**钩子函数context\_processor详解：**

**1.context\_processor：**上下文处理器。使用这个钩子函数，必须返回一个字典。这个字典中的值在所有模版中都可以使用。这个钩子函数的函数是，如果一些在很多模版中都要用到的变量，那么就可以使用这个钩子函数来返回，而不用在每个视图函数中的`render\_template`中去写，这样可以让代码更加简洁和好维护。

**2.代码演示：**

|  |
| --- |
| **from** flask **import** Flask,request,session,current\_app,url\_for,g,render\_template  **import** os  app = Flask(\_\_name\_\_)  app.config[**'SECRET\_KEY'**]=os.urandom(24) *#加盐 混淆原数据的作用*  @app.route(**'/'**)  **def** hello\_world():  print(**"hi"**)  session[**'uname'**]=**"momo"**  *# return "hello world "*  **return** render\_template(**"index.html"**)  @app.route(**'/li'**)  **def** mylist():  print(**"mylist"**)  *# print("直接取出",g.user)*  **if** hasattr(g,**"user"**):  print(**"条件取出"**, g.user)  *# return "hello world "*  **return** render\_template(**'list.html'**)  @app.before\_request  **def** before\_request():  *# print('在视图函数执行之前执行的钩子函数')*  *# 场景：若用户已经登录了，验证时把用户名放入session中,之后取出来，放入钩子函数，以后访问的视图函数中可直接取出来使用*  uname = session.get(**'uname'**)  print(uname)  **if** uname:  g.user = uname  @app.context\_processor  **def** context\_processor():  **if** hasattr(g,**'user'**):  **return** {**"current\_user"**:g.user}  **else**:  **return** {}  **if** \_\_name\_\_ == **'\_\_main\_\_'**:  app.run(debug=**True**) |