



# Algorithmie

01 – Introduction à l'algorithmie

**DWWM PE6 LYON**

Barthélémy DELUY – CC-BY-NC-SA 4.0

Formation : **Développeur** Web et Web Mobile

Algorithmie : la **base** du développement

=> Ce module est **LE PLUS IMPORTANT** de votre formation

Un algorithme ?



Définition Wikipédia :

Un algorithme est une suite finie et non ambiguë d'instructions et d'opérations permettant de résoudre une classe de problèmes

analogies : recettes de cuisine, notice de montage

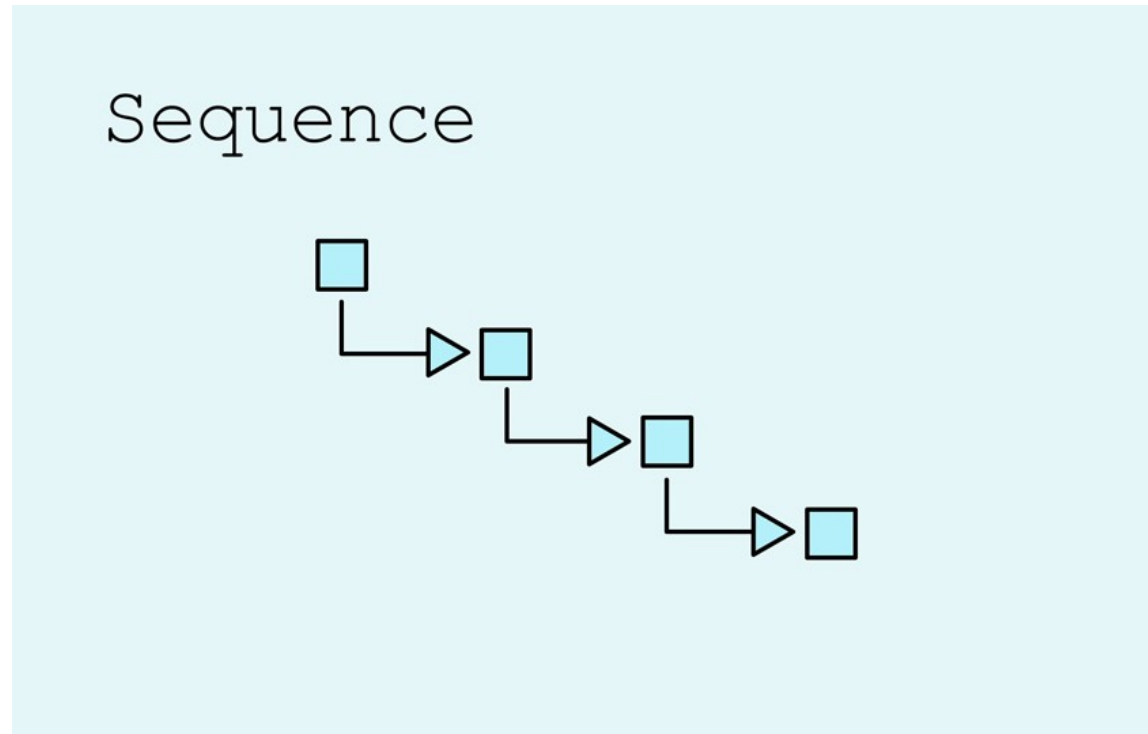
# Sommaire

1. La programmation séquentielle
2. Les mots-clés de l'algorithmie
3. Contrôle de flux 1 : Les branchements conditionnels simples
4. Contrôle de flux 2 : Les boucles TANT QUE

# 1. La programmation séquentielle

1/4

Le programme commence à la première instruction, et s'exécute séquentiellement jusqu'à la dernière :



# 1. La programmation séquentielle

2/4

ATTENTION : un ordinateur est profondément stupide. Il a besoin d'instructions exactes et précises sur ce qu'il doit faire ; il n'est pas capable de deviner ou de faire des hypothèses.

Ici, Josh joue particulièrement bien le rôle de la machine :

[https://www.youtube.com/watch?v=cDA3\\_5982h8](https://www.youtube.com/watch?v=cDA3_5982h8)

# 1. La programmation séquentielle

3/4

Première mise en pratique !

Choisissez l'un de ces thèmes, et rédigez les instructions en mode ELI5 qui permettent de réaliser le traitement décrit :

- Vous venez de vous réveiller, il faut préparer un café/thé/chocolat chaud
- Vous venez d'acheter un nouveau pc fixe, il faut l'installer physiquement sur votre bureau avec 2 écrans, un clavier et une souris sans fil, etc
- Vous recevez de la famille (8 personnes + 4 enfants) à dîner pour une occasion spéciale, il faut mettre la table au salon avec tout le nécessaire
- Vous êtes caissier d'un fast-food, vous recevez la commande d'un client et devez préparer son plateau



# 1. La programmation séquentielle

4/4

Pour votre thème, vous devez donc rédiger :

- La situation de départ dans laquelle vous êtes (les hypothèses que vous formulez)
- La suite d'instructions précises permettant d'arriver au résultat

Trouvez un binôme qui a choisi un thème différent du votre ; il jouera le rôle de Josh pour vous, et vous jouerez le rôle de Josh pour lui.

