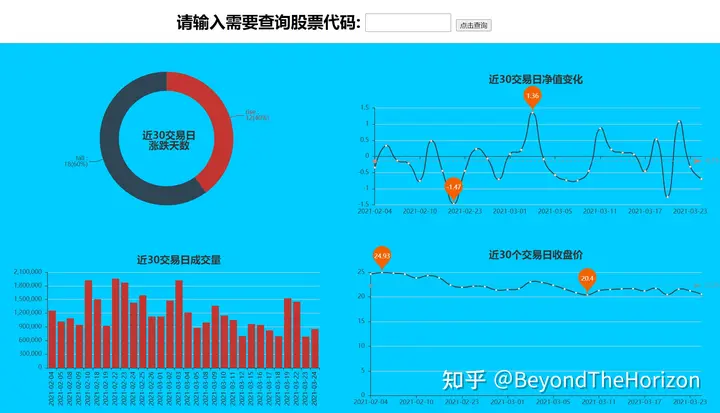
<https://www.zhihu.com/tardis/bd/art/359678550> 总体网站

**如何通过python获取股票数据并展示出来？**

效果图如下：



可以看出最近的股市跌的都比较厉害啊，越是跌，我们越是需要多予以关注

如何实现上述效果图呢？

我们计划分为3部分：

1、抓取数据并进行处理

2、将数据传递给前短展示页面

3、前端进行展示

**通过本文，你将学习到如下知识点：**

1. **如何使用python采集财经数据**
2. **如何加工采集到的数据**
3. **如何使用可视化面板将数据进行大屏展示**

第一部分收集数据部分程序为：getdata.py,代码详情如下：

这里引用了tushare模块，所以需要进行一下安装

pip install tushare

有兴趣的同学可以关注一下tushare网站：

[Tushare -财经数据接口包](https://link.zhihu.com/?target=http://tushare.org/" \t "https://www.zhihu.com/tardis/bd/art/_blank)

正式部分代码如下：

**import** tushare **as** ts

**def** **get\_data**(code):

dict\_return **=** {} *# 存放需要的数据*

data **=** ts**.**get\_hist\_data(code) *# 通过股票代码获取股票最近的数据*

data\_30 **=** data[:30]**.**iloc[::**-**1] *# 按照日期正序排列数据*

data\_30['rise'] **=** data\_30['price\_change'] **>** 0 *# 涨*

data\_30['fall'] **=** data\_30['price\_change'] **<** 0 *# 跌*

close **=** data\_30['close'] *#最近30个交易日的收盘价*

close\_index **=** list(close**.**index) *# 收盘价x轴数据*

close\_value **=** close**.**values**.**tolist() *# 收盘价y轴数据*

df\_diff **=** data\_30[['rise','fall']]**.**sum() *# 统计近30交易日的涨跌次数*

df\_diff\_index **=** list(df\_diff**.**index) *# 将数据转为列表格式*

df\_diff\_value **=** df\_diff**.**values**.**tolist() *# 将数据转为列表格式*

dict\_return['diff'] **=** [{"name":item[0],"value":item[1]} **for** item **in** list(zip(df\_diff\_index,df\_diff\_value))] *# 将数据制作成饼图需要的数据格式*

price\_change **=** data\_30['price\_change']**.**values**.**tolist() *# 统计近30交易日的价格变化*

volume **=** data\_30['volume']**.**values**.**tolist() *# 统计近30交易日的成交量*

*# 以下为将处理好的数据加入字典*

dict\_return['close\_index'] **=** close\_index

dict\_return['close\_value'] **=** close\_value

dict\_return['price\_change'] **=** price\_change

dict\_return['volume'] **=** volume

dict\_return['df\_diff\_index'] **=** df\_diff\_index

**return** dict\_return

这里面就1个函数，非常简单，通过引用tushare模块，获取到股票代码对应的30天内交易数据并生成字典

第二部分为数据的查询部分，代码为query.py,代码详情如下：

这里我们使用了轻量化的python web框架flask, 所以需要进行一下安装

pip install flask

正式代码如下：

**from** flask **import** Flask, request, render\_template**from** getdata **import** get\_data

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

@app**.**route('/query/', methods**=**['GET', 'POST'])**def** **query**():

**if** request**.**method **==** 'POST':

code **=** request**.**form**.**get('name')

dict\_return **=** get\_data(code)

**return** render\_template('query.html', dict\_return **=** dict\_return)

**else**:

dict\_return **=** get\_data('601318')

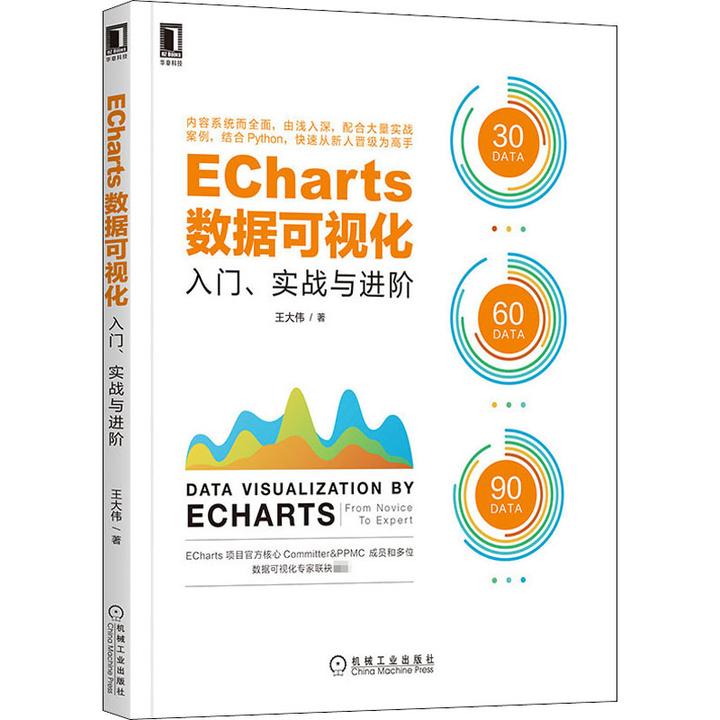
**return** render\_template('query.html', dict\_return **=** dict\_return)

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

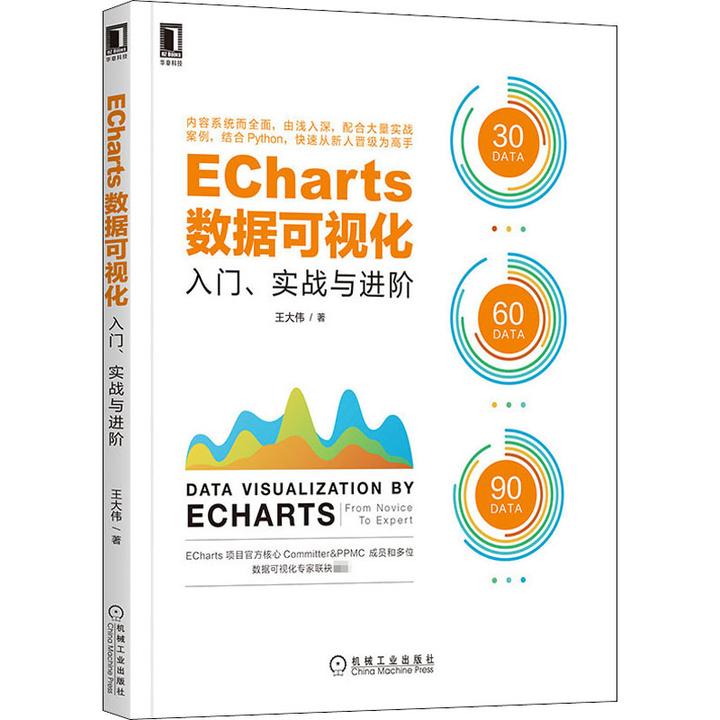
app**.**run(debug **=** **True**)

