Atitit 字符串模板与 字符串插值

目录

[1.1. 字符串插值是将字符串中的占位符替换为局域变量的过程 1](#_Toc12565)

[1.2. 模板字面量 是允许嵌入表达式的字符串字面量 1](#_Toc6777)

[1.3. 格式化字符串[编辑] 1](#_Toc31826)

[2. 单个的变量插值与map对象插值 2](#_Toc3945)

[2.1. 实现方式 变量插值 2](#_Toc13361)

[2.2. 对象属性插值 gson objectJson对象模式 2](#_Toc21475)

[3. 常见场景 2](#_Toc4636)

[3.1. Json obj 插值字符串 2](#_Toc26932)

[3.2. Map插值字符串 3](#_Toc12688)

[4. 常见的类库velocity等 3](#_Toc9921)

## 字符串插值是将字符串中的占位符替换为局域变量的过程

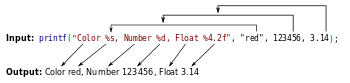
。许多编程语言都可以做到

## 模板字面量 是允许嵌入表达式的字符串字面量

## 格式化字符串[[编辑](https://zh.wikipedia.org/w/index.php?title=%E6%A0%BC%E5%BC%8F%E5%8C%96%E5%AD%97%E7%AC%A6%E4%B8%B2&action=edit&section=0&summary=/* top */ " \o "编辑首段)]

维基百科，自由的百科全书

[跳到导航](https://zh.wikipedia.org/wiki/%E6%A0%BC%E5%BC%8F%E5%8C%96%E5%AD%97%E7%AC%A6%E4%B8%B2" \l "mw-head)[跳到搜索](https://zh.wikipedia.org/wiki/%E6%A0%BC%E5%BC%8F%E5%8C%96%E5%AD%97%E7%AC%A6%E4%B8%B2" \l "p-search)

[](https://zh.wikipedia.org/wiki/File:Printf.svg)

一个printf函数的例子

**格式化字符串**（英语：**format string**），是一些程序设计语言在格式化输出API函数中用于指定输出参数的格式与相对位置的字符串参数，例如C、C++等程序设计语言的printf类函数，其中的转换说明（conversion specification）用于把随后对应的0个或多个函数参数转换为相应的格式输出；格式化字符串中转换说明以外的其它字符原样输出

。[[1]](https://zh.wikipedia.org/wiki/%E6%A0%BC%E5%BC%8F%E5%8C%96%E5%AD%97%E7%AC%A6%E4%B8%B2" \l "cite_note-1)

# 单个的变量插值与map对象插值

## 实现方式 变量插值

## 对象属性插值 gson objectJson对象模式

**private** **static** String processVars(String sql, JsonObject asJsonObject) {

JsonObject JsonObject1 = asJsonObject;

// new JsonParser().parse(t).getAsJsonObject();

Set<Entry<String, JsonElement>> setE = JsonObject1.entrySet();

**for** (Entry<String, JsonElement> entry : setE) {

// System.out.println(entry.getKey());

// System.out.println(entry.getValue());

JsonElement value = entry.getValue();

sql = sql.replace("@" + entry.getKey() + "@", "'" + value.getAsString() + "'");

}

**return** sql;

# 常见场景

## Json obj 插值字符串

Only diy ...bcz volocity is self obj root ..not supt map or jsonobj

**private** **static** String processVars(String sql, JsonObject asJsonObject) {

JsonObject JsonObject1 = asJsonObject;

// new JsonParser().parse(t).getAsJsonObject();

Set<Entry<String, JsonElement>> setE = JsonObject1.entrySet();

**for** (Entry<String, JsonElement> entry : setE) {

// System.out.println(entry.getKey());

// System.out.println(entry.getValue());

JsonElement value = entry.getValue();

sql = sql.replace("@" + entry.getKey() + "@", "'" + value.getAsString() + "'");

}

**return** sql;

## Map插值字符串

# 常见的类库velocity等