Atitit 字符串模板技术 python

# **2.15 字符串中插入变量**

       match\_type=str(tds[1].string)

        #print(tds[1].string)   #saishimchen

        match\_time=str(tds[2].string)

        home\_team=str(tds[3].string)

        away\_team=str(tds[5].string)

       # print(str (saisiMeinchen) +str(time) +str( judui) +str( kadwi))

        # 执行sql语句

        md5= sdk.encry.get\_md5(str (match\_type) +str(match\_time) +str( home\_team) +str( away\_team)   )

        #sdk.mysql. uniqueIdx("kok\_match\_t", "md5key", md5,conn)

        rel\_sql = string.Template("INSERT INTO kok\_match\_t(id, match\_type, match\_time, home\_team,away\_team,md5key)values" \

                                "('$id','$match\_type','$match\_time','$home\_team','$away\_team','$md5')  " )

        ##replace  the def vals..beirs if use json mode,,must manual defi  cant auto by var

        rel\_sql= string.Template(rel\_sql.safe\_substitute(  vars()))

        rel\_sql=(  rel\_sql.safe\_substitute( { "id":sdk. tools.generate\_id()} )   )

## **问题**

你想创建一个内嵌变量的字符串，变量被它的值所表示的字符串替换掉。

## **解决方案**

Python 没有对在字符串中简单替换变量值提供直接的支持。

如果没有直接的支持，可以使用但是通过使用字符串的 format() 方法来解决这个问题。比如：

**>>>** s = '{name} has {n} messages.'**>>>** s.format(name='Guido', n=37)'Guido has 37 messages.'

## **讨论**

多年以来由于 Python 缺乏对变量替换的内置支持而导致了各种不同的解决方案。 作为本节中展示的一个可能的解决方案，你可以有时候会看到像下面这样的字符串格式化代码：

**>>>** name = 'Guido'**>>>** n = 37**>>>** '%(name) has %(n) messages.' % vars()'Guido has 37 messages.'>>>

你可能还会看到字符串模板的使用：

**>>> import** **string>>>** s = string.Template('$name has $n messages.')**>>>** s.substitute(vars())'Guido has 37 messages.'>>>

然而， format() 和 format\_map() 相比较上面这些方案而已更加先进，因此应该被优先选择。 使用 format() 方法还有一个好处就是你可以获得对字符串格式化的所有支持 (对齐，填充，数字格式化等待)， 而这些特性是使用像模板字符串之类的方案不可能获得的。

# Replace替换法

  rel\_sql = f"INSERT INTO kok\_match\_t(id, match\_type, match\_time, home\_team,away\_team,md5key)values" \

                                "('{id}','@match\_type@','@match\_time@','@home\_team@','@away\_team@','@md5key@')  "

       # rel\_sql=rel\_sql.replace('@id@', str(movie\_id));

        rel\_sql=rel\_sql.replace('@match\_type@', str (saisiMeinchen));

        rel\_sql=rel\_sql.replace('@match\_time@',str(time));

        rel\_sql=rel\_sql.replace('@home\_team@', str( judui));

        rel\_sql=rel\_sql.replace('@away\_team@', str( kadwi));

        rel\_sql=rel\_sql.replace('@md5key@', md5);

# 字符串解释器

## **Formatted String Literals(Python3.6)**

这种格式化字符串的方法是由 Python3.6 新加入的。它允许你在字符串内内嵌 Python 运算式:

a = 11b = 2**return** f'{a} + {b} = {a + b};{a} - {b} = {a - b}'

11 + 2 = 13;11 - 2 = 9

它也同时支持 str.format () 方法中的那些格式化符，且使用方法也一样:

num = 222**return** f'{num} 的16进制表示是 {num:#x}'

222 的16进制表示是 0xde

## **如何抉择？**

有这么多种方法格式化字符串，该如何抉择呢？

[Python-String-Formatting](https://dbader.org/blog/python-string-formatting) 中有一个很好的建议:

