





**DRY**

**DON'T REPEAT YOURSELF**

MAGNET 2-DUALS - THOUGHTS

« Every piece of knowledge must have a single, unambiguous, authoritative representation within a system », *Andy Hunt*

Cela se traduit dans les logiciels par:

- L'utilisation d'abstractions: `class`, `interface`, `package` ... mais aussi `type`, `module`, `typeclass`, `trait`, `function`, `s-expression` ...
- La normalisation des données

INDÉPENDANTES

# DRY

## DON'T REPEAT YOURSELF

« Every piece of knowledge must have a single, unambiguous, authoritative representation within a system », *Andy Hunt*

Cela se traduit dans les logiciels par:

- L'utilisation d'abstractions: `class`, `interface`, `package` ... mais aussi `type`, `module`, `typeclass`, `trait`, `function`, `s-expression` ...
- La normalisation des données

INDÉPENDANT DES IDIOMES



# SOLID

## SINGLE-RESPONSIBILITY PRINCIPLE

« There should never be more than one reason for a `class` to change. »,  
*Robert C. Martin a.k.a Uncle Bob*

Peut être modifiée en

« There should never be more than one reason for an `abstraction` to change. »

Cela se traduit dans les logiciels par:

- Créer des abstraction qui ont un faible couplage entre elles
- Séparer les données et les comportements: *Visitor pattern, Iterators, Functional programming, Modular programming*