**Pycharm开启debug模式和修改host：**

在代码中制作一个错误:

@app.route(**'/'**)

**def** hello\_world():

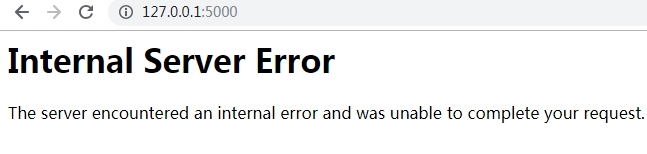
a = 1

b = 0

c = a/b

**return 'Hello World!'**

启动并访问：



控制台倒是给出了错误提示信息，但是 我们希望在浏览器也能有相应的提示信息



对于 pycharm2017，可这样处理

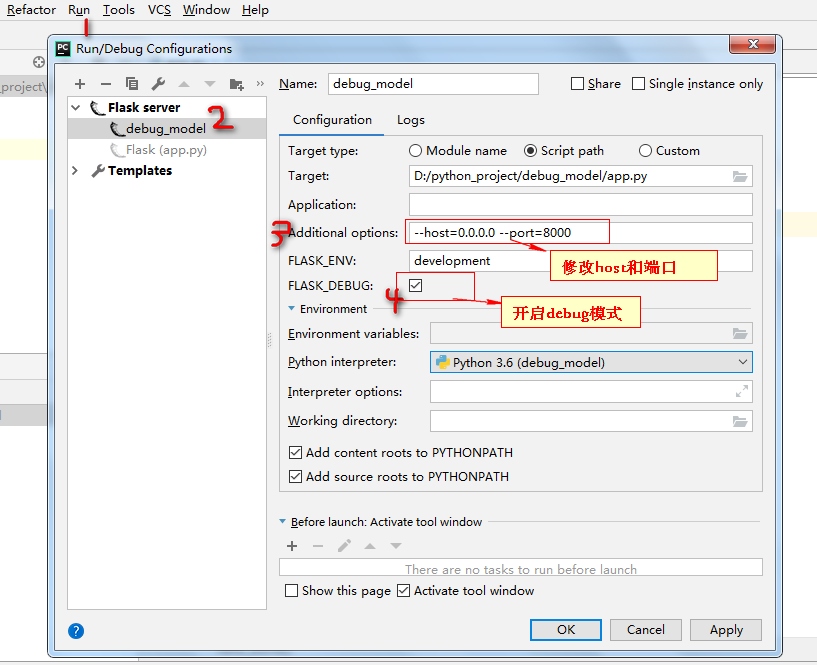
**if** \_\_name\_\_ == **'\_\_main\_\_'**:

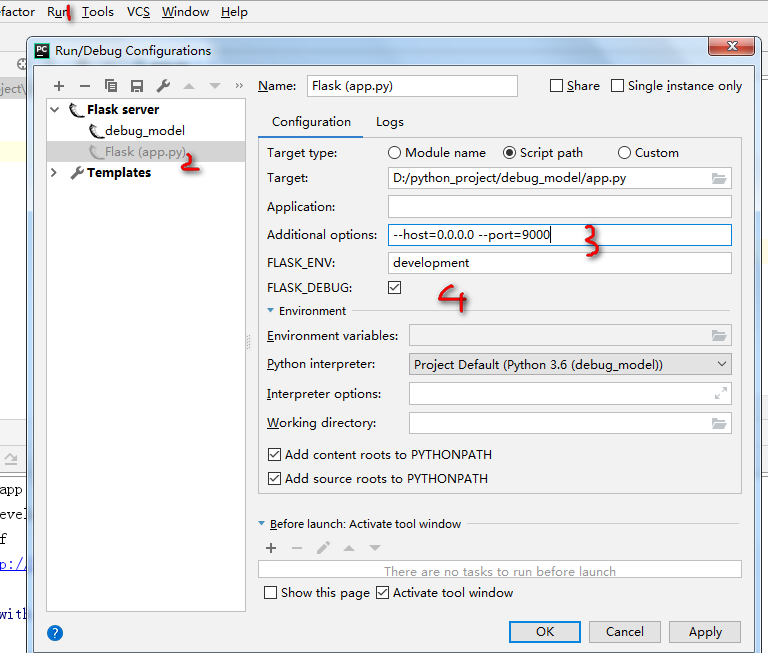
#app.run()

#PyCharm2017.xx 开启debug调式模式

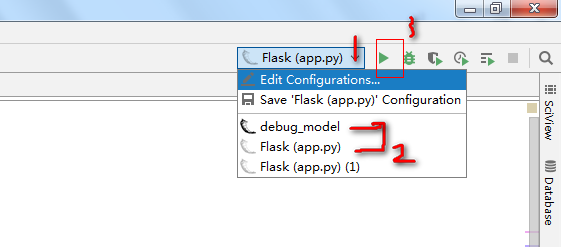
app.run(debug=**True**)

在 Pycharm 2018 中，如果想要开启 debug 模式和更改端口号，则需要编辑项目配置。直接在 app.run 中更改是无效的。示例图如下：

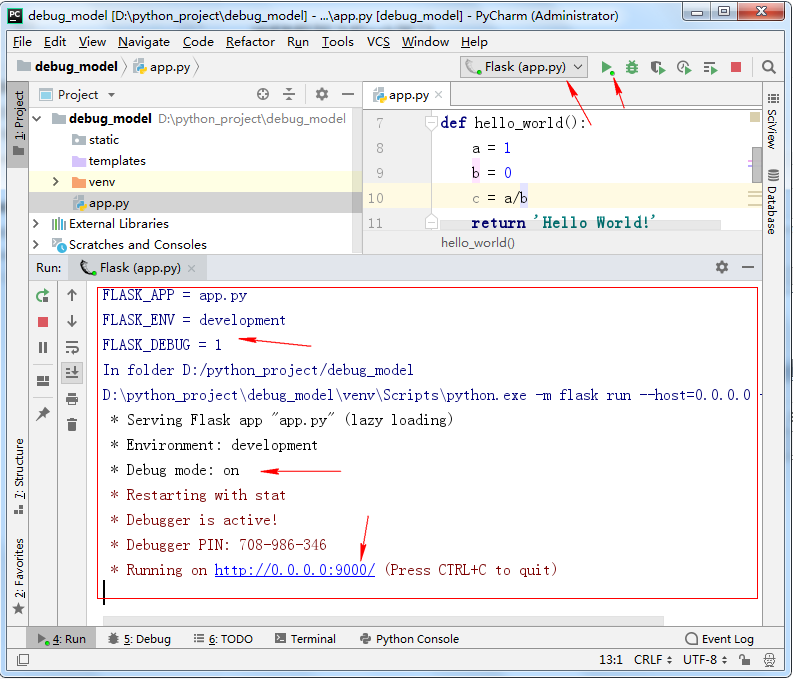




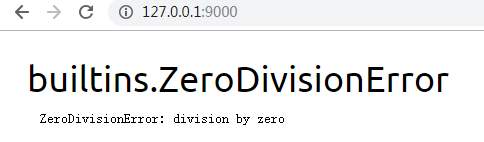
此时，注意 运行技巧



看结果



看浏览器访问结果



**总结1：为什么需要开启DEBUG模式**

1. 如果开启了DEBUG模式，那么在代码中如果抛出了异常，在浏览器的页面中可以看到具体的错误信息，以及具体的错误代码位置。方便开发者调试。

2. 如果开启了DEBUG模式，那么以后在`Python`代码中修改了任何代码，只要按`ctrl+s`，`flask`就会自动的重新加载最新代码。不需要手动点击重新运行。

**总结2：配置DEBUG模式的四种方式**

1. 在`app.run()`中传递一个参数`debug=True`就可以开启`DEBUG`模式。

pycharm2017适用 但pycharm2018需要配置项目信息才行

2. 给`app.deubg=True`也可以开启`debug`模式。

如：

app = Flask(\_\_name\_\_)

app.debug = **True**

pycharm2017适用 但pycharm2018不适用

3. 通过配置参数的形式设置DEBUG模式：`app.config.update(DEBUG=True)`。

如：

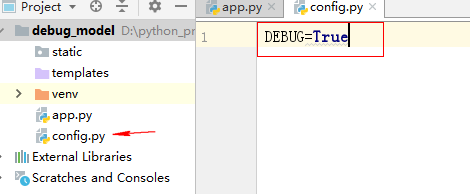
app = Flask(\_\_name\_\_)

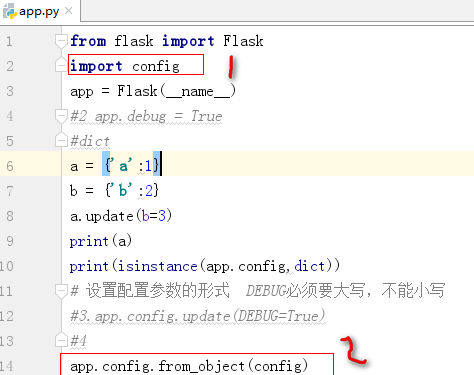
app.config.update(DEBUG=**True**)

pycharm2017适用 但pycharm2018不适用

4. 通过配置文件的形式设置DEBUG模式：`app.config.from\_object(config)`。

如：





pycharm2017适用 但pycharm2018不适用

**扩展点：了解即可**

如果想要在网页上调试代码，那么应该输入`PIN`码。