第三部分为前端展示部分，使用了html及echarts技术

如果有想了解html及echarts的同学，可以看看如下的书籍，本人看完后真的是受益匪浅啊



IMG_258



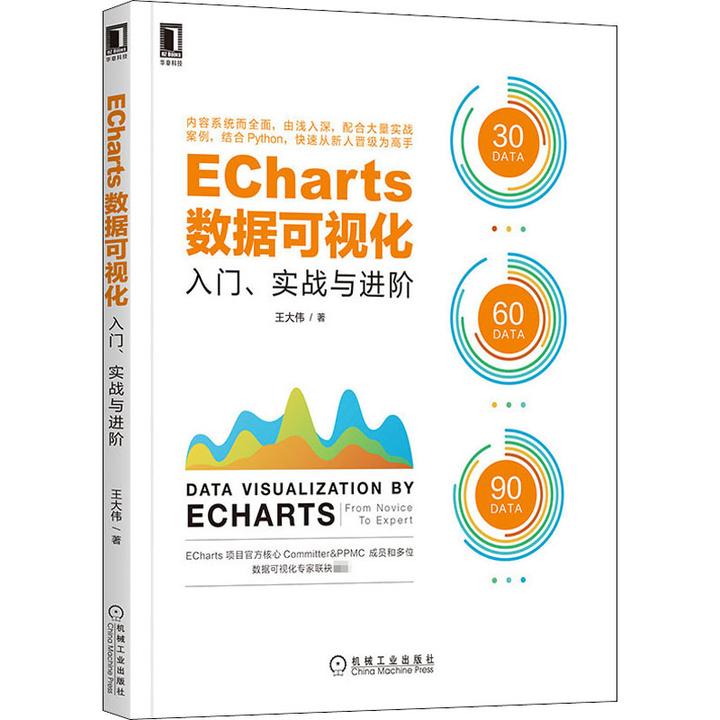
**ECharts数据可视化 入门、实战与进阶**

京东

¥63.00

**去购买**

IMG_260



IMG_262

echarts首页如下：

[Apache ECharts](https://link.zhihu.com/?target=https://echarts.apache.org/zh/index.html" \t "https://www.zhihu.com/tardis/bd/art/_blank)



下面是前端的查询展示页代码：query.html

**<!DOCTYPE html>**<head><meta charset**=**"utf-8"><title>股票实时查询产品DEMO</title><script src**=**"{{ url\_for('static',filename='echarts.js') }}"></script></head><body>

*<!-- 为ECharts准备一个具备大小（宽高）的Dom -->*

<center>

<form id**=**"form" name**=**"form" method**=**'POST' action**=**'/query/' style**=**"text-alige:left">

<h1>请输入需要查询股票代码:

<input type**=**"text" name**=**"name" style**=**"height:30px;width:160px;font-size:30px;">

<input type**=**"submit" value**=**"点击查询" ><h1>

</form>

<div id**=**"main" style**=**"width:1500px;height:750px;alige:center"></div>

*<!-- ECharts单文件引入 -->*

<script type**=**"text/javascript">

*// 基于准备好的dom，初始化echarts图表* **var** myChart **=** echarts.init(document.getElementById('main'));

**var** option **=** {

title**:** [

{

text**:** '近30交易日\n涨跌天数',

x**:** '25%',

y**:** '25%',

textAlign**:** 'center',

textBaseline**:** 'middle',

textStyle**:** {

fontSize**:** 20

}

},

{

text**:** '近30交易日净值变化',

x**:** '73%',

y**:** '9%',

textAlign**:** 'center',

textBaseline**:** 'middle',

textStyle**:** {

fontSize**:** 20

}

},

{

text**:** '近30交易日成交量',

x**:** '20.8%',

y**:** '55%',

textStyle**:** {

fontSize**:** 20

}

},

{

text**:** '近30个交易日收盘价',

x**:** '73%',

y**:** '55%',

textAlign**:** 'center',

textBaseline**:** 'middle',

textStyle**:** {

fontSize**:** 20

}

},

],

tooltip**:** {

trigger**:** 'axis',

axisPointer**:** {

type**:** 'shadow'

