

# TAKE AWAY

## LES OPTIONS

A utiliser pour modéliser l'absence de valeur

Sécurisant, surtout quand on dispose d'ADT

Facile à manipuler avec syntaxe spécifique

*(let\* -> OCaml, let! -> F#, do notation -> Haskell, for comprehension -> Scala)*

... ou à défaut flatMap / bind

Dans certains langages Option s'appelle  
Optional (Java) ou Maybe (Scala, Haskell)



# MODÉLISER UNE ERREUR POTENTIELLE

## RESULT

```
type weapon = string
type target = string
type impacted = | Impacted of target

type ('a, 'b) result =
  | Ok of 'a
  | Error of 'b

let arm_your_bow : (weapon, exa) result = Error (Failure "not carried")
let targeted_monster : (target, string) result = Error "too far"
```