

INTRODUCTION AUX ALGORITHMES

Qu'est ce que la programmation ?

Qu'est ce qu'un programme

Un programme est un ensemble d'instructions (ou d'étapes) destinées à être exécutées par un ordinateur.

Une instruction dicte à l'ordinateur l'action qu'il doit effectuer avant de passer à l'instruction suivante.

Qu'est-ce qu'un algorithme ?

“Un algorithme permet de décomposer un problème complexe en une suite d'opérations simples.”

3 principaux types d'actions d'un algorithme

1. Je prends une bouilloire
2. J'ajoute de l'eau
3. J'allume la bouilloire
4. Tant que l'eau ne bout pas
5. J'attends
6. Dès que l'eau bout, je verse l'eau dans le mug avec le thé
7. Tant que le thé n'est pas infusé
8. J'attends
9. Dès que le thé est infusé, j'enlève le sachet de thé
10. Si je préfère le thé sucré alors j'ajoute du sucre
11. Tant que le thé n'est pas assez sucré
12. J'ajoute du sucre
13. Je goûte
14. Si je préfère le lait au citron alors
15. J'ajoute du lait
16. Sinon
17. J'ajouter du citron

- **Actions simples**
- **Actions qui s'exécutent sous une condition**
- **Actions qui se répètent**

Trouver les actions simples, conditionnelles et qui se répètent

1. Je demande à mon programme de sonner à 7h du matin.
2. Mon téléphone sonne;
3. Tant que je n'appuie pas sur un bouton d'arrêt, la sonnerie se répète en boucle
4. Si j'appuie sur le bouton snooze, l'alarme se re-déclenchera dans 5 min, sinon si j'appuie sur le bouton STOP, l'alarme s'arrête définitivement

Solution

1. J'active une alarme à un moment de la journée
2. Mon téléphone sonne;
3. Tant que je n'appuie pas sur un bouton d'arrêt, la sonnerie se répète en boucle
4. Si j'appuie sur le bouton snooze, l'alarme se re-déclenche dans 5 min, sinon si j'appuie sur le bouton STOP, l'alarme s'arrête définitivement

Comprendre les langages informatiques

Qu'est-ce qu'un langage informatique

Combien de langues connaissez-vous ? Sur Terre, il existe des milliers de langues qui nous permettent de communiquer.

Pour communiquer à un ordinateur des instructions qu'il devra exécuter par la suite, il vous faudra aussi utiliser une langue, mais une langue qu'il comprend. C'est ce qu'on appelle un **langage informatique**, ou du code. Et il en existe des milliers. (https://fr.wikipedia.org/wiki/Liste_de_langages_de_programmation)

Mais rassurez-vous, comme pour l'anglais, l'espagnol ou l'arabe, certaines langues sont plus populaires, et plus utilisées que d'autres. Chaque langage informatique est composé de sa propre syntaxe, ou règle, et sémantique ou signification.

Différences de syntaxe entre 2 langages humains

Prenons par exemple deux mots : " Bonjour ! ", et " ¡ Hola ! ". Ces deux mots sont différents, mais ils ont la même signification. On note qu'en espagnol, on utilisera deux points d'exclamation, à l'endroit et à l'envers, contrairement à un seul en français

C'est une règle de syntaxe différente entre ces deux langues. Il en va de même pour les langages informatiques.

Langues	Syntaxe
Français	Bonjour !
Espagnol	¡ Hola !

Différences de syntaxe entre 2 langues informatiques

Prenons deux langages informatiques : JavaScript et PHP.

Ces deux lignes de code font strictement la même chose. Ils affichent le texte "Bonjour à tous" sur une page.

Avez-vous remarqué la différence de syntaxe entre ces deux langages ? En JavaScript, on utilise `document.write`, parenthèses suivies du texte entre guillemets. Alors qu'en PHP, on utilise le mot `echo` suivi du texte "Bonjour à tous", entre guillemets.

Langages	Syntaxe
Javascript	<code>document.write("Bonjour à tous");</code>
PHP	<code>echo "Bonjour à tous";</code>

Qu'est-ce que le langage machine ?

