

Programmation Orientée Objet

Generics 1/2



Pourquoi faire ?

- Appliquer un même comportement sur des classes différentes
 - Stocker des éléments dans des listes (`Int`, `String`, `MyClass`, ...)
 - Trier des éléments (`Int`, `String`, `MyClass`, ...)

Avantages

- Gérer un nouveau type sans tout casser !
- Plus de vérifications à la compilation

En Kotlin

- Marquer une classe / fonction pour qu'elle accepte un type générique
- Dans le code, manipuler un type générique plutôt qu'un type concret
- Type générique souvent appelé **T** pour Type

Syntaxe d'une fonction générique

```
fun <T> myFunction(var1: T) {  
    // Explosive stuff  
}
```