**Flask\_app上下文详解:**

app上下文，也叫应用上下文。

**应用上下文：**

应用上下文是存放到一个`LocalStack`的栈中。和应用app相关的操作就必须要用到应用上下文，

比如通过`current\_app`获取当前的这个`app`名字。

**注意1：**

在视图函数中，不用担心应用上下文的问题。因为视图函数要执行，那么肯定是通过访问url的方式执行的，

那么这种情况下，Flask底层就已经自动的帮我们把应用上下文都推入到了相应的栈中。

**注意2：**

如果想要在视图函数外面执行相关的操作，

比如获取当前的app名称，那么就必须要手动推入应用上下文：

第一种方式：便于理解的写法

|  |
| --- |
| **from** flask **import** Flask,current\_app  app = Flask(\_\_name\_\_)  *#app上下文*  app\_context = app.app\_context()  app\_context.push()  print(current\_app.name)  @app.route(**'/'**)  **def** hello\_world():  print(current\_app.name) *#获取应用的名称*  **return 'Hello World!'**  **if** \_\_name\_\_ == **'\_\_main\_\_'**:  app.run(debug=**True**) |

第二种方式：用with语句

|  |
| --- |
| **from** flask **import** Flask,current\_app  app = Flask(\_\_name\_\_)  *#app上下文*  *#换一种写法*  **with** app.app\_context():  print(current\_app.name)  @app.route(**'/'**)  **def** hello\_world():  print(current\_app.name) *#获取应用的名称*  **return 'Hello World!'**  **if** \_\_name\_\_ == **'\_\_main\_\_'**:  app.run(debug=**True**) |

**图解：**

