

# JAVA FONDAMENTAUX

---

Présenté par :  
**Xavier TABUTEAU**

## Les records

# Les records

Depuis la version 14 du JDK, il est possible d'utiliser une classe de type « record ». Ce type permet de créer une classe immuable. On ne peut pas hériter d'un record car c'est implicitement une classe de type « final ».

Une classe immuable est une classe n'ayant que des attributs « private final » accessibles par des getteurs, un constructeur permettant d'initialiser ces attributs, une redéfinition des méthodes hashCode(), toString() et equals(). Il n'y a pas de setteur du fait que les attributs ont « final ». La classe la plus connue correspondant à cette description est la classe String.

L'intérêt des records c'est que tout le code nécessaire pour faire cela, peut-être faire en une seule ligne.

Syntaxe :

```
« Visibilité » record nomRecord(type Param1, ...)
{
    ...
}
```

## Les records

# Les records

Attention, l'écriture d'un « record » simple ne génère pas automatiquement de constructeur sans paramètre, les getters générés n'ont pas le préfixe get et les records ne sont pas héritables du fait d'être « final ».

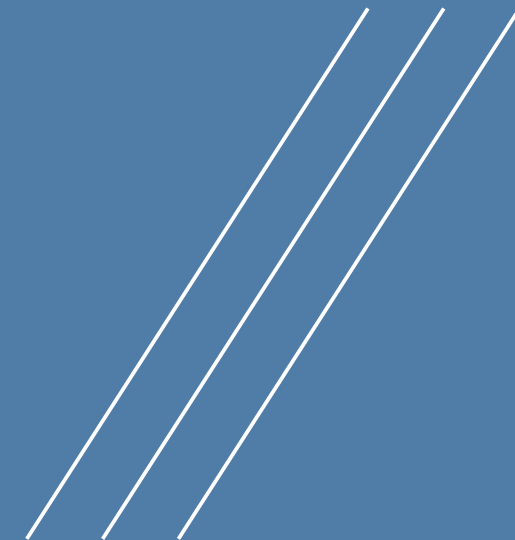
A l'utilisation les « records » s'utilisent comme s'il s'agissait d'une classe traditionnelle en tenant compte des indications mentionnées précédemment.

Il est possible de compléter un record avec nos propres méthodes, constructeurs etc.  
Si l'on veut ajouter un constructeur sans paramètre, il faudra aussi redéfinir le constructeur à n paramètres car le « records » ne générera pas le constructeur initial.

L'ajout de méthode se fait comme pour une classe normale mais la méthode n'aura pas accès en écriture sur les attributs du fait qu'ils ont « final ».

Records.java

# JAVA FONDAMENTAUX



Présenté par  
**Xavier TABUTEAU**