



Python

旅游规划小助手

Python2班 第N小组

何智钧

林元靖

陈宛佳

contents

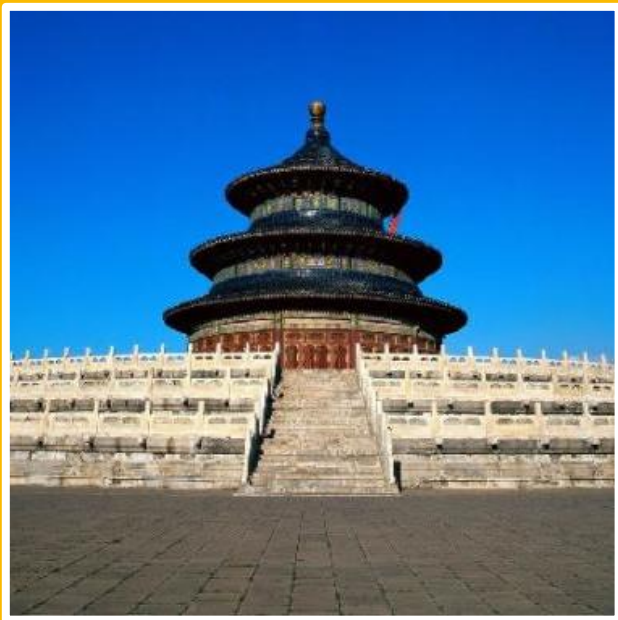
- 项目意义及解决的问题
- 实现步骤及代码呈现
- 网页效果展示
- 收获及改进反思



旅游

意义：用户广泛&意义持久

解决的问题：路线规划与及时调整



实现步骤

1. 爬取城市介绍
2. 爬取热门景点并可视化
3. 规划最短路径
4. 爬取景点经纬度
5. 将景点与最短路径可视化
6. 爬取旅社信息
7. 爬取城市未来天气情况



实现代码：爬取城市介绍

Beautifulsoup: 百度百科城市介绍

```
108 #搜索城市信息
109 class CityInfo(models.Model):
110     def __init__(self, city_name):
111         self.city_name=city_name
112     def getCityinfo(self):
113         #生成网址
114         url = 'https://baike.baidu.com/item/{}'.format(parse.quote(self.city_name))
115         header = {"User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.3
116         r = requests.get(url, headers=header)
117         #提取城市简介所在部分的html
118         soup = BeautifulSoup(r.text, "html.parser")
119         html = soup.select(".lemma-summary")
120         # 提取文字并转为字符串
121         raw_string = "".join([x.get_text() for x in html])
122         #去除所有空白字符
123         new_string = raw_string.split()
124         #去除百度百科内容的脚注符号并重组回字符串
125         clear = re.compile(r"\\d*\\W*\\d*\\")
126         city_intro = "".join([i for i in new_string if not clear.search(i)])
127         return city_intro
128
```

实现代码：爬取热门景点

```
#从马蜂窝搜集热门景点游记与点评
class HotDestination(models.Model):
    def __init__(self,city,path_svg,path_html):
        self.city = city
        self.path_svg = path_svg
        self.path_html = path_html
        #准备一个url模板
        self.url_pattern="http://www.mafengwo.cn/search/q.php?q="+self.city+"&p={0}&t=pois&kt=1"
        #准备请求头
        self.headers={"User-Agent":"Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML,

#获取前page_number页的url列表
def get_url_list(self,page_number):
    url_list=[]
    for i in range(1,page_number+1):
        url=self.url_pattern.format(i)
        url_list.append(url)
    return url_list

#根据url发送请求，获取数据
def get_page(self,url):
    response = requests.get(url,headers=self.headers)
    return response.content.decode()
```

Requests: 马蜂窝

实现代码：爬取热门景点

#解析页面数据

```
def get_datas(self, page):
    element = etree.HTML(page)
    #定义空列表，保存后面的数据
    data_list=[]

    #1.获取标签列表
    lis=element.xpath("//*[@id='_j_search_result_left']/div/div/ul/li")

    #2.遍历标签列表
    for li in lis:
        #提取景点名称
        #定义一个字典用于保存数据
        item={}
        name="".join(li.xpath("./div/div[2]/h3/a//text()"))
        if name.find("景点")==-1:
            continue
        #删除景点括号内的英文名
        name = re.sub("\\ (..*?\\)", "", name)
        item["name"]=name.replace("景点 - ", "")

        #获取点评数量
        comment=li.xpath("./div/div[2]/ul/li[2]/a//text()")[0]
        #截取其中的数字
        comment_1=comment.lstrip("蜂评(")
        comment_2=comment_1.rstrip(")")
        item["comment_number"]=int(comment_2)
```

Xpath: 解析页面数据

(景点名称、评论数量、游记数量)

