**Golang**

Intro

* Langage développé par Google en interne (plus facile à écrire/entretenir/construire)
* Très populaire pour l’écriture d’APIs et DevOps tools
* Se situe à un niveau intermédiaire entre les langages simples/complexes
* Syntaxe : similaire à d’autres langages notamment **C**
* Rapide et peu gourmand en mémoire
* Syntaxe simple pour les programmes multi-thread (à instructions multiples)
* Fournit des fonctionnalités orientées objet
* Dispose d’une poubelle

I Compilation de fichiers

* *go build file-name.go* : créer le fichier file-name.go exécutable de notre programme
* *ls* : voir le programme d’origine et son fichier exécutable
* *. /file-name* : exécuter le fichier
* *go run file*-name.go *:* ne crée pas de fichier mais en exécute un déjà existant

1. Go packages

* Les projets peuvent contenir beaucoup de fichiers .go organisés en packages
* Tous les fichiers .go qui ont rapport avec une partie de notre programme vont aller dans un package
* D’autres fichiers .go ayant rapport avec une autre partie dans un autre package
* Déclaration packages

*package file-name*

*import ‘’fmt’’*

*func file-name(){*

*fmt.Println(‘’Hello World’’)*

*}*

* ***Package file-name*** : déclare au compilateur à quel package appartient file-name.go
* ***import ‘’fmt’’***: permet l’utilisation de codes d’autres packages, ici la fonction ***Println*** du pakage ***fmt***
* ***func***: déclaration de fonction, suivi du nom de la fonction ***file-name()***, suivi de ***{code de la fonction}***

1. Importer packages multiples

* Déclaration multiples d’***import*** ou unique avec des ***()***

*import ‘’package 1’’*

*import ‘’package 2’’*

*import (*

*‘’package 3’’*

*‘’package 4’’*

*)*

* Utiliser un alias

*import p1 ‘’package 1’’*

*import p2 ‘’package 2’’*

*import (*

*p1 ‘’package 1’’*

*p2 ‘’package 2’’*

*)*

II Go Ressources

1. Go Doc

* Programme ***go doc*** : extraire/consulter la documentation des fichiers ***.go***
* Information sur un package : ***go doc package-name*** (par ex : ***go doc fmt***)
* Info sur la fonction d’un paquet : ***go doc pakage-name.function-name*** (ex : ***go doc fmt.Println***)