

# Initiation à l'algorithmique / Qu'est-ce que le codage ?

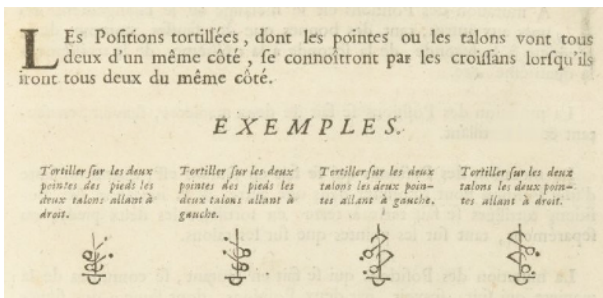
## Introduction

Un algorithme est une succession d'instructions organisées de manière ordonnée, permettant de résoudre certains types de problèmes.

Le codage est une façon de représenter des instructions, dans un langage spécifique :

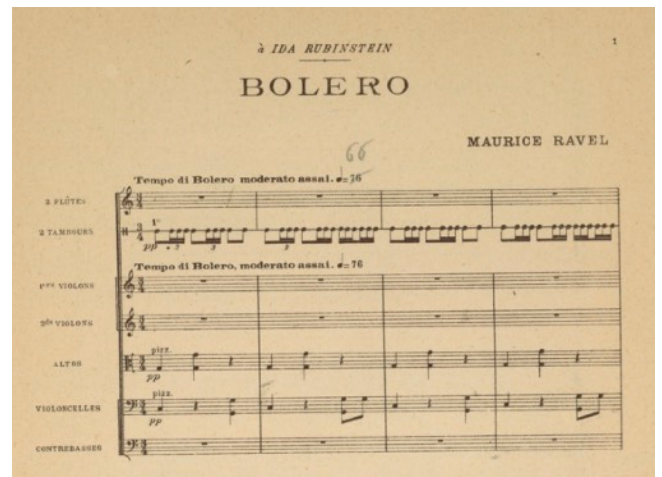
Exemples de codage :

- la notation musicale (sous forme de partition : il en existe plusieurs langages).
- la notation chorégraphique (il en existe plusieurs langages).
- le codage informatique (il en existe de nombreux langages ! Scratch, Python, Swift, C++...)
- le plus simple : le « pseudo-code » : succession d'instructions en langage clair, avec des verbes à l'infinitif (« Peser 100 g de farine », « Verser la farine dans le bol »...)

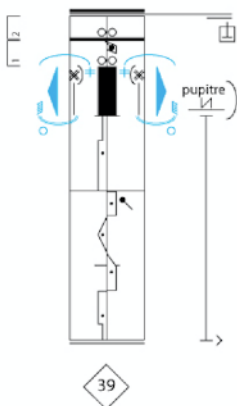


Extrait de *Chorégraphie, ou l'Art de décrire la danse par caractères, figures et signes démonstratifs* de M. Feuillet et Raoul Auger (1701)

« Source gallica.bnf.fr / Bibliothèque nationale de France »



Début de la partition du *Boléro* de Maurice Ravel (1929)



« Partition chorégraphique » en cinématographie  
Laban. Extrait de *Sei solo*  
création 2009 de Raphaël Cottin  
(source Wikimedia Commons)

*Sei solo - corrente*  
chorégraphie et cinématographie : Raphaël Cottin / violon : Hélène Schmitt  
compagnie La Poétique des Signes - extrait de la création 2009  
musique : 1<sup>er</sup> partita pour violon seul de J. S. Bach - courante

exemples de code en langages

Python (source : [fr.wikipedia.org](http://fr.wikipedia.org))

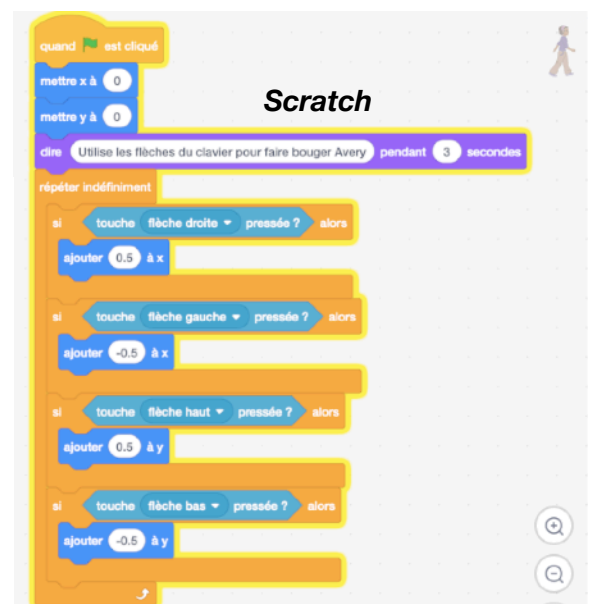
Swift (source : application Swift Playgrounds)

Scratch

La liste des nombres pairs :

```
liste = [entier for entier in range(10) if entier % 2 == 0]
# liste = [0, 2, 4, 6, 8]
```

Python



Un codage ne correspond pas nécessairement à un algorithme ! (par exemple, une partition de musique ou de danse est une forme codée, mais qui ne correspond pas à un algorithme).

Mais tout algorithme peut être codé, souvent de différentes façons (diagramme logique, code informatique...)

Swift

```
moveForward()
moveForward()
turnLeft()
moveForward()
collectGem()
moveForward()
turnLeft()
moveForward()
moveForward()
toggleSwitch()
```

Beaucoup de choses peuvent être décomposées en une succession d'instructions simples (la Macarena, une posture de yoga, la recette du gâteau au yaourt...).

