



Alexandre Cuva

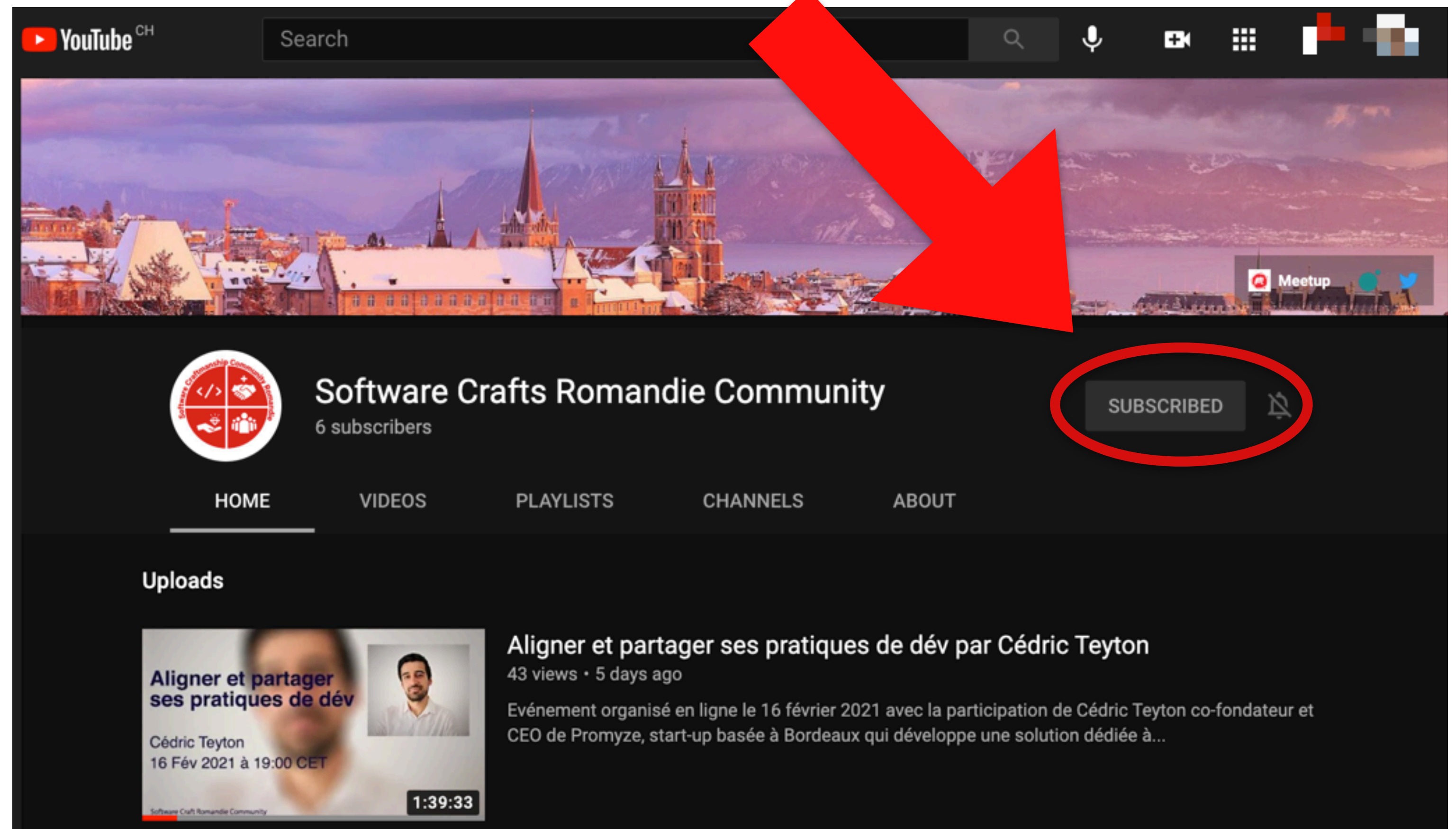
Amélioration de la lisibilité avec les les objets Calisthénique

SCR Community - Meetup Page



<https://www.meetup.com/Lausanne-Software-Craftsmanship-Meetup-Group/>

YouTube SCR



<https://www.youtube.com/channel/UCvc4OjUhLWUq0Ib9TFi2mgw>

Nos Sponsors





Alexandre Cuva

Amélioration de la lisibilité avec les les objets Calisthénique

Au sujet de l'orateur

Alexandre Cuva est un **coach agile entreprise** et **Agile Technique**, où il aide les équipes à adopter les pratiques Modernes de développement comme le **TDD**, **DDD** et **l'Event Storming**.

