modification de la valeur associée à un champ modifiable se fait avec l'opérateur `<-`.

```
# v.age <- 11;;  
- : unit = ()  
# v.age;;  
- : int = 11
```



L'opérateur `<-` ne retourne pas de valeur, d'où le type `unit` de l'expression `v.age <- 11`. On dit que cet opérateur réalise un *effet de bord*.

L'opérateur `with` réalise une *copie* des champs, qu'ils soient modifiables ou non.

```
# let titi = { toto with prenom = "titi" };;  
val titi : t = { prenom = "titi"; age = 11 }
```

2 Références

Une *référence* est un pointeur sur une cellule mémoire. On crée une référence vers une cellule mémoire contenant l'entier 10 de la manière suivante :

```
let x = ref 10
```

On peut modifier la valeur contenue dans la cellule avec l'opérateur `:=`, appelé opérateur d'*affectation*, et la lire à l'aide de l'opérateur `!`, appelé opérateur de *déréférencement*.

```
# x:=11;;  
- : unit = ()  
# !x;;  
- : int = 11
```

Les références ne sont qu'un *sucre syntaxique* pour des enregistrements à un seul champ mutable. Toutes les remarques de la section précédente (effets de bord, copie) sont donc toujours valables.

```
type 'a ref = { mutable contents : 'a }  
let affect r v = r.contents <- v  
let deref r = r.contents
```



Il ne faut pas confondre les deux types suivants :

```
type t = { prenom : string; mutable age : int }  
type t = { prenom : string; age : int ref }
```



L'opérateur d'égalité `=` ne permet pas de savoir si deux cellules mémoires sont différentes, il ne permet que de comparer leur contenu. Il faut utiliser l'opérateur `==` pour comparer l'adresse de deux cellules.

```
# ref 5 = ref 5;;  
- : bool = true  
# ref 5 == ref 5;;  
- : bool = false
```