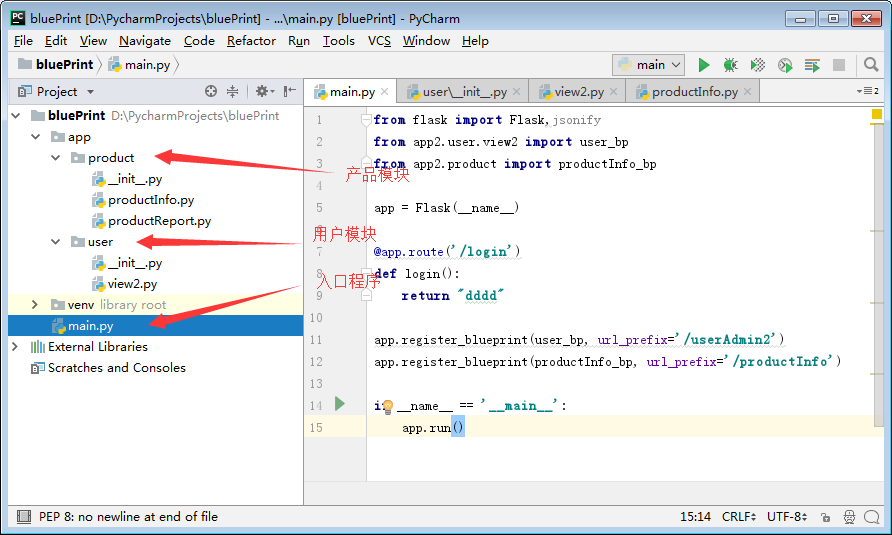
# 使用Blueprint（蓝图）创建模块化网站

大型网站需要模块化编程，通常是一个业务模块的相关代码都放到一个文件里。 Flask库里自带的Blueprint类可以方便的将网站划分为多个模块来进行编程。

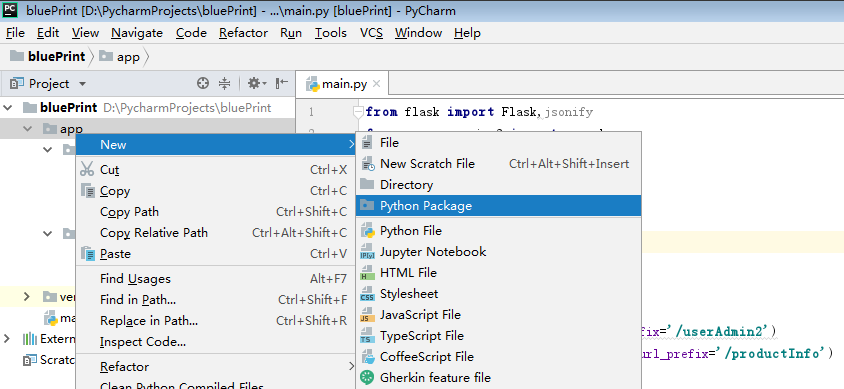
## 利用Blueprint规划网站结构



## 使用Blueprint

（1）、新建一个模块的总体文件夹app。也可以是其他名称。

（2）、右键单击app，在app下按业务功能（即模块）新建包。



新建包和新建文件夹的区别：新建包会自动产生一个空的\_\_init\_\_.py文件。

（3）、然后在某个包（可以理解为模块）下面建立程序文件。通常是views.py，也可以任意取名。

**from** flask **import** Blueprint  
  
user\_bp = Blueprint(**'userAdmin'**, \_\_name\_\_)  
  
@user\_bp.route(**'/'**, methods=[**'POST'**, **'GET'**])  
**def** index():  
 **return '<h1>Hello, this is admin blueprint</h1>'**

要使用Blueprint，首先要导入对象。

利用Blueprint类的构造函数创建一个“蓝图”对象。

然后利用创建的蓝图对象就可以设置路由，和利用Flask类构造的Flask对象设置路由的规则一样。

（4）、在主程序里注册蓝图对象

**from** app.user.view2 **import** user\_bp  
  
app = Flask(\_\_name\_\_)  
app.register\_blueprint(user\_bp, url\_prefix=**'/user'**)

**if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 app.run()

register\_blueprint为注册函数，第一个参数为蓝图对象，第二个参数为访问路径的前缀名称。

因为蓝图对象是在具体模块的py文件里创建的，因此要使用它们，要先汇入相关模块的蓝图对象。

（5）、某个包下有多个py文件，可以在\_\_init\_\_.py文件里创建蓝图对象，然后其他py文件汇入它即可。

**from** flask **import** Blueprint  
  
productInfo\_bp = Blueprint(**'product'**, \_\_name\_\_)  
  
**from** app.product **import** productInfo  
**from** app.product **import** productReport

后面两行是导入该包的含有路由设置的模块，这样后面主程序才能访问到模块里的路由。

这样，在主程序里只需要汇入\_\_init\_\_.py文件里的这个蓝图对象即可，这样，其他py 文件就不用一一去汇入了。

**…**

**from** app.product **import** productInfo\_bp  
…  
app.register\_blueprint(productInfo\_bp, url\_prefix=**'/productInfo'**)

…

url\_prefix的名称不一定要和模块的文件夹名一样。