

Prise en main de CaML Light - 3

Lycée Thiers 2015

Les vecteurs de CaML

- ① Structure linéaire, ordonnée, homogène
- ② Accès direct indexé
- ③ Type paramétré \rightarrow polymorphisme
- ④ Pourquoi des vecteurs ?

Cinq manipulations de base

- 1 Création
- 2 Accès en lecture
- 3 Accès en écriture
- 4 Calcul de la longueur
- 5 Extraction d'un sous-vecteur

Création et accès en lecture

```
let v = [|1;4;2;8;5;7|];;  
v : int vect = [|1;4;2;8;5;7|]
```

⇒ Numérotation à partir de 0

```
v.(0);;  
- : int = 1
```

```
v.(1);;  
- : int = 4
```

```
v.(5);;  
- : int = 7
```

```
v.(6);;
```

Exception non rattrapée:

```
Invalid_argument "vect_item"
```

```
vect_item;;
```

```
- : 'a vect -> int -> 'a = <fun>
```

```
vect_item v 5;;
```

```
- : int = 7;
```

Abbréviation

v.(i) au lieu de `vect_item v i`

Accès en écriture

```
v.(0) <- 0;;  
- : unit = ()
```

```
v;;  
- : int vect = [|0;4;2;8;5;7|]
```

```
v.(6) <- 0;;  
Exception non rattrapée:  
Invalid_argument "vect_assign"
```

```
vect_assign;;  
- : 'a vect -> int -> 'a -> unit = <fun>
```

Accès en écriture

```
vect_assign v 0 1;;  
- : unit = ()  
  
v;;  
- : v = [|1;4;2;8;5;7|]
```

Abbréviation

`v.(i) <- a` au lieu de `vect_assign v i a`

Calcul de la longueur

```
vect_length;;  
- : 'a vect -> int = <fun>
```

```
vect_length v;;  
- : int = 6
```

```
vect_length [||];;  
- : int = 0
```


Création d'un vecteur (bis)

```
make_vect;;
```

```
- : int -> 'a -> 'a vect = <fun>
```

Syntaxe

```
make_vect longueur valeur
```

```
let w = make_vect 6 0;;
```

```
w : int vect = [|0;0;0;0;0;0|]
```

```
for k = 0 to 5 do w.(k) <- k * k done;;
```

```
- : unit = ()
```

```
w;;
```

```
- : int vect = [|0;1;4;9;16;25|]
```

Extraction d'un sous-vecteur

```
sub_vect;;
```

```
- : 'a vect -> int -> int -> 'a vect = <fun>
```

Syntaxe

```
sub_vect vecteur indice_debut longueur
```

```
w;;
```

```
- : int vect = [|0;1;4;9;16;25|]
```

```
sub_vect w 2 3;;
```

```
- : int vect = [|4;9;16|]
```

On récapitule !

- 1 Création :
 - `make_vect`
 - `let v = [|...|]`
- 2 Accès en lecture
`v.(i)`
- 3 Accès en écriture
`v.(i) <- a`
- 4 Longueur :
`vect_length`
- 5 Extraction :
`sub_vect`

Quelques questions classiques ...

- 1 Valeur moyenne
- 2 Plus grand élément
- 3 Rang de la première occurrence d'une valeur
- 4 Echange des valeurs de deux termes
- 5 Tri d'un vecteur

Valeur moyenne

```
let moyenne v =  
  if v = [| |] then failwith "vecteur vide!"  
  else let n = vect_length v in  
    let s = ref 0. in  
    for k = 0 to n-1 do  
      s := !s +. v.(k)  
    done;  
    !s /. (float_of_int n)  
;;  
moyenne : float vect → float = <fun>  
  
moyenne [|1.;2.;3.;4.;5.;99. |];;  
- : float = 19.0
```

Plus grand élément

```
let max_vect v =  
  let n = vect_length v in  
  let m = ref v.(0) in  
    for i = 1 to n-1 do  
      if v.(i) > !m then m := v.(i)  
    done;  
  !m  
;;
```

```
max_vect : 'a vect → a = <fun>
```

```
max_vect [|1;5;13;7;9|];;
```

```
- : int = 13
```

```
max_vect [|"arbre";"arbuste";"abricot"|];;
```

```
- : string = "arbuste"
```

Rang de la première occurrence

```
let rang1 v x =  
  let n = vect_length v in  
  let rg = ref (-1) in  
  let k = ref 0 in  
    while (!k < n) & (!rg < 0) do  
      if v.(!k) = x then rg := !k;  
      k := !k + 1  
    done;  
  !rg  
;;  
rang1 : 'a vect → 'a → int = <fun>  
  
rang1 [|1;1;1;0;1;1;1;0;0|] 0;;  
- : int = 3
```

Echange des valeurs de deux termes

```
let echange_mal v i j =  
  v.(i) <- v.(j);  
  v.(j) <- v.(i)  
;;  
echange_mal : 'a vect -> unit = <fun>
```

```
let echange v i j =  
  let tmp = v.(i) in  
  v.(i) <- v.(j);  
  v.(j) <- tmp  
;;  
echange : 'a vect -> unit = <fun>
```


“Tri à bulles” d’un vecteur

```
let tri_bulles v =  
  let n = vect_length v in  
  for longueur = n downto 2 do  
    for i = 0 to longueur - 2 do  
      if v.(i+1) < v.(i) then echange v i (i+1)  
    done  
  done  
;;  
tri_bulles : 'a vect → unit = <fun>  
  
let v = [|1;3;2;5;4|] in  
  tri_bulles v;  
  v  
;;  
- : [|1;2;3;4;5|]
```

Création d'un vecteur aléatoire

Le module “random” ...

`random__int, random__float`

pas vraiment aléatoire !...

`random__init`

```
random__init (unix_time ());;
```

Création d'un vecteur aléatoire

```
let alea_vect longueur maxval =  
  let v = make_vect longueur 0 in  
  for i = 0 to longueur - 1 do  
    v.(i) <- random_int (maxval + 1)  
  done;  
  v  
;;  
alea_vect : int → int → int vect = <fun>  
  
let v = alea_vect 10 100;;  
v : int vect = [|10; 83; 62; 31; 42; 56; 75|]
```

Tri à bulles d'un vecteur aléatoire

```
let v = alea_vect 10 100;;  
v : int vect = [|10; 83; 62; 31; 42; 56; 75|]  
  
tri_bulles v;;  
- : unit = ()  
  
v;;  
- : int vect = [|10; 31; 42; 56; 62; 75; 83|]
```

The End!