Pourquoi existe-t-il autant de langages informatiques différents, surtout s'ils font la même chose ?

La réponse est très simple. Chaque langage a ses propres forces et faiblesses.

Certains sont idéaux pour des petits programmes, tandis que d'autres ont été conçus pour gérer des calculs mathématiques complexes.

Quel que soit le langage que vous choisissiez, ce langage sera en réalité décomposé en un autre langage que seuls les ordinateurs peuvent comprendre : **le langage machine**.

Le langage machine est la langue la plus proche de l'ordinateur, c'est du binaire : une succession de 0 et de 1, parfaitement incompréhensible pour l'humain. C'est pour cela que les langages de programmation ont été créés.



L'objectif final d'un langage informatique est donc d'être un intermédiaire entre notre langage à nous, le langage humain, et le langage machine, que parle l'ordinateur.

Plus un langage est bas niveau, plus il se rapproche du langage machine, ce qui le rend plus difficile à comprendre pour les développeurs.

Plus le langage est haut niveau, plus il se rapproche des langages humains, c'est-à-dire qu'il sera composé de mots-clés, d'une structure et d'une syntaxe plus facile à comprendre et surtout à apprendre.

Il est important de noter que chaque langage de programmation possède une documentation, qui est un peu comme le " mode d'emploi " du langage.

Documentation Javascript :

<https://developer.mozilla.org/fr/docs/Web/JavaScript>

LANGAGE HUMAIN

Langage de haut niveau
Plus simple et plus éloigné du fonctionnement de la machine

- Java
- C# .NET
- Python
- Ruby...

Langage de bas niveau
Plus complexe et plus proche du fonctionnement de la machine

- C
- C++
- Objective-C...

011010101100
011101010101
101011101001
100010101010
100111101010

- Assembleur

LANGAGE MACHINE (Binaires)

Environnement de développement

Langages	Extension du fichier
Javascript	.js
Python	.py
PHP	.php
Java	.java
Ruby	.rb
C	.c
C++	.cpp

Comprendre la compilation et
l'interprétation d'un code

Trois façons de traduire du code source

- Compiler, c'est-à-dire le traduire en binaire
- Interpréter, c'est-à-dire le lire en temps réel et exécuter les instructions
- Hybride (compiler + interpréter)

Quelle est la différence entre compiler et interpréter ?

Prenons un exemple. Vous essayez de monter une maquette, mais la notice est dans une langue que vous ne comprenez pas. Vous pouvez, bien sûr, traduire la notice avant. Dans ce cas, c'est un langage compilé.

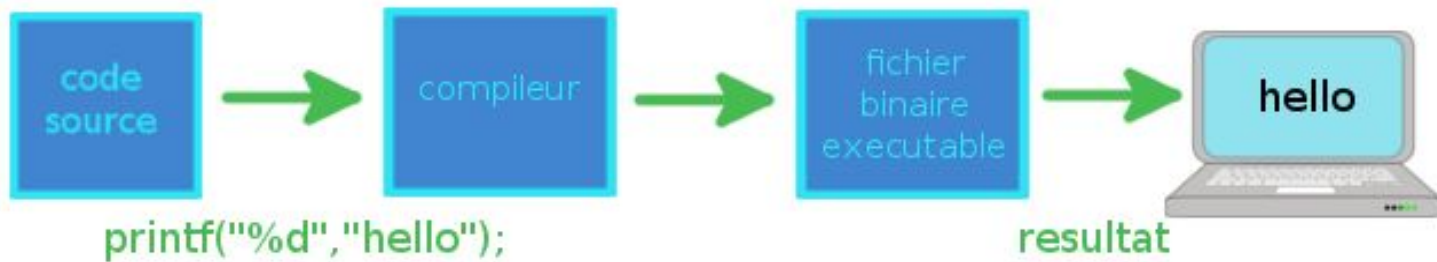
Mais vous pouvez aussi demander à quelqu'un qui parle cette langue de lire en traduisant la notice pendant que vous montez le modèle. Dans ce cas, c'est un langage interprété.