#获取游记数量

```
article=li.xpath("./div/div[2]/ul/li[3]/a//text()")[0]
#截取其中的数字
article_1=article.lstrip("游记(")
article_2=article_1.rstrip(")")
item["article_number"]=int(article_2)
```

#保存在一开始定义的空列表中

```
data_list.append(item)
```

```
return data_list
```

实现代码：热门景点可视化

```
#排序与制图
def present(self, data_list, destination_number):
    #3. 根据评论数量进行排序
    se_list=sorted(data_list, key=lambda x:x['comment_number'], reverse=True)

    #4. 做出柱状图
    #定义图表形式与框架
    #使用NeonStyle, 渐变特效多
    my_style=pygal.style.RotateStyle("#FF7F00", base_style=NeonStyle)
    my_style.title_font_size=20
    my_style.label_font_size=10
    my_style.major_label_font_size=12
    my_style.legend_font_size=16
    my_style.tooltip_font_size=24
    #tooltip_border_radius使得tip框四个角变圆滑
    destination_bar=pygal.Bar(style=my_style, tooltip_border_radius=10)
    destination_bar.title=self.city+"热门景点点评与游记数量"
    destination_bar.width=1200
    #X轴景点名向右旋转30度
    destination_bar.x_label_rotation=30
    #X轴景点名最多显示5个字, 省略者鼠标移至上方会显示全名
    destination_bar.truncate_label=5
    #主标签间隔
    destination_bar.x_labels_major_every=1
    destination_bar.y_labels_major_every=3
    #添加数据
    destinations=[]
    comments=[]
    articles=[]
```

对景点排序（用点评数量反映热门程度）

NeonStlye: 画图
(景点名称、评论数量、
游记数量)

实现代码：热门景点可视化

```
# 搜到的景点大于等于destination_number个时
if len(se_list) >= destination_number:
    i = 0
    while i < destination_number:
        # 加入百度地图链接
        comments.append({"value": se_list[i]['comment_number'], "xlink": self.getLocation(se_list[i]['name'])})
        articles.append({"value": se_list[i]['article_number'], "xlink": self.getLocation(se_list[i]['name'])})
        destinations.append(se_list[i]['name'])
        i += 1
    # 使用双纵坐标
    # 标签部分还加入提示
    destination_bar.add({"title": "点评数", "tooltip": "Comments"}, comments)
    destination_bar.add({"title": "游记数", "tooltip": "Travels"}, articles, secondary=True)
    destination_bar.x_labels = destinations
# 搜到的景点不足destination_number个时
else:
    for i in range(0, len(se_list)):
        # 加入百度地图链接
        comments.append({"value": se_list[i]['comment_number'], "xlink": self.getLocation(se_list[i]['name'])})
        articles.append({"value": se_list[i]['article_number'], "xlink": self.getLocation(se_list[i]['name'])})
        destinations.append(se_list[i]['name'])
    # 使用双纵坐标
    # 标签部分还加入提示
    destination_bar.add({"title": "点评数", "tooltip": "Comments"}, comments)
    destination_bar.add({"title": "游记数", "tooltip": "Travels"}, articles, secondary=True)
    destination_bar.x_labels = destinations
# 保存图表
destination_bar.render_to_file(self.path_svg)
```

加入百度地图链接

实现代码：最短路径的计算

1、对选定景点进行全排列→

```
for x in pailie:
    path1 = []
    #对用户是否输入起点和终点进行判断
    if self.startpoint in self.chosen_spots and self.endpoint in self.chosen_spots:
        startpoint_num = self.chosen_spots.index(self.startpoint) # 起点
        endpoint_num = self.chosen_spots.index(self.endpoint) # 终点
        if x[0] == startpoint_num and x[len(chosen_location) - 1] == endpoint_num:
            for y in x:
                path1.append(y)
            all_path.append(path1)
    elif self.startpoint in self.chosen_spots and self.endpoint not in self.chosen_spots:
        startpoint_num = self.chosen_spots.index(self.startpoint) # 起点
        if x[0] == startpoint_num:
            for y in x:
                path1.append(y)
            all_path.append(path1)
    elif self.endpoint in self.chosen_spots and self.startpoint not in self.chosen_spots:
        endpoint_num = self.chosen_spots.index(self.endpoint) # 终点
        if x[len(chosen_location) - 1] == endpoint_num:
            for y in x:
                path1.append(y)
            all_path.append(path1)
    else:
        for y in x:
            path1.append(y)
        all_path.append(path1)
```

```
#计算最短路径
def getPath(self,chosen_location):
    #景点名形式的最短路径
    bestway_num = []
    bestway = []

    #生成景点的全排列
    array = list(range(len(chosen_location)))
    all_path = []
    pailie = list(itertools.permutations(array))
```

