

Le Langage Go

# Interface

## Interêt d'une Interface

# Définition

# Définition

**Regroupement nommé de  
signatures de fonctions**



**Une fonction est un contrat**

**“ Pour ces inputs, je m'engage à fournir cet output**

*-- Une fonction honnête, 2018*

**”**

**On sait que l'on veut un comportement**

**On sait que l'on veut un comportement**

**On ne veut pas connaître l'implémentation**



# Exemple

# Exemple

**Un animal peut avoir plusieurs comportements**



“ Roarrrrr !!!

-- *Un lion utilisant une fonction mal codée, 2017*

”



“ BzzzzzzZzzzzzzZ !

-- *Une abeille tombant sur un bout de code super classe, 2014*

”

# Les 2 "parlent"

**Les 2 "parlent"**

**Mais d'une manière complètement différente**

# Pseudo-code interface

# Pseudo-code interface

```
interface AnimalTalker {  
    func Talk()  
}
```



# Pseudo-code interface

```
interface AnimalTalker {  
    func Talk()  
}
```

```
var animal AnimalTalker = Lion{...}
```

# Pseudo-code interface

```
interface AnimalTalker {  
    func Talk()  
}
```

```
var animal AnimalTalker = Lion{...}  
animal.Talk() // Rooaaarr!!
```

# Pseudo-code interface

```
interface AnimalTalker {  
    func Talk()  
}
```

```
var animal AnimalTalker = Lion{...}  
animal.Talk() // Rooaaarr!!
```

```
animal = Bee{...}  
animal.Talk() // BzzzzzzZz!
```

# Polymorphisme

# Polymorphisme

**Une fonction qui a plusieurs formes**

# **Etape ultime de la boite noire**