}

},

backgroundColor**:**'#00ccff',

grid**:** [{

left**:** '5%',

right**:** '55%',

top**:** '60%',

bottom**:** '5%',

containLabel**:** **true**

},

{

gridindex**:** 1,

left**:** '50%',

right**:** '5%',

top**:** '60%',

bottom**:** '5%',

containLabel**:** **true**

},

{

gridindex**:** 2,

left**:** '50%',

right**:** '5%',

top**:** '17%',

bottom**:** '55%',

containLabel**:** **true**

}

],

xAxis**:** [

{

type**:** 'category',

data**:** {{dict\_return['close\_index']**|**tojson}},

axisLabel**:** {

interval**:** 0,

rotate**:**90

},

},

{

gridIndex**:** 1,

type**:** 'category',

boundaryGap**:** **false**,

data**:** {{dict\_return['close\_index']**|**tojson}}

},

{

gridIndex**:** 2,

type**:** 'category',

boundaryGap**:** **false**,

data**:** {{dict\_return['close\_index']**|**tojson}}

}

],

yAxis**:** [

{

type**:** 'value',

axisLabel**:** {

formatter**:** '{value} '

},

boundaryGap**:** [0, 0.02]

},

{

gridIndex**:** 1,

type**:** 'value',

axisLabel**:** {

formatter**:** '{value} '

}

},

{

gridIndex**:** 2,

type**:** 'value',

axisLabel**:** {

formatter**:** '{value} '

}

}

],

series**:** [{

name**:** '',

type**:** 'bar',

label**:** {

normal**:** {

show**:** **true**,

position**:** 'top',

rotate**:**90,

show**:** **false**

}

},

data**:** {{dict\_return['volume']**|**tojson}}

},

{

name**:** '',

type**:** 'pie',

center**:** ['25%', '25%'],

radius**:** ['25%', '35%'],

label**:** {

normal**:** {

formatter**:** '{b} :\n{c}({d}%)'

}

},

data**:** {{dict\_return['diff']**|**tojson}}

},

{

xAxisIndex**:** 1,

yAxisIndex**:** 1,

name**:** '',

type**:** 'line',

lineStyle**:** {

normal**:** {

color**:** ''

}

},

*//data: ['416', '382', '318', '184', '215', '265', '557', '954', '1627', '1180', '2416', '2678', '3021', '2590','2100','1809','2300','2539',],* data**:** {{dict\_return['close\_value']**|**tojson}},

smooth**:** **true**,

markPoint**:** {

data**:** [{

type**:** 'max',

name**:** '最大值',

symbolSize**:** 60

},

{

type**:** 'min',

name**:** '最小值',

symbolSize**:** 60

}

],

itemStyle**:** {

normal**:** {

color**:** '#F36100'

}

}

},

markLine**:** {

data**:** [{

type**:** 'average',

name**:** '平均值'

}]

}

},

{

xAxisIndex**:** 2,

yAxisIndex**:** 2,

name**:** '',

type**:** 'line',

lineStyle**:** {

normal**:** {

color**:** ''

}

},

*//data: ['416', '382', '318', '184', '215', '265', '557', '954', '1627', '1180', '2416', '2678', '3021', '2590','2100','1809','2300','2539',],* data**:** {{dict\_return['price\_change']**|**tojson}},

smooth**:** **true**,

markPoint**:** {

data**:** [{

type**:** 'max',

name**:** '最大值',

symbolSize**:** 60

},

{

type**:** 'min',

name**:** '最小值',

symbolSize**:** 60

}

],

itemStyle**:** {

normal**:** {

color**:** '#F36100'

}

}

},

markLine**:** {

data**:** [{

type**:** 'average',

name**:** '平均值'

}]

}

}

]

};

*// 为echarts对象加载数据*  myChart.setOption(option);

</script>

</center></body></html>

为了能够正确的使用echarts绘制出我们所需要的图表，我们还需要提前在echarts网站把js文件下载下来哦

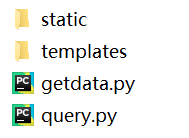
怎么下载呢？

[下载 - Apache ECharts](https://link.zhihu.com/?target=https://echarts.apache.org/zh/download.html" \t "https://www.zhihu.com/tardis/bd/art/_blank)

