Historique et objectifs du langage

Origines de Go

- Le langage Go, aussi appelé Golang, a été conçu chez Google en 2007 par Robert Griesemer, Rob Pike, et Ken Thompson.
- L'objectif initial était de résoudre des problèmes de productivité logicielle au sein de Google, en particulier pour des systèmes à grande échelle et des applications réseau.
- Go a été annoncé au public en novembre 2009.
- Go a été inspiré par les langages de programmation C, Pascal et Oberon, mais avec un accent particulier sur la simplicité, l'efficacité et la concurrence.
- Les informations détaillées sur l'origine de Go peuvent être trouvées dans l'article "The Go Programming Language" écrit par les créateurs du langage, disponible sur le blog officiel de Go.

Contributeurs clés

- Les principaux contributeurs à la création et au développement de Go sont Robert Griesemer, Rob Pike et Ken Thompson, tous employés de Google à l'époque.
- Rob Pike est l'un des auteurs du système d'exploitation Plan 9, et Ken Thompson est l'un des créateurs du langage C et du système d'exploitation Unix.
- L'équipe de Go a grandi au fil du temps, avec de nombreux contributeurs au sein de Google et de la communauté open source.
- Le code source de Go est hébergé sur GitHub, où la liste des contributeurs peut être consultée.

Objectifs de Go

- Simplicité : Go vise à être simple à lire et à écrire. Il dispose d'une syntaxe minimaliste qui facilite la lecture du code par les humains.
- Efficacité : Go a été conçu pour tirer parti de l'architecture multi-cœur des ordinateurs modernes. Son système de goroutines et de canaux permet de réaliser du code concurrent de manière naturelle.
- Interopérabilité : Go supporte l'interfaçage avec du code C, permettant ainsi d'utiliser des bibliothèques existantes.
- Sécurité : Go a un typage fort et le système de gestion de mémoire automatique aide à éviter certaines classes de bugs et de vulnérabilités.
- La documentation officielle de Go, disponible sur le site golang.org, fournit des informations plus détaillées sur les objectifs et la philosophie du langage.

Comparaison de Go avec d'autres langages populaires

Comparaison syntaxique

C vs Go

- C utilise des en-têtes pour la déclaration et l'importation de packages, tandis que Go utilise le mot clé `import`.
- Les fonctions en Go sont déclarées avec le mot clé `func`, tandis qu'en C, le type de retour est utilisé.
- Go dispose du ramasse-miettes, il n'est donc pas nécessaire de gérer manuellement la mémoire comme en C.
- C utilise des pointeurs pour la manipulation de structures, tandis que Go peut les manipuler directement.
- Exemple :

```c
// C code
#include <stdio.h>
void main() {
```

Comparaison des performances

C vs Go

- C est généralement plus rapide que Go en raison de son contrôle de bas niveau sur le matériel, mais Go est plus facile à optimiser en raison de ses fonctionnalités de haut niveau comme les goroutines.

Python vs Go

- Go est compilé en code machine, ce qui le rend beaucoup plus rapide que Python, qui est un langage interprété.
- Go a également une meilleure performance en matière de concurrence grâce à ses goroutines.

Java vs Go

- Go est généralement plus rapide que Java en termes de temps d'exécution et utilise moins de mémoire, bien que Java ait des performances proches grâce à la JVM et au JIT.