

# Compilation vs. interprétation

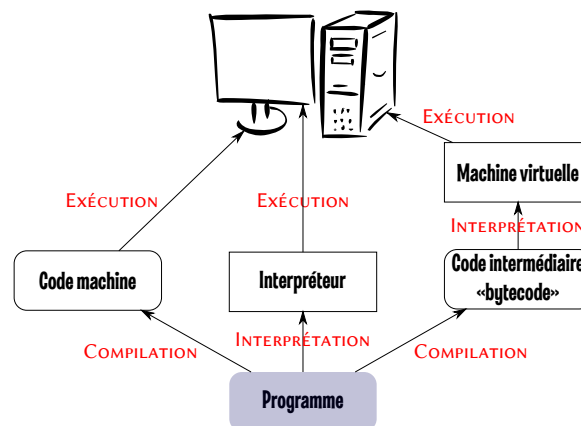
**Pré-requis** : aucun

UN PROGRAMME ÉCRIT DANS UN LANGAGE DE PROGRAMMATION quel qu'il soit (C, Java, BASH, Python, ...) est contenu dans un fichier de texte lisible par un être humain. Pour être exécuté par un ordinateur, le texte doit être traduit — directement ou indirectement — en **code machine**. Cela peut se faire de trois façons :

**Compilation.** Un programme, le *compilateur*, transforme le **code source** en du *code machine* directement exécutable par le processeur ;

**Interprétation.** Un programme, l'*interpréteur*, lit chaque ligne du code source et l'interprète en terme de code à faire exécuter par le processeur ;

**Compilation/interprétation.** Le code source est compilé en **code intermédiaire** pour une **machine virtuelle**, lui-même interprété par le programme simulant cette machine virtuelle sur la machine physique.



Certains langages de programmation sont majoritairement compilés (C, C++, ...), d'autres sont majoritairement interprétés ou compilés sous forme intermédiaire interprétée (Java, Python, ...). Cependant, il existe des interpréteurs pour les premiers (e.g., **CLING** pour C++) et des compilateurs pour les seconds (e.g., Excelsior JET pour Java).

Le passage par un interpréteur ralentit généralement l'exécution d'un programme, même si de nombreuses techniques permettent de réduire de façon importante la différence entre un programme interprété et un programme compilé.

## Glossaire

**code intermédiaire** Suite d'instructions comprises par un processeur sans existence physique dont le comportement est simulé par un programme tournant sur une machine réelle

**code machine** Suite de nombres identifiant les instructions du processeur à exécuter

**code source** Texte d'un programme écrit dans un langage de programmation tel que C, Python, ...

**machine virtuelle** Ordinateur virtuel simulé par un programme et comprenant un code machine particulier, le *code intermédiaire*