

JAVA 13

JÉRÉMY PERROUULT

A decorative wavy line in light blue and white, flowing vertically along the left edge of the slide.

INTRODUCTION

INTRODUCTION

INTRODUCTION

- Java 13 a apporté les changements suivants
 - Introduction des TextBlocks
 - Amélioration de la classes **String**
 - Nouvelle classe **CompactNumberFormat**
 - Nouvelle expression du bloc switch (introduction `yield`)



TEXTBLOCK

NOUVEAUTÉS JAVA 12

TEXTBLOCK

- Nouvelle syntaxe pour les textes sur plusieurs lignes

```
String json = "{\r\n" +  
    "\"name\" : \"Jeremy\", \r\n" +  
    "\"email\" : \"jeremy@perrouault.fr\" \r\n" +  
    "}";  
  
System.out.println(json);
```

TEXTBLOCK

- En utilisant les triples guillemets (""")

```
String jsonTextBlock = """
{
    "name": "Jeremy",
    "email": "jeremy@perrouault.fr"
}
""";

System.out.println(jsonTextBlock);
```

- Plus besoin non plus d'échapper les guillemets dans la chaîne
- Attention cependant, les tabulations et espaces en début de ligne feront parti du texte ...
- C'est un objet **String** donc toutes les fonctionnalités sont disponibles

A decorative wavy line in light blue and white, running vertically along the left side of the slide.

STRING

NOUVEAUTÉS JAVA 13

STRING

- De nouvelles fonctionnalités sont ajoutées
 - `stripIndent` Permet de supprimer les indentations
 - `translateEscapes` Transforme les caractères d'échappement « `\\` » en « `\` »
 - `\\t` devient `\t` par exemple
 - `formatted` Permet de formater la chaîne directement, comme **`String.format`**

```
String prenom = "Jeremy";
String email = "jeremy@perrouault.fr";
String jsonTextBlock = """
{
    "name": "%s",
    "email": "%s"
}
""".formatted(prenom, email);

System.out.println(jsonTextBlock);
```




SWITCH

NOUVEAUTÉS JAVA 12

SWITCH

- Ajout du mot-clé `yield` pour le bloc Switch
 - Qui permet de retourner une valeur alors qu'on se trouve dans un bloc d'instructions

SWITCH

```
DayOfWeek jourSemaine = LocalDate.now().getDayOfWeek();

String typeJour = switch (jourSemaine) {
    case MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY -> {
        "Semaine"; // KO
    }
    default -> "Week-end";
};

System.out.println(typeJour);
```

```
DayOfWeek jourSemaine = LocalDate.now().getDayOfWeek();

String typeJour = switch (jourSemaine) {
    case MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY -> {
        return "Semaine"; // KO
    }
    default -> "Week-end";
};

System.out.println(typeJour);
```

SWITCH

```
DayOfWeek jourSemaine = LocalDate.now().getDayOfWeek();

String typeJour = switch (jourSemaine) {
    case MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY -> {
        yield "Semaine"; // OK
    }
    default -> "Week-end";
};

System.out.println(typeJour);
```