Il est cofondateur de SoCraAgile, SmartDev et SoCraDev.


+22 ans expérience internationale,
dont +15 ans en agilité



Connecté avec moi ou **suivez moi** sous LinkedIn

<https://www.linkedin.com/in/cuvaalex/>

REFACTORING

- Quand nous trouvons une duplication (Règle de )
- Quand nous brisons les règles de la Calisthénie des Objets.



Lesson 1

TRAINING PROGRAMME



LA RÈGLE DES 80-20 DU REFACTORING

80 % de la valeur du refactoring
provient de **l'amélioration de la
lisibilité..**

20% de la valeur restante provient de
la modification de la conception.



Lesson 1

TRAINING PROGRAMME

ALCOR
academy

Directives de refactoring

REFACTOR LA LISIBILITÉ AVANT LA CONCEPTION

De petites améliorations de la *lisibilité* du code peuvent améliorer radicalement sa *compréhension*.

Commencez par donner de *meilleurs noms* aux variables, méthodes et classes.

L'idée est **d'exprimer l'intention** plutôt que les détails de mise en œuvre. Nous recommandons l'approche *d'Arlo Belshee* pour l'attribution de noms.

<https://www.digdeeproots.com/articles/naming-as-a-process/>



Lesson 1

TRAINING PROGRAMME



Object Calisthenics

Le mot **calisthénie** vient des mots grecs anciens

*kalos (κάλλος), qui signifie "beauté" et
sthenos (σθένος), qui signifie "force".*



Lesson 4 & 5

TRAINING PROGRAMME



Object Calisthenics règles

- Un seul niveau d'indentation par méthode
- N'utilisez pas le mot-clé ELSE
- Enveloppez toutes les primitives et les chaînes de caractères
- Collections de première classe (enveloppez toutes les collections)
- Un seul point par ligne ~~dog.Body.Tail.Wag()~~ => `dog.ExpressHappiness()`
- Pas d'abréviations
- Gardez toutes les entités petites
[10 files per package, 50 lines per class, 5 lines per method, 2 arguments per method]
- Pas de classes comportant plus de deux variables d'instance
- Pas de getters/setters/properties publics



Lesson 4 & 5

TRAINING PROGRAMME



Object Calisthenics: POURQUOI?

Un seul niveau d'indentation par méthode

- aide à garder les méthodes concentrées sur une seule chose
- aide à réduire la taille des unités

N'utilisez pas le mot-clé ELSE

- aide à maintenir une seule voie d'exécution principale
- promouvoir l'utilisation du polymorphisme pour traiter les cas conditionnels complexes

Enveloppe toutes les primitives et les chaînes de caractères

- Rendre le code plus explicite en exprimant l'intention par des types qui deviennent des attracteurs de comportement



Lesson 4 & 5

TRAINING PROGRAMME



Object Calisthenics: POURQUOI?

Collections de première classe (envelopper toutes les collections)

- tout comportement spécifique à la collection sera attiré vers un seul endroit (filtering, sorting, joins, etc.)
- la modification de la représentation interne n'affectera pas les clients, ce qui améliore le découplage

Pas de getters/setters/properties

- s'aligne sur l'idée de la programmation OO comme un réseau d'entités qui collaborent en se transmettant des messages (en appelant des méthodes) les unes aux autres
- promouvoir l'encapsulation et une approche du type "dire, ne pas demander".



Lesson 4 & 5

TRAINING PROGRAMME



Object Calisthenics: POURQUOI?

