

Sommaire

Général

Le niveau d'un langage de programmation

Le paradigme d'un langage de programmation

Différences Compilateur/Interpréteur

C

C++

Java

C#

Python

PHP

Visual Basic .NET

JavaScript

Assembleur

Présentation générale des langages de programmation

- ▶ Les langages de programmation sont en quelque sorte une manière de s'adresser à une machine avec une écriture compréhensible par l'humain

Présentation générale des langages de programmation

- ▶ Les langages de programmation sont en quelque sorte une manière de s'adresser à une machine avec une écriture compréhensible par l'humain
- ▶ Dans les faits, ils permettent d'écrire des algorithmes mathématiques dans une syntaxe plus ou moins proche de l'algèbre

Présentation générale des langages de programmation

- ▶ Les langages de programmation sont en quelque sorte une manière de s'adresser à une machine avec une écriture compréhensible par l'humain
- ▶ Dans les faits, ils permettent d'écrire des algorithmes mathématiques dans une syntaxe plus ou moins proche de l'algèbre
- ▶ Ils sont ensuite compilés ou interprétés pour être lus et compris par l'ordinateur

Présentation générale des langages de programmation

- ▶ Les langages de programmation sont en quelque sorte une manière de s'adresser à une machine avec une écriture compréhensible par l'humain
- ▶ Dans les faits, ils permettent d'écrire des algorithmes mathématiques dans une syntaxe plus ou moins proche de l'algèbre
- ▶ Ils sont ensuite compilés ou interprétés pour être lus et compris par l'ordinateur
- ▶ Le compilateur/interpréteur transforme le code écrit par le développeur en langage machine (le binaire), compris nativement par la machine

Le niveau d'un langage de programmation

- L'on classe les langages de programmation par "niveaux"

Le niveau d'un langage de programmation

- ▶ L'on classe les langages de programmation par "niveaux"
- ▶ Ce niveau représente l'éloignement entre le langage réel de la machine (le binaire) et les constructions qu'on lui superpose.

Le niveau d'un langage de programmation

- ▶ L'on classe les langages de programmation par "niveaux"
- ▶ Ce niveau représente l'éloignement entre le langage réel de la machine (le binaire) et les constructions qu'on lui superpose.
- ▶ Ainsi, un langage de bas niveau (comme l'assembleur) sera très proche du langage machine et donc bien moins compréhensible par un humain

Le niveau d'un langage de programmation

- ▶ L'on classe les langages de programmation par "niveaux"
- ▶ Ce niveau représente l'éloignement entre le langage réel de la machine (le binaire) et les constructions qu'on lui superpose.
- ▶ Ainsi, un langage de bas niveau (comme l'assembleur) sera très proche du langage machine et donc bien moins compréhensible par un humain
- ▶ A contrario, un langage de haut niveau (comme le Java) sera bien mieux compris par l'homme mais comportera de nombreuses couches d'abstraction qui l'éloigneront du langage machine (ce qui peut notamment poser des problèmes d'optimisation sur des dispositifs embarqués et/ou peu puissants)

Le paradigme d'un langage de programmation

- ▶ Chaque langage de programmation possède un ou plusieurs paradigmes

Le paradigme d'un langage de programmation

- ▶ Chaque langage de programmation possède un ou plusieurs paradigmes
- ▶ Le paradigme détermine la manière dont le programme va être pensé et rédigé

Le paradigme d'un langage de programmation

- ▶ Chaque langage de programmation possède un ou plusieurs paradigmes
- ▶ Le paradigme détermine la manière dont le programme va être pensé et rédigé
- ▶ Par exemple, le paradigme impératif décrit une suite d'instructions (d'actions que le programme va exécuter) qui se déroulera de haut en bas, et que rien ne pourra entraver ou modifier

Le paradigme d'un langage de programmation

- ▶ Chaque langage de programmation possède un ou plusieurs paradigmes
- ▶ Le paradigme détermine la manière dont le programme va être pensé et rédigé
- ▶ Par exemple, le paradigme impératif décrit une suite d'instructions (d'actions que le programme va exécuter) qui se déroulera de haut en bas, et que rien ne pourra entraver ou modifier
- ▶ D'autres paradigmes existent, tel le paradigme événementiel, où des petites séquences impératives seront exécutées en réaction à des actions prédéfinies ; Ou bien l'orienté objet, où l'on déclare des objets qui interagissent ensemble, etc...

Différences Compilateur/Interpréteur

- ▶ Un compilateur est un programme qui va transformer le code écrit dans un langage donné (par exemple : le C) en une séquence d'instructions binaires qui sera exécutée par le processeur.

Différences Compilateur/Interpréteur

- ▶ Un compilateur est un programme qui va transformer le code écrit dans un langage donné (par exemple : le C) en une séquence d'instructions binaires qui sera exécutée par le processeur.
- ▶ Un interpréteur est un programme qui va lire le code écrit (par exemple : en Python) pour le traduire à la volée en séquences d'instructions binaires compréhensible par la machine.

C



- ▶ Première version : 1972
- ▶ Auteur : Dennis Ritchie
- ▶ Paradigmes : Impératif, Procédural, Structuré
- ▶ Utilisation : Généraliste
- ▶ Niveau : Haut (Compilé)
- ▶ Popularité : 2^{ème} langage le plus utilisé en 2016 (16%)

C++



- ▶ Première version : 1983
- ▶ Auteur : Bjarne Stroustrup
- ▶ Paradigmes : Générique, Objet, Procédural
- ▶ Utilisation : Généraliste
- ▶ Niveau : Haut (Compilé)
- ▶ Popularité : 3^{ème} langage le plus utilisé en 2016 (7%)

Java



- ▶ Première version : 23 mai 1995
- ▶ Auteur : Sun Microsystems
- ▶ Paradigmes : Objet, Structuré, Impératif, Fonctionnel, Réfléxif
- ▶ Utilisation : Généraliste/Web
- ▶ Niveau : Très haut (Pré-Compilé : D'abord compilé en ByteCode, puis interprété par la JVM)
- ▶ Popularité : 1^{er} langage le plus utilisé en 2016 (21%)

C#



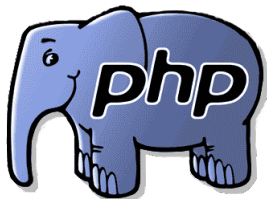
- ▶ Première version : 2000
- ▶ Auteur : Microsoft
- ▶ Paradigmes : Objet, Structuré, Impératif
- ▶ Utilisation : Généraliste/Web
- ▶ Niveau : Haut (Compilé)
- ▶ Popularité : 4^{ème} langage le plus utilisé en 2016 (5%)

Python



- ▶ Première version : 1990
- ▶ Auteur : Guido van Rossum
- ▶ Paradigmes : Objet, Impératif, Fonctionnelle
- ▶ Utilisation : Généraliste/Web
- ▶ Niveau : Très haut (Interprété)
- ▶ Popularité : 5^{ème} langage le plus utilisé en 2016 (4%)

PHP



- ▶ Première version : 1994
- ▶ Auteur : Rasmus Lerdorf
- ▶ Paradigmes : Impératif, Objet, Fonctionnel, Procédural, Réflexif
- ▶ Utilisation : Web
- ▶ Niveau : Très haut (Interprété)
- ▶ Popularité : 6^{ème} langage le plus utilisé en 2016 (3%)

Visual Basic .NET



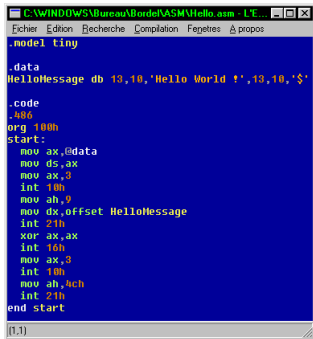
- ▶ Première version : 2001
- ▶ Auteur : Microsoft
- ▶ Paradigmes : Objet, Structuré, Impératif
- ▶ Utilisation : Générique
- ▶ Niveau : Très haut (Pré-Compilé : D'abord compilé en CIL, puis interprété par l'interpréteur .NET)
- ▶ Popularité : 7^{ème} langage le plus utilisé en 2016 (3%)

JavaScript



- ▶ Première version : 1995
- ▶ Auteur : Brendan Eich
- ▶ Paradigmes : Script, Objet, Fonctionnel, Impératif
- ▶ Utilisation : Web
- ▶ Niveau : Très haut (Interprété)
- ▶ Popularité : 8^{ème} langage le plus utilisé en 2016 (3%)

Assembleur



```
.model tiny
.data
HelloMessage db 13,10,'Hello World !',13,10,'$'

.code
.486
org 100h
start:
    mov ax,@data
    mov ds,ax
    mov ax,3
    int 10h
    mov ah,9
    mov dx,offset HelloMessage
    int 21h
    xor ax,ax
    int 16h
    mov ax,3
    int 10h
    mov ah,hch
    int 21h
end start
```

- ▶ Première version : 1954
- ▶ Auteur : Nathaniel Rochester
- ▶ Utilisation : Codage bas niveau/Informatique embarquée
- ▶ Niveau : Bas (Langage Machine)
- ▶ Popularité : 9^{ème} langage le plus utilisé en 2016 (2%)