Langage compilé

Si un langage est de type compilé, il aura besoin d'un compilateur pour être traduit.

Les compilateurs traduisent le code en langage machine. Une fois cette traduction effectuée, vous pouvez distribuer le fichier résultant sur tous les postes des utilisateurs.

Exemple : C, C++ et Objective-C



Langage compilé

Avantages :

- Programme immédiatement disponible à démarrer
- Plus “rapide”, car optimisé
- Code source est privée

Inconvénients :

- Non multi-plateforme
- Nécessite des étapes supplémentaires pour tester

Langage interprété

Concernant les langages interprétés comme PHP ou JavaScript, ils auront besoin d'un interpréteur.

Les interpréteurs informatiques, quant à eux, traitent votre code source à chaque exécution, ligne par ligne, et c'est à l'autre utilisateur d'avoir l'interpréteur nécessaire disponible sur sa machine

Exemple : PHP et JavaScript



Langage interprété

Avantages :

- Multi-plateforme
- Simple à tester
- Facile à débbugger
- Code source public (Communauté open source)

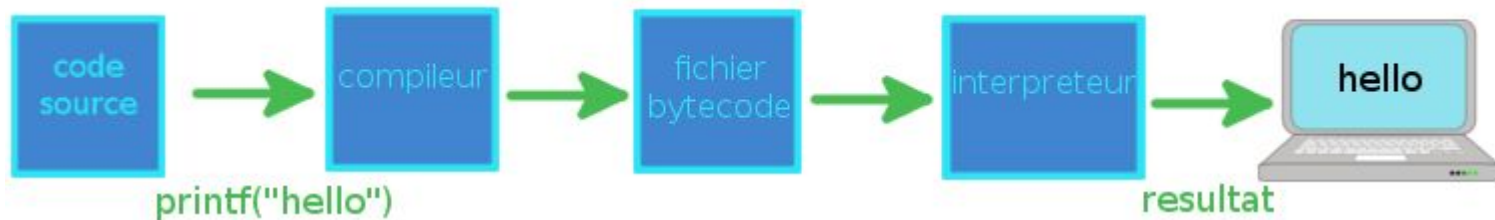
Inconvénients :

- Requier un interpréteur
- Pour chaque exécution, le programme doit être interprété préalablement

Langage hybride

Certains langages sont considérés comme hybrides car ils requièrent d'être préalablement compilés pour ensuite être interprétés

Exemple : Java, C# et Python



Langage hybride

Avantages :

- Multi-plateforme
- Simple à tester
- Facile à débbugger
- Code source compilé seulement si version différente de version compilé
- Code source public (Communauté Open Source)

Inconvénients:

- Requiert un interpréteur

Déboguer du code

Erreurs courantes en programmation

- Erreur de syntaxe
- Erreur sémantique (ou erreur de logique)
- Erreur d'exécution

Erreur de syntaxe

Les règles du langage (sa syntaxe) n'ont pas été respecté

Erreur de syntaxe

Bonne syntaxe

```
console.log("Hello World");
```

Mauvaise syntaxe

```
console.log("Hello World')
```

```
console.log("Hello World";
```

```
Console.log("Hello World");
```

```
consolelog("Hello World");
```


Erreur sémantique (ou de logique)

La sortie du programme n'est pas celle que vous attendiez

Erreur sémantique (ou de logique)

Code correct

```
let prenom = "Bob";  
  
alert( "Hello " + prenom );
```

Résultat : Hello Bob

Code avec erreur sémantique

```
let prenom = "Bob";  
  
alert( "Hello prenom" );
```

Résultat : Hello prenom

Erreur d'exécution (ou Runtime error)

Erreur de programme qui se produit pendant l'exécution du programme. L'ordinateur n'a pas pu exécuter une partie de votre code.

Exemple:

- Mémoire insuffisante
- votre programme essaie de lire un fichier qui n'existe plus
- Erreur de calcul

Erreur d'exécution (ou Runtime error)

```
let i = 1;  
  
while ( i < 10 ) {  
    console.log( i );  
}
```

Résultat: Boucle infini