Anki

q:

Ocaml
Comment initialiser une liste?
a:

```
let l = [0 ; 1 ; 2]
```

Anki

q:

Ocaml
Comment initialiser une array?
a:

```
let l = [|0 ; 1 ; 2|]
```

Anki

q:

Ocaml
Comment concaténer deux chaines de caractères?
a:

Opérateur `^`.

Anki

q:

Ocaml
physical equality vs structural equality
a:

- physical equality:
  Compare with `==`
  → it compares the addresses of the given values
  → you really shouldn't write code that uses physical equality (`==`) on immutable values

Anki

q:

Ocaml
Comment utiliser des variables mutables? Comment ça marche dans des structs?

Qu'est l'aliasing?
a:

Initialisation
Utilisation de `ref`

```
let mut_bool = ref false in
let mut_list = ref [] in
```

Affectation
Utilisation de `:=`

```
let x = ref 3. ;;
x := (!x)**3. ;;
```

Déréférencement
Utilisation de `!` **avant**

Aliasing
Si `x` est une référence et que l'on écrit `let y=x`, on fait « pointer » `y` vers la même case mémoire que `x` (on parle d'égalité physique). Cela signife que toute modifcation ultérieure du contenu de `x` afectera `y`, et inversement.

Dans un enregistrement
On peut mettre dans un type des champs mutables.
On les edit alors grace à `<-`

```
type compteur =
  { debut : int
  ; fin : int
  ; mutable courant : int
  }
let c = {debut = 1; fin = 5; courant = 1};;
c.courant <- 4;;
```

Anki

q:

Ocaml
Boucle for
a:

OCaml propose deux variantes de la boucle for, selon que l'indice croît ou décroît:

```
for compteur = start to finish do
  expression
done
for compteur = start downto finish do
  expression
done
```

Remarque:
- Les bornes sont incluses
- compteur est considéré comme une variable de type int locale à la boucle et ne peut **pas être modifé** « à la main » dans le corps de la boucle.
- Dans d'autres langages on peut utiliser `for` pour itérer des structures de données: en ocaml, on préfèrera utiliser `List.iter`, `Array.iter`

Anki

q:

Ocaml
Boucle while
a:

```
while condition do
  expression
done;
```

Anki

q:

Ocaml
Utilisation d'exceptions
a:

Les exceptions peuvent etre utilisées pour sortir d'une boucle par exemple. Elles peuvent contenir des données.

La syntaxe du `try` `with` est à bien connaitre, et l'initialisation d'une exception aussi.

```
exception Found of int
let find_first_even lst =
  try
    List.iter (fun x ->
          if x mod 2 = 0 then raise (Found x)) lst;
    None  (* If no even number is found *)
  with
  | Found x -> Some x
```

Duplicate id: 250401

Anki

q:

a:

Duplicate id: 250401

Anki

q:

a:

Duplicate id: 250401

Anki

q:

a: