**动态数据可视化**

**一**：**创建 flask 项目；**

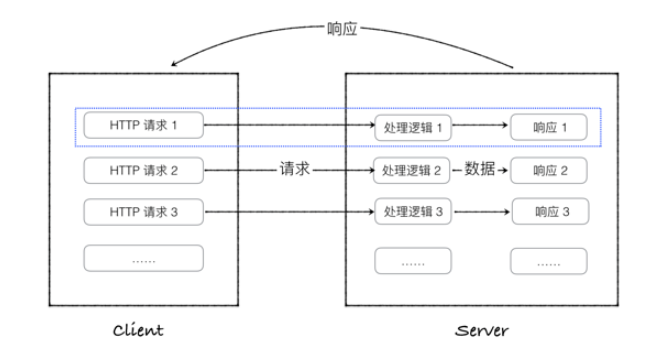
1: pip install flask (下载安装flask 库)

  安装Flask。 用户需要用管理员身份打开CMD，然后在期中输入pip install flask。当然，如果安装时提示pip版本过低，也可以采用pip install –upgrade pip指令来对pip进行更新。

二：**配置路由：**

1：认识 flask的原理：

1.1 http://docs.jinkan.org/docs/flask/ flask 官方文档；

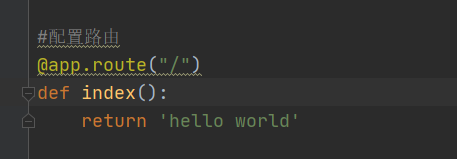


Flask本身是python中的后端框架，也是web服务端后端轻量级框架，可定 制性，来增加flask的功能；

2:实现我的第一个hello world

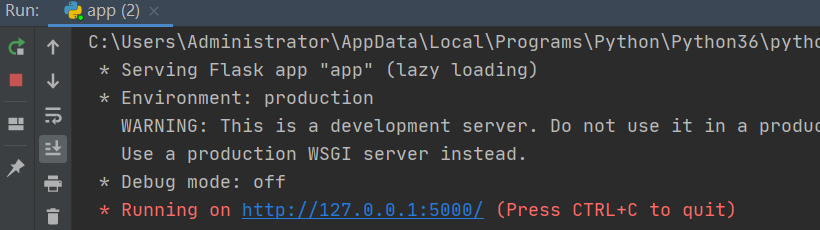
2.1：想要实现 在页面中展现 数据，那么我们得配置路由

什么是路由，路由就是（页面来了数据，将这个数据进行分配）如图：

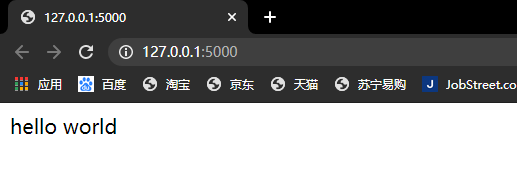


再次打开服务，进入服务器放开的地址

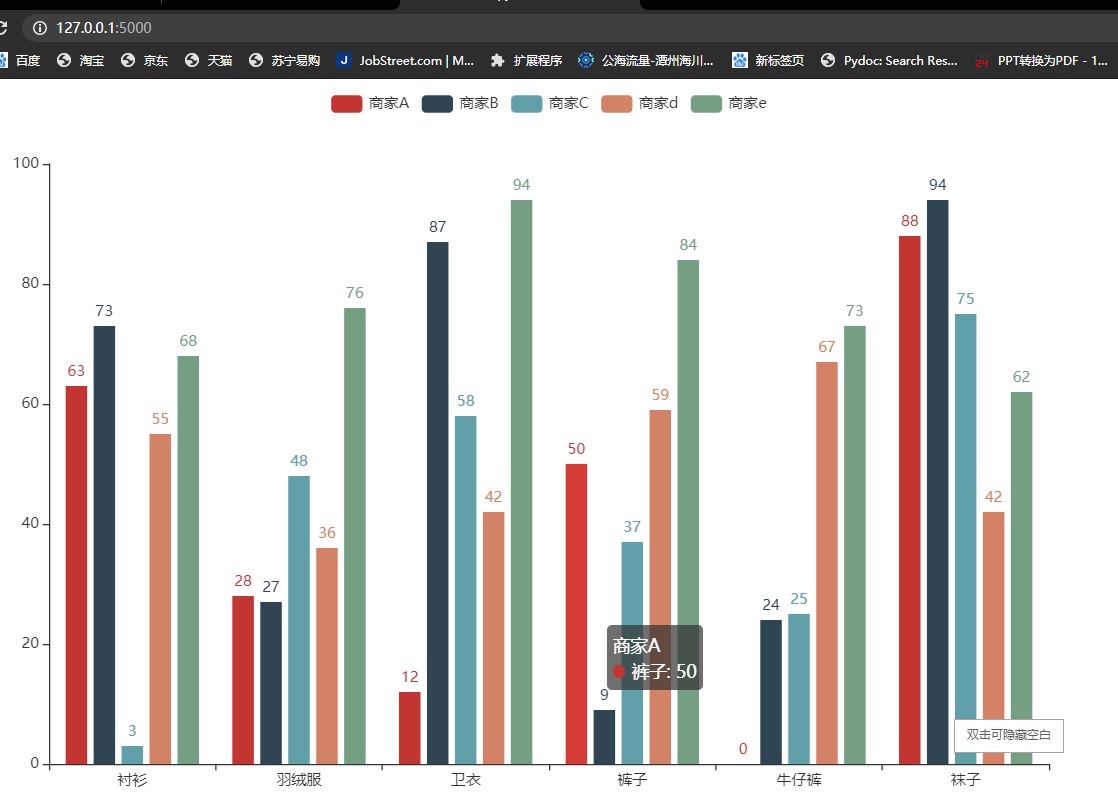
2.2： 那么页面是如何访问我们写好的代码的呢？



2.3：进入页面后 就显示好啦；



1.3：原来flask已经为我们集成好了 后端服务，这里的服务就是 服务器，对没错，一行代码 创建服务器， 我们可以通过 [http://127.0.0.1:5000/ 在访问到](http://127.0.0.1:5000/访问到) 我们写的页面，当然也可以看到我们写的页面数据；



可以看见我们的数据是从服务 地址中拿的 那么这里的 服务器地址是什么呢？其实这个地址就是 服务器在下面放开的端口 ，browser 通过地址就可以访问到他的 数据，然后将数据渲染出来；

**三：实践部分；**

1：后端功能代码：

总结：

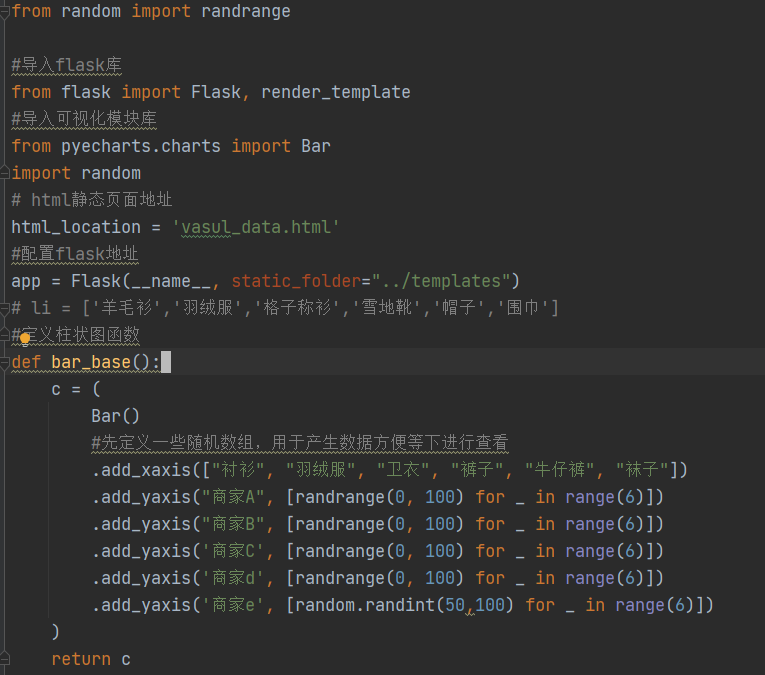
明白什么是后端

**第一部分:**

创建 flask app 将生成的数据传递到前端;

从前端向后端请求数据

1：生成数据





页面请求部分;



**第二部分： 实现动态数据展示；**

**编写异步请求从前端向后端请求数据**

编写服务端产生的数据；

和之前一致;

**页面部分;**

**# 引入js文件**

<script src="https://cdn.bootcss.com/jquery/3.0.0/jquery.min.js"></script>

<script type="text/javascript" src="https://assets.pyecharts.org/assets/echarts.min.js"></script>

# body部分；

**<h1>案例二动态请求数据</h1>**

**<div id="bar" style="width:1000px; height:600px;"></div>**

**<script>**

**<!--设置每次节点请求的时间-->**

**var chart = echarts.init(document.getElementById('bar'), 'white', {renderer: 'canvas'});**

**$(**

**function () {**

**fetchData(chart);**

**setInterval(fetchData,1800);**

**}**

**);**

**<!--请求的位置 url：请求的位置;dataType：请求的数据类型;-->**

**function fetchData() {**

**$.ajax({**

**type: "GET",**

**url: "http://127.0.0.1:5000/barChart",**

**dataType: 'json',**

**success: function (result) {**

**chart.setOption(result);**

**}**

**});**

**}**

**</script>**

**最终效果;**

**动态展示;**

