



# Les langages compilés vs interprétés

**RÉALISÉ PAR:  
JOUICHAT KHADIJA**





# Plan

- Introduction
- Langages Compilés
- Langages  
Interprétés
- Comparaison des  
Performances
- Cas d'Utilisation et
- Évolution  
Tendances
- Conclusion



# Introduction

Les langages de programmation se divisent en deux catégories : les langages compilés, comme C, qui sont traduits en code machine avant leur exécution, et les langages interprétés, tels que JavaScript, qui sont exécutés ligne par ligne par un interpréteur. Ce choix est crucial dans le développement moderne, car les langages interprétés permettent une rapidité de prototypage et d'itération, tandis que les langages compilés offrent des performances optimisées pour des applications nécessitant une gestion rigoureuse des ressources.

```
margin: 0;
padding: 0;
font-size: 1em;
background: url(Cyber.jpg);
background-size: 100% 100%;
}
.box{
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%,
  width: 400px;
  padding: 40px;
  background: #000000;
  box-sizing: border-box;
  box-shadow: 0 15px 25px #000000;
  border-radius: 10px;
}
.box h2{
  margin: 0 0 30px;
  padding: 0;
  color: #fff;
  text-align: center;
}
.box h3{
  margin: 0 0 10px;
  padding: 0;
  color: #fff;
  text-align: center;
}
.inputBox{
  position: relative;
}
```



# Les Langages Compilés

## Avantages:

Ces langages offrent une performance optimisée grâce à l'exécution rapide du code machine. De plus, la vérification des erreurs à la compilation permet d'identifier et de corriger les problèmes avant l'exécution, réduisant les risques d'erreurs.

## Définitions et exemples:

Les langages compilés sont traduits en code machine par un compilateur avant l'exécution. Des exemples notables incluent C, C++ et Rust, qui sont souvent utilisés dans des domaines nécessitant performance et contrôle des ressources.

## Inconvénients:

Cependant, le processus de compilation peut être long, ralentissant le cycle de développement. De plus, leur nature statique limite la flexibilité, rendant les ajustements plus difficiles, ce qui peut être un inconvénient dans les environnements de développement agiles.

# Les Langages Interprétés

## Avantages:

L'un des principaux avantages des langages interprétés est leur flexibilité, permettant un développement rapide et itératif. Ils sont particulièrement adaptés pour créer des scripts et des prototypes, car les modifications peuvent être testées immédiatement sans étape de compilation.

## Définitions et exemples:

Les langages interprétés sont des langages de programmation exécutés par un interpréteur à la volée, sans nécessiter de compilation préalable. Des exemples courants incluent Python, JavaScript et Ruby, qui sont souvent utilisés pour le développement web et les scripts.

## Inconvénients:

Ces langages présentent généralement des performances inférieures par rapport aux langages compilés, ce qui peut poser des problèmes dans des applications exigeantes. De plus, les erreurs de code peuvent être détectées uniquement lors de l'exécution, ce qui augmente le risque de bugs en production.

## Comparaison des Performances

La comparaison des performances entre langages compilés et interprétés peut être illustrée à l'aide de graphiques ou de tableaux. Ces visuels peuvent montrer les temps d'exécution pour des tâches similaires, mettant en évidence les différences de performance. Par exemple, un algorithme de tri peut être exécuté en quelques millisecondes dans un langage compilé, tandis que le même algorithme en langage interprété peut prendre plusieurs secondes. Cette analyse permet de mieux comprendre les compromis entre vitesse d'exécution et facilité de développement.

## Cas d'Utilisation

Le choix entre un langage compilé et un langage interprété dépend des exigences spécifiques du projet. Un langage compilé est idéal pour des applications nécessitant des performances élevées, comme les systèmes d'exploitation ou les logiciels embarqués. En revanche, un langage interprété est préférable pour des projets qui nécessitent une flexibilité rapide, comme le développement web ou les prototypes. Cette distinction permet aux développeurs de sélectionner le langage le plus approprié en fonction de leurs besoins.







# Évolution et Tendances

Les langages de programmation évoluent constamment, avec une tendance vers la convergence entre les langages compilés et interprétés. Des technologies comme la compilation juste-à-temps (JIT) combinent les avantages des deux paradigmes, permettant une exécution rapide tout en conservant la flexibilité des langages interprétés. Cette évolution a un impact significatif sur le développement moderne, facilitant l'adoption de nouveaux langages et l'intégration de différents paradigmes dans un même projet.

# Conclusion

En résumé, la compréhension des différences entre langages compilés et interprétés est essentielle pour les développeurs. Chaque type de langage présente des avantages et des inconvénients qui influencent le choix en fonction du projet. Il est crucial de sélectionner le bon type de langage pour répondre aux besoins spécifiques, que ce soit pour des performances optimisées ou une flexibilité rapide, afin d'assurer le succès du développement logiciel.





Merci pour votre

attention !