

# Style de code du groupe Côté

Nous codons tous: en MATLAB, en C, en Python, en Objective-C, en Swift, en C++, en assembleur. L'acte de programmer est à la base de plusieurs des tâches que nous faisons, c'est donc pourquoi il doit être bien fait.

## Les règles

1. Le code doit se lire comme un texte.  
*Les noms de variables et les noms de fonctions sont importants. Une variable booléenne peut s'appeler "isDone". Un tableau représentant une image peut s'appeler "image". Une fonction peut avoir un verbe d'action dans son nom.*
2. On prend un style "camel-case" c'est à dire la première lettre est minuscule, et chaque mot est ensuite commencé par une majuscule. On ne prend pas de underscore (\_).
3. Une fonction ne devrait faire qu'une seule chose.
4. Juste parce que le code fonctionne ne veut pas dire que le code est bon.  
*Les tests de validité, le style, la simplicité, la facilité à maintenir le code sont toutes des caractéristiques essentielles d'une tâche bien faite.*
5. Il n'y a pas de pire commentaires que des commentaires qui ne sont pas à jour ou qui sont redondants.  
*Si le code se lit comme du texte, les commentaires ne sont pas nécessaires.*
6. Les *if-else* s'écrivent et s'indentent comme ceci, car ce style est compact:  

```
if (condition) {  
    // du code  
} else {  
    // d'autre code  
}
```
7. La meilleure fonction n'a pas d'argument. La deuxième meilleure fonction n'a qu'un seul argument. La troisième meilleure fonction n'a que deux arguments. Il n'y a pas de quatrième meilleure fonction.
8. Du code sans test c'est du code incomplet. MATLAB a des UnitTest. Python a unittest. Swift et Objective-C ont XCTest, C++ a plusieurs librairies de Unit Testing. Tous les langages ont maintenant des librairies de tests.  
*Écrire des tests permet de mieux comprendre son propre code et de mieux le factoriser, en plus de le rendre plus fiable. Il permet de se lancer dans des grandes modifications sans avoir peur de briser quelque chose.*
9. L'utilisation d'un système de "versioning" comme subversion, CVS, ou Git est essentielle.  
*Le versioning permet de modifier le code avec assurance et permet de redonner à un collaborateur le code tel qu'il était en date d'une publication. Le groupe dcclab possède son serveur de Version (subversion) sur le serveur.*
10. Lorsqu'on travaille et qu'on "commit" le code, on tente de regrouper nos modifications par sous-projet et on commit tous les fichiers ensemble.