

Le Langage Go

**Struct**

**Receiver**

# 2 types de code

# **Les structs**

# **Les fonctions**

# Les structs

**Regrouper des données**

# Les fonctions

**Effectuer des traitements**

# Les méthodes

# **Les méthodes**

**Effectuer des traitements  
sur des données**

# Syntaxe



# Syntaxe

```
type User struct {  
    Name string  
}
```

# Syntaxe

```
type User struct {  
    Name string  
}
```

```
func (u User) SayHello() {  
    fmt.Printf("Hello %v!\n", u.Name)  
}
```

# Syntaxe

```
type User struct {  
    Name string  
}
```

```
func (u User) SayHello() {  
    fmt.Printf("Hello %v!\n", u.Name)  
}
```

- `u` est un **value receiver** pour la méthode `SayHello()`

# Syntaxe

```
type User struct {  
    Name string  
}
```

```
func (u User) SayHello() {  
    fmt.Printf("Hello %v!\n", u.Name)  
}
```

- `u` est un **value receiver** pour la méthode `SayHello()`
- Les champs de `u` sont accessibles pour la fonction

# Utilisation de la méthode

# Utilisation de la méthode

```
u := User{"Bob"}
```

# Utilisation de la méthode

```
u := User{"Bob"}  
u.SayHello() // output: Hello Bob!
```

# Pourquoi **value** receiver ?



**Parce-que pour la fonction, le struct  
est copié en tant que valeur**

# Conséquence...

**Un méthode avec un `value receiver`  
ne peut jamais modifier  
le struct original**