本人推荐使用在线定制的方式下载我们的js文件哦，因为这样下载下来的图表类型最丰富了



好的说到这里，我们再讲一下，整个工程的框架

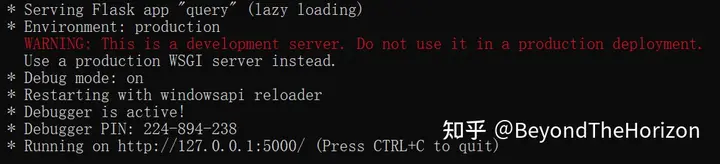


其中static文件夹存放我们从echarts网站下载好的js文件

templates文件夹存放我们的前端展示页面文件query.html

如何让程序跑起来呢，我们只需要在当前页面下打开cmd窗口，执行如下语句即可：

python query.py



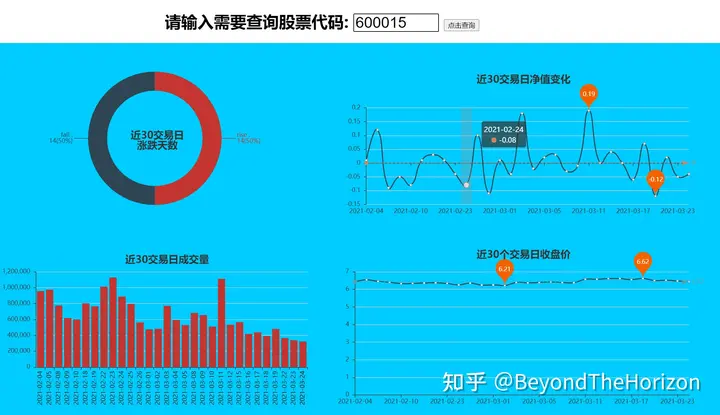
如图，此时我们只需要在浏览器里输入：

[http://127.0.0.1:5000/query](https://link.zhihu.com/?target=http://127.0.0.1/query" \t "https://www.zhihu.com/tardis/bd/art/_blank)