Un point par ligne

- expose l'intention en cachant l'implémentation, une approche "dire, ne pas demander".
- réduit la quantité de connaissances que vous devez avoir pour permettre un comportement

N'abrégez pas

- si nous avons besoin d'un long nom, il est probable qu'il nous manque un concept : rendez-le explicite.
- Les abréviations peuvent prêter à confusion, il est donc préférable d'être clair.
- la frappe n'est pas le goulot d'étranglement

Garder des entités de petite taille

- Les petites classes se concentrent sur une seule chose, ce qui les rend plus faciles à comprendre et à réutiliser.



Lesson 4 & 5

TRAINING PROGRAMME



Object Calisthenics: POURQUOI?

Pas de classes avec plus de deux variables d'instance

- plus il y a de variables d'instance, plus la cohésion au sein de la classe est faible
- en général, les classes avec plus d'une variable d'instance sont des orchestrateurs, et les classes avec une seule variable d'instance sont des actionneurs : séparer les responsabilités

Heuristics

- ***Dites-le, ne demandez pas*** → Indiquer aux objets d'effectuer des actions au lieu de leur demander des données et de les traiter ailleurs. Similaire à l'absence de getters/setters.
- ***Loi de Déméter*** → Chaque composant ne doit parler qu'à ses amis ; ne pas parler aux inconnus. Similaire à un point par ligne.



Lesson 4 & 5

TRAINING PROGRAMME



KATA



Tennis refactoring by Emily Bache

Nettoyer le code de façon à ce que quelqu'un puisse le lire et le comprendre facilement. Vous pouvez trouver le repository ici :

<https://github.com/emilybache/Tennis-Refactoring-Kata/tree/master/typescript>



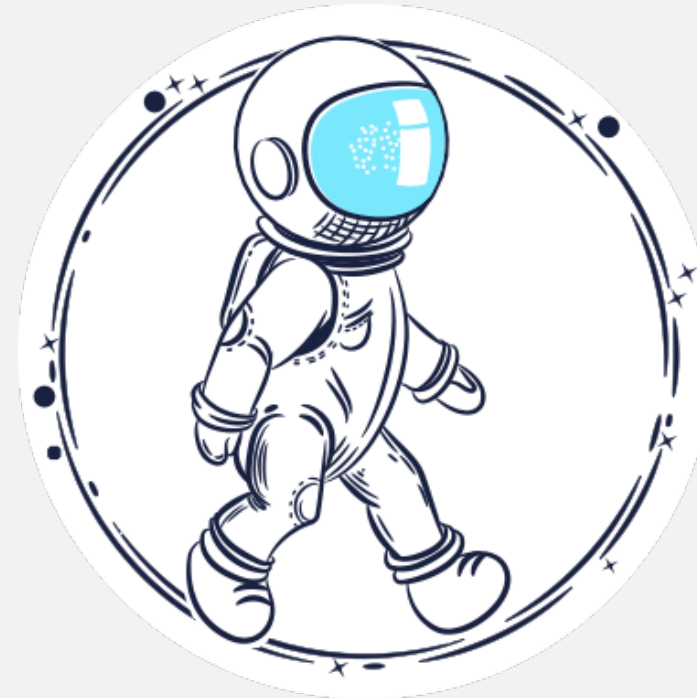
Lesson 1

TRAINING PROGRAMME



Excellence technique

En 3 étapes



MARCHE
COMPÉTENCES TECHNIQUES
Des bases d'automatisation



COURRIR
PRODUIRE ET LIVREE
Conception et architecture modulaires



VOLER
PEOPLE & CULTURE
Développement axé sur
le métier



Réserver un appel avec nous

Nous sommes basée à
Londres, UK et à Nyon, Suisse

Nous servons nos clients dans le
monde.

alcor.academy/booking
info@alcor.academy



ALESSANDRO DI GIOIA

alex@alcor.academy

[parajao](https://twitter.com/parajao)



MARCO CONSOLARO

marco@alcor.academy

[consolondon](https://twitter.com/consolondon)



ALEXANDRE CUVA

alex.cuva@alcor.academy

[cuvaalex](https://twitter.com/cuvaalex)

Merci

Contactez moi, si vous avez besoin de plus d'information

Alexandre Cuva
+41 78 715 83 09
alexandre.cuva@socraagile.ch

<https://www.linkedin.com/in/cuvaalex/>