

Os.Args

Commande CLI

Nous allons aujourd'hui voir, comment récupérer des arguments depuis votre CLI lorsque vous exécutez votre commande préféré:

```
go run main.go
```

Plusieurs points à noter qui sont importants:

- On peut mettre autant d'arguments que l'on souhaite
- Les arguments sont pris en compte après le mot-clé `run`
- Le premier argument est donc la direction vers votre fichier `.go`
- Lorsque vous exécutez la commande mettez vous à la racine du fichier go

Voici un exemple d'utilisation lors de l'exécution de la commande:

```
go run main.go "coucou" "test"
```

Dans notre exemple on peut donc voir que nous avons 3 arguments:

- Le premier: notre direction vers notre fichier `.go`
- Le second: notre string `coucou`
- Le troisième: notre string `test`

Code

Maintenant que nous avons vu comment cela se déroule dans le CLI, il faut maintenant regarder au niveau du code comment le récupérer.

Exemple d'utilisation:

```
package main

import "os"

func main() {
    arguments := os.Args
}
```

Avant d'aller plus loin dans les explications plusieurs points à savoir:

- Il faut importer le package `os`
- `os.Args` retourne un tableau qui contient chaque argument
- Chaque argument du tableau est une string

Dans l'exemple d'utilisation au-dessus, on voit donc l'import du package `os` ainsi que la déclaration et attribution de la variable `arguments` égal à `os.Args`.

Actuellement dans notre exemple `arguments` est donc une variable qui est un tableau contenant un/ou plusieurs argument(s) de type `string`.

La taille de la variable `arguments` va donc dépendre du nombre d'arguments que vous allez envoyer dans la commande en CLI.

Dernier exemple d'utilisation:

Fichier `go`

```
package main

import "os"

func main() {
    arguments := os.Args
}
```

CLI

```
go run main.go "test" "exemple" "1"
```

Là dans notre exemple la variable `arguments` est donc un tableau contenant 4 `string`:

- La direction de votre fichier
- `"test"`
- `"exemple"`
- `"1"`

Vous avez maintenant toutes les clés en main pour pouvoir réussir cette quête !

Bon courage à tous !

N'oubliez pas de toujours tester votre code en locale !