Tout code nécessite la définition, en préambule, de certaines choses indispensables à son exécution :

- dans le cas d'une recette, il faut d'abord définir la liste des ingrédients et des accessoires et récipients nécessaires.
- dans le cas d'une danse, on **initialisera** le code en précisant quelle posture adopter avant d'exécuter les instructions (commencer une Macarena accroupi ?... pas pratique ! On *intialisera* donc le codage de la danse en précisant qu'il faut se tenir debout, les bras le long du corps, jambes légèrement écartées).
- dans un code informatique, on définira des variables, on précisera la famille (bibliothèque) d'instructions utilisée, etc.

### Un exemple : **À FAIRE**

1. Écrire ci-dessous en « pseudo-code », la succession d'instructions permettant d'effectuer un brossage de dents (on s'arrêtera à la partie « brossage » seule, sans aller jusqu'au rinçage de la bouche !).
2. Ensuite, faire **exécuter** ce code par votre binôme, pour détecter les erreurs ou les oublis : c'est le **débogage** du code (appelé aussi débogage — un anglicisme). Corriger ces erreurs et oublis, sur la feuille.



# Créer un programme pour faire des calculs de vitesse (voire de distance, ou de durée...)

Pour le codage, on utilisera la version en ligne de Scratch 3, à l'adresse <https://scratch.mit.edu/projects/editor/> ou le logiciel « Scratch desktop » à installer sur son ordinateur (<https://scratch.mit.edu/download>)

## Première étape – analyser une résolution de calcul et en tirer un pseudo-code

### Partie 1

À partir du calcul résolu et corrigé (document fourni), écrire en pseudo-code, sur une feuille à part, la succession d'instructions permettant de recueillir les données nécessaires et de faire le calcul demandé (Question 1 de l'exercice).

Adapter si besoin le pseudo-code pour qu'il permette de réaliser *n'importe quel* calcul du même type.

### Partie 2

Définir le cahier des charges du programme que vous réaliserez, en fonction de ce que vous estimez capable de réaliser. Faire un choix parmi les niveaux 2 et 3 proposés ci-dessous.

## Cahier des charges du programme

Définis le cahier de charges de ton programme en cochant la case correspondant aux critères que tu choisis de retenir, selon le niveau de difficulté et ce que tu te sens en mesure de faire — tu peux cocher deux cases si tu envisages de faire deux programmes de complexités différentes.




Difficulté pour le programmeur ou la programmatrice	Intérêt pour les utilisateurs du programme	critère	cocher l'option choisie
★	😞	l'utilisateur doit saisir la distance en mètre	
		l'utilisateur doit saisir la durée en seconde	
		on affiche la valeur de la vitesse uniquement en m/s	
★★	😊	l'utilisateur doit saisir la distance en kilomètre	
		l'utilisateur doit saisir la durée en minute	
		on affiche la valeur de la vitesse uniquement en km/h	
★★★	😄	l'utilisateur peut saisir la distance en mètre ou en kilomètre	
		l'utilisateur saisit la durée en heure, minute, seconde	
		on affiche la valeur de la vitesse en m/s et en km/h	
★★★★	😄🤖	l'utilisateur choisit s'il veut calculer v, d ou t	
		l'utilisateur dispose de choix pour les unités des valeurs saisies	
		le programme affiche le résultat dans une unité qui le rend facile à comprendre	

**Attention : ce qui est plus facile à programmer sera plus pénible à utiliser !**

Par exemple, la personne qui utilisera le programme appréciera de ne pas avoir à convertir la durée du parcours en seconde : (un peu) plus complexe à programmer, mais beaucoup plus simple à utiliser !

## Avant la deuxième étape...

Regarder attentivement les vidéos en ligne dont les liens sont donnés ci-dessous (en texte et QR-code)

Les variables dans Scratch	Les capteurs	Faire des calculs avec les opérateurs
durée : 4 min 53	(jusqu'à 1 min 30 et à partir de 7 min 55)	durée : 6 min 45
<a href="https://youtu.be/hnbfxoiMFQ0">https://youtu.be/hnbfxoiMFQ0</a>	<a href="https://youtu.be/8c-taygQcN8">https://youtu.be/8c-taygQcN8</a>	<a href="https://youtu.be/5gvdhD_eFFQ">https://youtu.be/5gvdhD_eFFQ</a>
		

## Deuxième étape – Préparer la création du programme

Dans un premier temps, bien définir le cahier des charges de ton programme (tableau page précédente).

Écrire un nouveau « pseudo-code » détaillé permettant de répondre au cahier des charges défini.

À partir de ce pseudo-code, recenser dans le tableau ci-dessous toutes les variables que devra utiliser le programme que tu vas réaliser. Indiquer pour chacune de ces variables s'il s'agit d'un nombre ou de texte (même si Scratch s'en moque).

**Conseil : donner à chaque variable un nom qui évoque ce qu'elle représente.**

nom de la variable	à quoi correspond-elle ?	est-ce un nombre ? un texte ?

## Troisième étape – Réaliser le programme

Réalise, dans Scratch, le programme qui permettra à l'utilisateur d'obtenir la vitesse moyenne, après avoir renseigné la distance parcourue et la durée du parcours.

### Et au-delà...

Planifier puis réaliser un programme qui permette de **calculer la distance, ou la durée, ou la vitesse moyenne, à partir des deux autres grandeurs.**

La notion de « message » dans Scratch pourra être utile pour cela.

**Les événements et les messages** (durée 12 min 51)

<https://youtu.be/QKoXyUk5Hel>

Ou simplement les questions, avec des variables pour enregistrer les réponses !

