

**Le langage Go**

**Approfondir Go**

**Erreur custom**

# En Go le type `error` n'est qu'une interface renvoyant un message d'erreur

```
type error interface {  
    Error() string  
}
```

**Comment faire pour ajouter des informations à une erreur ?**

**Un code d'erreur, un module, etc**

**Il suffit de faire un struct et  
d'implémenter l'interface !**

# Exemple d'erreur custom

```
type MyError struct {  
    Module string  
    Err error  
}  
  
func (e *MyError) Error () string {  
    return fmt.Sprintf("Module=%v, error=%v", e.Module, e.Err)  
}
```

# Utilisation de l'erreur custom

```
func process() error {  
    return &MyError {  
        Module: "main",  
        Err: fmt.Errorf("Could not execute process()"),  
    }  
}  
  
func main() {  
    err := process()  
    if err != nil {  
        // Affiche Module=main, error=Cound not execute process()  
        fmt.Println(err)  
    }  
}
```

**Allons faire un exemple en code**