

TAKE AWAY

DU TAKE AWAY

Utilisez `Option.t` pour représenter la possibilité d'une absence de valeur

Utilisez `Result.t` pour représenter la possibilité d'une erreur

Utilisez `Base.Validate` si vous voulez un accumulateur d'erreur pour valider des données

Dans la majorité des cas ce que vous voulez c'est `Result.t` !!!



SIDE QUEST

Qualité du code

Raisonner sur un logiciel et à plus large échelle sur un SI est complexe

Jusqu'à présent nous nous sommes concentré sur des tactiques (ADT, Option/Result, PBT) pour résoudre des problèmes récurrents (machine états implicites, null pointer errors, exemples mal choisis, ...) ... mais spécifiques !

Quelles sont les stratégies d'architecture généralisables ?

