

# INTRODUCTION AUX CONCEPTS ORIENTÉS CRAFT



## LES KATAS



Avec cette série de mini-BDs, j'essaie de présenter des concepts orientés craft afin de les rendre accessibles aux non-initiés.



[in/jeremy-sorant](#)



[jsorant](#)



## DÉFINITION

Le terme “kata” vient des arts martiaux où les combattants répètent un très grand nombre de fois les mêmes coups et enchaînements.

Il s’agit d’exercices relativement simples que l’on peut accomplir en un quelques minutes ou dizaines de minutes. Certains vont demander d’implémenter un algorithme, d’autres de retravailler du code existant.



## MOTIVATIONS

En répétant ces exercices simples, je deviens à l'aise avec certains types de problèmes à résoudre et je les identifie plus rapidement au quotidien.

Je peux aussi travailler une méthodologie ou un outil en particulier comme un langage de programmation, un framework de tests, le TDD, un paradigme de programmation, l'utilisation de mon IDE...



## PARTAGE

Résoudre un kata à plusieurs permet de partager et de découvrir des pratiques et des approches.

Je pratique régulièrement des katas en pair ou en mob dans cette optique.



## DRIVER / NAVIGATOR

Les termes “driver” et “navigator” sont spécifiques au développement en pair ou en mob.

Le driver désigne la personne qui détient le clavier. Elle code en suivant les instructions d'une autre personne désignée comme le navigator.



## PERSONNALiSATIONS

Pour travailler certains aspects précis du développement logiciel, ou pour ajouter un peu de fun et de piment, je m'impose parfois une ou plusieurs contraintes lors de ma résolution de kata.

Ces contraintes me forcent à envisager ma pratique différemment et me permettent de découvrir de nouvelles méthodes de résolution.



## BABY STEPS

Cette contrainte consiste à faire l'effort de résoudre un kata en TDD en imaginant toujours la step la plus petite possible pour avancer.

Je travaille le découpage de problèmes en sous problèmes et j'utilise TDD pour me guider au maximum dans la résolution du kata.



## TEST 88 COMMIT || REVERT

Pour la contrainte “TCR”, je mets en place un timer court, par exemple 2 minutes. A chaque fois que le timer arrive à expiration, je dois revert le code au dernier commit. Je ne peut commit le code que si l’ensemble des tests est vert, et dans ce cas, je réinitialise le timer.

C’est une contrainte qui force l’utilisation de baby-steps.



## IMMUTABLE

Cette contrainte consiste à utiliser résoudre un kata sans utiliser d'état. Pour cela, aucun objet ne peut être modifié après sa création, aucune variable ne peut être réassignée et aucune collection ne peut être modifiée.

L'immutabilité permet de m'éviter des bugs liés aux états. Cette contrainte me permet d'explorer des stratégies pour la mettre en place efficacement au quotidien.



## PRIMITIVE OBSESSION

Pour cette contrainte, on va chercher à encapsuler tous les types “primitifs” de notre langage ainsi que les chaînes de caractères et les collections (number, boolean, string, list, map, set...).

Avec cette contrainte, je rends le code plus explicite et je limite les actions possibles sur chaque objet à celles nécessaires pour mon cas d’usage.



## TELL ! DON'T ASK !

Pour cette contrainte, aucune méthode ne peut renvoyer de valeur.

Ceci me force à respecter le paradigme orienté objet en demandant aux objets d'encapsuler leurs comportements .



## NO FLOW CONTROL

Cette contrainte interdit d'utiliser les instructions de contrôle de flux (if, switch, while). Les boucles for sont autorisées.

Ceci me force à utiliser le polymorphisme, les structures de recherches (list, map...) ou d'autres concepts afin de limiter le nombres de branches dans mon programme.



## PING PONG

Cette contrainte implique de faire le kata en pair.

On change les rôles après avoir écrit un test.

La première personne A écrit un test et passe le clavier à la seconde personne B qui le résout et fait le refactoring nécessaire. Puis B écrit le test suivant et passe le clavier à A. Et ainsi de suite...



## MUTE PING PONG

Cette contrainte impose les mêmes conditions que pour le “ping pong” avec une contrainte supplémentaire : il est interdit de parler, d'écrire ou de dessiner.

Je ne peux utiliser que le code pour communiquer, mais je n'ai pas le droit d'utiliser les commentaires à cet effet.



# NO MOUSE

Il est interdit d'utiliser la souris.

Cette contrainte me force à utiliser les raccourcis clavier et me permet de gagner en rapidité au quotidien.



## CATALOGUES

Voici les deux catalogues de katas que j'utilise :

[codingdojo.org](http://codingdojo.org)  
[kata-log.rocks](http://kata-log.rocks)

Jérémy Sorant  
ingénieur logiciel senior  
chez SHODO LYON



SHODO LYON

I'ESN craft militante  
#justicesociale