Faites autant d'itération que nécessaire pour arriver à un algo complet.

## 2. Les mots-clés de l'algorithmie

1/4

Il y a une certaine façon de présenter un algorithme :

- Tout d'abord, l'algo commence par le mot-clé « PROGRAMME » suivi du nom de votre programme
- À la ligne suivante, on trouve le mot-clé « DÉBUT »
- À la fin de l'algo, on trouve le mot-clé « FIN »
- Entre les mots-clés « DÉBUT » et « FIN », toutes les instructions sont *indentées*, c'est-à-dire décalées d'une tabulation vers la droite

## 2. Les mots-clés de l'algorithmie

2/4

```
PROGRAMME Saluer  
DÉBUT  
    Dire « Bonjour »  
    Serrer la main  
FIN
```

## 2. Les mots-clés de l'algorithmie

3/4

Il est d'usage de remplacer les actions triviales (souvent incorporées au langage) ou hors de portée de l'algorithme par un verbe les décrivant :

- Saisir, pour demander une saisie utilisateur
- Afficher, pour afficher à l'écran
- Récupérer\_messages\_twitter
- EnregistrerEnBase

## 2. Les mots-clés de l'algorithmie

4/4

Il est d'usage de remplacer les actions triviales (souvent incorporées au langage) ou hors de portée de l'algorithme par un verbe les décrivant :

- Saisir, pour demander une saisie utilisateur
- Afficher, pour afficher à l'écran
- Récupérer\_messages\_twitter
- EnregistrerEnBase

### 3. Les branchements conditionnels

1/5

Il est très rare qu'un programme s'exécute séquentiellement de la première à la dernière instruction d'une traite.

Dans la très large majorité des cas, il est nécessaire de ne réaliser certaines parties du traitement que sous certaines conditions : faire le plein si le réservoir est vide, baisser le feu si le lait déborde, etc.

On va dans cas utiliser la syntaxe :

```
SI <condition> ALORS  
    [Instructions]  
FINSI
```

# 3. Les branchements conditionnels

2/5

La condition doit **systematiquement** pouvoir être évaluée comme étant *VRAIE* (true) ou *FAUSSE* (false) => la condition correspond à une question dont la réponse est « oui » ou « non ».

Il est possible de cumuler plusieurs conditions simples à l'aide des mots-clés **ET** et **OU**, pour construire des conditions complexes :

SI la viande est cuite ET l'accompagnement est chaud ALORS  
    Dresser l'assiette  
FINSI

### 3. Les branchements conditionnels

3/5

A	B	A ET B	A OU B
F	F	F	F
F	V	F	V
V	F	F	V
V	V	V	V

Attention : il existe une priorité entre le ET et le OU, comme en maths !  
Le ET est prioritaire sur le OU :

$$A \text{ ET } B \text{ OU } C \iff (A \text{ ET } B) \text{ OU } C$$

Il existe également l'opérateur NON qui permet d'inverser une condition :  
SI NON(Utilisateur connecté) ALORS Afficher « Accès limité ! » FIN SI



### 3. Les branchements conditionnels

4/5

Il est possible d'utiliser le mot-clé SINON pour fournir un bloc alternatif d'instructions, au cas où la condition du SI ne serait pas satisfaite :

SI Utilisateur connecté ALORS

Afficher profil

SINON

Afficher formulaire de connexion

FINSI

# 3. Les branchements conditionnels

5/5

Mise en application !

Reprenez l'algorithme de votre thème (ou prenez un autre thème), et ajoutez des contrôles conditionnels.

## 4. Les boucles TANT QUE

1/3

Il est régulièrement nécessaire de répéter une ou plusieurs instructions pour obtenir le résultat escompté.

Il est tout à fait possible de copier-coller les instructions si vous connaissez le nombre de répétitions au moment d'écrire le programme.

Dans le cas contraire, il faut utiliser une boucle TANT QUE

## 4. Les boucles TANT QUE

2/3

```
TANT QUE <condition> FAIRE  
    <instructions>  
FIN TANTQUE
```

Comme pour le SI, la condition doit forcément s'évaluer à « VRAI » ou « FAUX »

## 4. Les boucles TANT QUE

3/3

Reprenez votre thème, et ajoutez la gestion des répétitions.

# Conclusion

Vous disposez dorénavant de toutes les briques nécessaires pour faire des traitements automatiques.

Nous verrons cet après-midi les variables, procédures, et fonctions, qui vous permettront d'ajouter un peu de dynamisme à vos algorithmes, et de pouvoir résoudre tous les problèmes qui se présenteront à vous.

