

# TAKE AWAY

## LES RESULTS

A utiliser pour modéliser les cas d'erreur

Sécurisant, surtout quand on dispose d'ADT

Facile à manipuler avec une syntaxe spécifique

Parfois encodé avec un seul paramètre

```
type 'a result =  
  | Ok of 'a  
  | Error of exn
```

Dans certains langages Result est remplacé par Either (Scala, Haskell), c'est un type plus générique qui par « convention » représente les erreurs dans la valeur Left

```
type ('a, 'b) either =  
  | Left of 'a  
  | Right of 'b
```



# MORE

## ACCUMULATEURS D'ERREURS

On souhaite parfois remonter l'ensemble des erreurs rencontrées (formulaire, json, csv, compilation, ...) : nous avons besoin d'accumulateurs d'erreurs !

**Problème** Result.t va contenir la première erreur qui sera propagée par bind ou map

On aimerait quelque chose qui ressemble à

```
type ('ok, 'error) result =  
  | Ok of 'ok  
  | Error of 'error list
```

*(ce type n'existe pas dans Stdlib c'est une illustration)*

**Avec OCaml Base (Stdlib alternative), il existe une fonction** combine\_errors

val combine\_errors : ('ok, 'err) [t](#) list -> ('ok list, 'err list) [t](#)

**... mais cela devient vite fastidieux à manipuler**