

WORKSHOP GOLANG

La Concurrency en GO

GOLANG



```
package main

import "fmt"

func main() {
    fmt.Println("Hello World !")
}
```

INSTALLATION



```
sudo pacman -S go
```



```
sudo apt-get install golang
```



```
sudo dnf install go
```



```
brew install go
```

INSTALLATION

Lancez la commande

```
mkdir ~/.go
```

Et ajoutez

```
export GOPATH=~/.go
```

Dans votre .bashrc

INSTALLATION

Lancez la commande

`go version`

Vous devriez avoir un résultat qui ressemble à:

`go version go1.13.1 linux/amd64`

Partie 1: PARALLELISM

-> Utilisez les goroutines pour faire des appels à l'API Chuck Norris Facts

-> Utilisez une `chan` pour print les facts et le numéro de la goroutine dans le `main`

```
± % ./Part1
0 `Chuck Norris hates it when his women have multiple orgasms and pass out before he's even gotten warmed up.`
4 `Women that work in an office with Chuck Norris, soon learn not ask him for his 3 hole punch.`
3 `The smell of Chuck Norris feet can scare a skunk.`
1 `Chuck norris eats "man bear pig" bacon for breakfast with al gore blood v8 juice`
6 `It is better to give than to receive. This is especially true of a Chuck Norris roundhouse kick.`
5 `Chuck Norris can sit in a chair, pull up on the sides, and lift himself off the ground.`
7 `Chuck Norris can pull your entire skeletal system through any orifice on your body.`
8 `Chuck Norris will never have a heart attack. His heart isn't nearly foolish enough to attack him.`
2 `Chuck Norris's blood smells like cologne.`
9 `When Chuck Norris was 3 years old, he went Back to the Future in his Little Tikes car.`
```


TIPS: Partie 1

Le repo de ce workshop -> `git clone https://github.com/ctinarelli/WorkshopGO.git`

`go run exemples/hello.go, go run exemples/http.go, go run exemples/goroutine.go`

Des exemples des features de GO <https://gobyexample.com/>

Documentation sync.WaitGroup <https://golang.org/pkg/sync/#WaitGroup>

Partie 2: PIPELINING

- > Séparez chaque opérations, du parsing de la requête au print, en une fonction
- > Lancez chaque fonctions dans une goroutine
- > Depuis le `main`, envoyez dans cette `pipeline` un nombre de requête que vous récupèrerez sur `os.Stdin`

```
± % ./Part2
> 3
0 Pound for pound, Chuck Norris will pound the shit out of you.
1 A long time ago, in a galaxy far,far away.... Chuck Norris roundhouse kicked the shit out of somebody.
> 2 Chuck Norris is your father!
coucou
Can't parse `coucou` as an integer
>
Goodbye :)
```


TIPS: Partie 2

```
func ...(in <-chan ..., out chan<- ...)
```

```
go run exemples/hexshell.go
```

La première fonction ne prend qu'une channel `in`

La dernière ne prend qu'une channel `out`

Pas besoin de `sync.WaitGroup`