

Le Langage Go

# Les bases du langage

## Erreurs

# Plusieurs stratégies possibles

# Plusieurs stratégies possibles

Code d'Erreurs, Exceptions, Pattern Matching, ...

**En Go, tout se base sur le  
retour de fonction multiple**

# Exemple classique

# Exemple classique

```
func MyFunc() (int, error) { ... }
```

# Exemple classique

```
func MyFunc() (int, error) { ... }
```

```
v, err := MyFunc()
```

# Exemple classique

```
func MyFunc() (int, error) { ... }
```

```
v, err := MyFunc()  
if err != nil {  
    fmt.Printf("Error in MyFunc: %v", err)  
}
```



# Forme d'un code Go standard



```
v, err := MyFunc1()  
if err != nil {  
    return err  
}
```

```
v, err := MyFunc2()  
if err != nil {  
    return err  
}
```

```
v, err := MyFunc3()  
if err != nil {  
    return err  
}
```

```
v, err := MyFunc4()  
if err != nil {  
    return err  
}
```

# Répétitif

**Répétitif**  
**Mais efficace !**

**On lit toujours le code  
progressivement**

# Favoriser le **early return**

# Code non-early return

```
func MyFunc(condition bool) (int, err) {  
    if (condition) {  
        // code  
        return 42, nil  
    }  
    return 0, errors.New("Error!")  
}
```

# Code non-early return

```
func MyFunc(condition bool) (int, err) {  
    if (condition) {  
        if (!condition2) {  
            return 0, errors.New("Error 2!")  
        }  
        // code  
        return 42, nil  
    }  
    return 0, errors.New("Error!")  
}
```



# Code non-early return

```
func MyFunc(condition bool) (int, err) {  
    if (condition) {  
        if (!condition2) {  
            return 0, errors.New("Error 2!")  
        }  
        // code  
        return 42, nil  
    }  
    return 0, errors.New("Error!")  
}
```

- Valide, mais grandit difficilement

# Code early return

```
func MyFunc(condition bool) (int, err) {  
    if (!condition) {  
        return 0, errors.New("Error!")  
    }  
    // code  
    return 42, nil  
}
```

# Code early return

```
func MyFunc(condition bool) (int, err) {  
    if (!condition) {  
        return 0, errors.New("Error!")  
    }  
    if (!condition2) {  
        return 0, errors.New("Error 2!")  
    }  
    // code  
    return 42, nil  
}
```