就可以查看效果了，非常方便，比如查看华夏银行的数据

股票代码：600015

查询后获得的结果如下：



本文的信息量还是比较大的，如果没有相应基础的同学看起来应该有些吃力

不过没有关系，我给你准备了如下基本书籍，可以帮助你轻松入门哦

最后赠读者们一句话：**今天的智力投资，都会是日后宝贵的财富、本钱！**



IMG_269



**HTML5+CSS3+JavaScript从入门到精通（标准版） web前**

京东

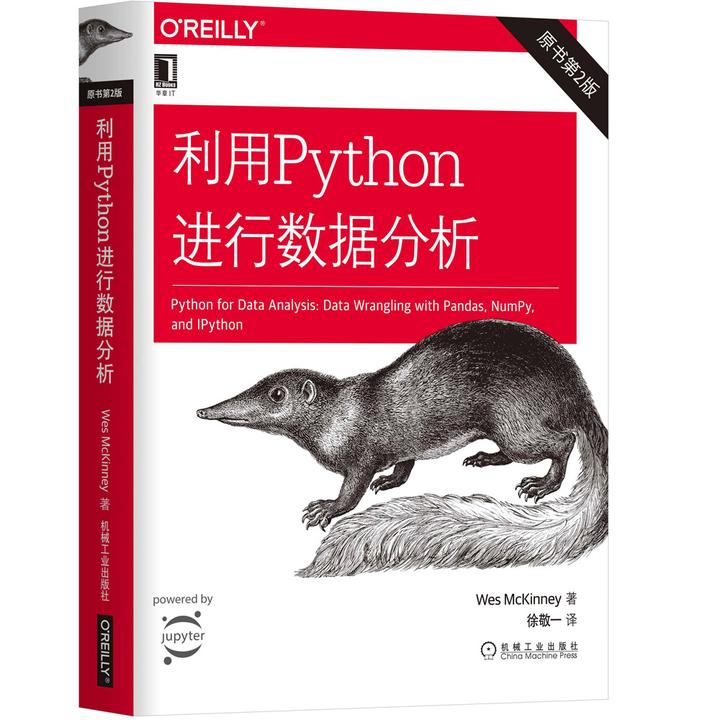
¥67.40

**去购买**

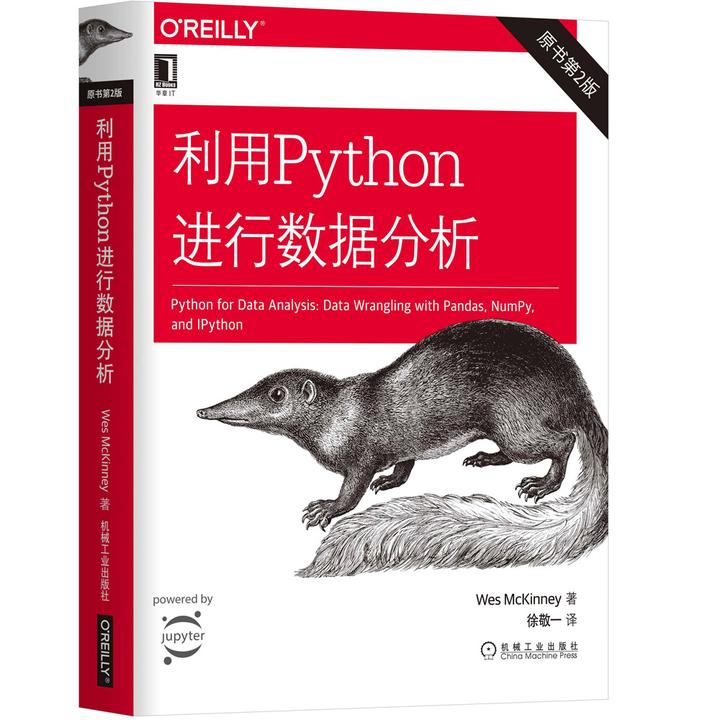
IMG_271



IMG_273



IMG_275



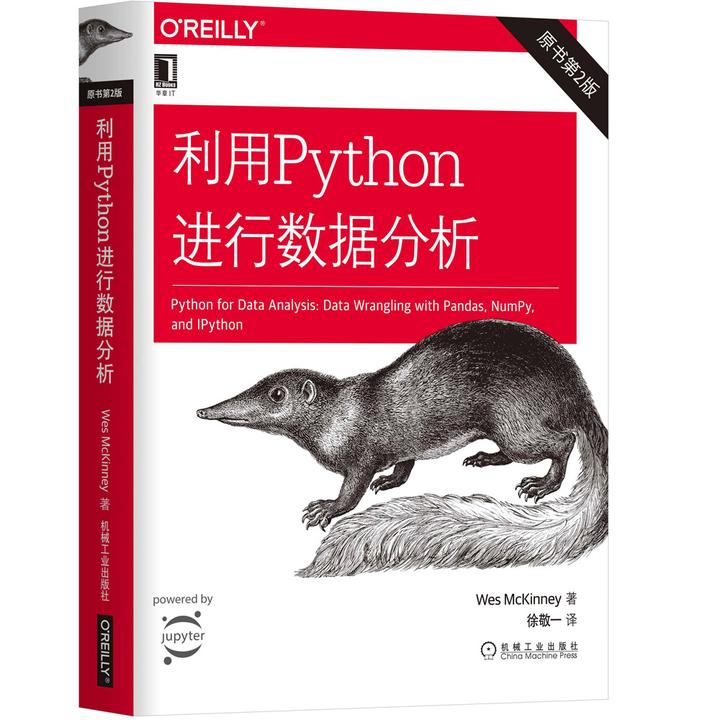
**利用Python进行数据分析（原书第2版）**

京东

¥98.20

**去购买**

IMG_277



IMG_279



IMG_281



**大数据可视化大屏html页面模板源码echarts项目jquery**

淘宝

¥48.00

**去购买**

IMG_283



IMG_285

编辑于 2021-03-25 · 著作权归作者所有

### **新建templates文件夹，与py文件保持同一级，将html文件放进templates文件夹内.否则“jinja2.exceptions.TemplateNotFound”**

1、项目下面是否有templates文件夹，你的html文件是否放进了里面；

2、templates文件夹是否和你运行的py文件在同一级目录；

3、render\_template('\*\*\*.html')这里面的名字是否正确，别打错了；

4、app = Flask(\_\_name\_\_, template\_folder='templates', static\_folder="\*\*\*\*",static\_url\_path="\*\*\*\*") 在最开始的这句话中，template\_folder后面一定要跟上templates；