« Hello world ! »

But du TP : Avoir un environnement de développement go opérationnel

1. Remarques

Pour ce cours, j'utiliserai le système d'exploitation Linux. À vous de l'adapter à l'OS que vous utilisez.

2. Installation

Pour écrire du code go, il est nécessaire de télécharger et d'installer l'environnement de développement go. Vous trouverez la dernière version sur le site :

- https://go.dev/dl/

Dans mon cas, j'utilise la dernière version Linux

- https://go.dev/dl/go1.21.3.linux-amd64.tar.gz

Décompresser le fichier go1.21.3.linux-amd64.tar.gz dans le répertoire tools. Puis ajoutons le chemin du compilateur au chemin par défaut.

- mkdir tools
- tar -C ~/tools/ -xzf ~/Téléchargements/go1.21.3.linux-amd64.tar.gz
- echo 'export PATH=$PATH:"$HOME/tools/go/bin"' >> .bashrc
- source .bashrc

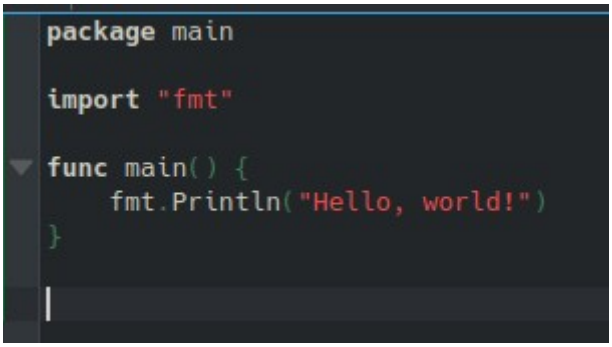
Vérifier que le compilateur fonction bien :

- go --version

Nous pouvons passer à l'étape suivante.

3. Compilation

Dans le cas de programme simple, un éditeur de texte suffit à coder votre programme. Dans mon cas, avec « kate », je crée le fichier « main.go » dans un répertoire « tp1 »



Puis, comme en C, je compile mon code pour obtenir un exécutable:

- `go build hello.go`

et de le lancer :

- `./hello`

Il est aussi possible d'avoir un exécutable 'temporaire'

- `go run hello.go`

4. Compilation

Quand de nombreux fichier compose notre programme, il est plus simple d'utiliser un framework Dans le cadre de ce cours, nous allons utiliser Visual Studio Code, avec le plugin Goland.
Vous pouvez le télécharger à l'adresse:
https://code.visualstudio.com/download
et l'installer dans le répertoire ~/tools
Lancer la console de VSC, et créer un nouveau répertoire «tp1»
file→open folder, puis créer le répertoire

Maintenant, créer le fichier main.go, qui affiche la célèbre phrase « Hello world »
VSC vous proposera d'installer le plugin go : répondre dans l'affirmative.

Vérifier que le programme se lance

- F5 ou run→ start debugging

Pour la suite, nous allons initialiser ce programme comme un nouveau package. Nous verrons plus tard dans le cours ce que cela signifie.

- go mod init tp/one

5. Directive d'exécution

Pour ne pas avoir de problème lorsque nous interagirons avec le programme, rajouter systématiquement le fichier launch.json dans le répertoire .vscode

```
{
	"version": "0.2.0",
	"configurations": [
		{
			"name": "Launch Package",
			"type": "go",
			"request": "launch",
			"mode": "auto",
			"program": "${workspaceFolder}",
			"console": "integratedTerminal"
		}
	]
}
```

Pour l'instant, ces connaissances nous suffisent pour aborder le cours.