←2、筛选（考虑到
起点和终点的限制）

实现代码：最短路径的计算

```
i = 0
distance2 = 1000000000000000 # 只需要规定一个足够大的数就行
while i < len(all_path):
    distance1 = 0
    for j in range(len(chosen_location) - 1):
        # 计算两点间距离
        sight1 = chosen_location[all_path[i][j]]
        sight2 = chosen_location[all_path[i][j + 1]]
        distance = (float(sight1[0]) - float(sight2[0])) ** 2 + (float(sight1[1]) - float(sight2[1])) ** 2
        distance = math.sqrt(distance)
        # 计算整条路径距离
        distance1 += distance
    # 选取最短路径
    if distance1 < distance2:
        distance2 = distance1
        bestway_num = all_path[i]
    i += 1

# 对应景点名
for i in bestway_num:
    bestway.append(self.chosen_spots[i])

# 返回最佳路线
return bestway
```

3、计算最短路径输出

(利用经纬度作为坐标，计算
每条满足要求路线的总距离)

实现代码：爬取推荐路线

爬取途牛网上满意度最高的旅游团旅行路线（为用户推荐），爬取过程与马蜂窝类似，但考虑到存在许多旅游团的路线相同，最后多加上了去掉重复路线的代码

```
#去掉重复路线
item=[]
if len(data_list)>2:
    for i in range(len(data_list)-2):
        try:
            trying=data_list.index(data_list[i],i+1)
        except:
            item.append(data_list[i])
        else:
            pass
    item.append(data_list[-1])
else:
    for i in data_list:
        item.append(i)
return item
```

实现代码：爬取经纬度

利用百度地图开放平台爬取景点的经纬度，并与景点名称一一对应，装入列表以备调用

```
# 遍历景点列表
for i in data_lists:
    # 获取经纬度
    try:
        ak = "SmC9G6mGN1TNQwrh5R804qT6ccMIxBol"
        url = "http://api.map.baidu.com/place/v2/search?query={0}&region={1}&output=json&ak={2}".format(
            parse.quote(i["name"]), parse.quote(self.city), ak)
        res = requests.get(url)
        json_data = json.loads(res.text)
        lng = float(json_data["results"][0]["location"]["lng"])
        lat = float(json_data["results"][0]["location"]["lat"])
        # 若不能得到经纬度则忽略这个景点
    except:
        pass
    else:
        sights.append(i["name"])
        # 热门指数=点评数*10%+游记数*90%
        values.append(round(i["comment_number"] * 0.1 + i["article_number"] * 0.9, 1))
        locations[i["name"]] = [lng, lat]
        information_list.append((self.city, i["name"]))
```

实现代码：制作景点分布图

利用pyecharts库的城市地图，添加景点标记点，并利用景点的热度值对标记点的颜色进行区分

```
#生成地图
def getMap(self, sights, values, locations):
    labels = zip(sights, values)
    #地图初始化
    sights_map = Geo(init_opts=opts.InitOpts(width='1140px', height='600px', bg_color = "pink"))
    #选择城市地图
    if self.city=="中西区" or self.city=="东区" or self.city=="九龙城区" or self.city=="南区" or self.city=="深水埗区" or self.city=="湾仔区" or self.city=="黄大仙区":
        sights_map.add_schema(maptype="香港")
    elif self.city=="圣安多尼堂区" or self.city=="望德堂区" or self.city=="风顺堂区" or self.city=="圣方济各堂区" or self.city=="嘉模堂区" or self.city=="大堂区" or self.city=="圣老楞佐堂区":
        sights_map.add_schema(maptype="澳门")
    elif self.city=="台北市" or self.city=="高雄市" or self.city=="台北县" or self.city=="台中县" or self.city=="台南县" or self.city=="台南市" or self.city=="桃园县":
        sights_map.add_schema(maptype="台湾")
    else:
        sights_map.add_schema(maptype=self.city.rstrip("市"))
    #添加景点标记点
    for sight in sights:
        sights_map.add_coordinate(sight, locations[sight][0], locations[sight][1])
    sights_map.add("热门指数=点评数*10%+游记数*90%", labels, type_="effectScatter")
    #不显示位置标签，加入涟漪特效
    sights_map.set_series_opts(label_opts=opts.LabelOpts(is_show=False),
                               effect_opts=opts.EffectOpts(is_show=True, brush_type="stroke"))
    #全局设置
    sights_map.set_global_opts(
        visualmap_opts=opts.VisualMapOpts(is_show=True, max_=max(values), min_=min(values)),
        title_opts=opts.TitleOpts(
```



实现代码： 规划路线可视化

在原本景点分布图的基础上，添加为用户规划的路线的连线展示

```
try:
    sights_map.add("最佳路线", best_route, type_=ChartType.LINES,
                  effect_opts=opts.EffectOpts(symbol=SymbolType.ARROW, symbol_size=6, color="blue"),
                  linestyle_opts=opts.LineStyleOpts(curve=0.2))
#若用户只输入1个地点，则不连线
except:
    pass
```


实现代码： 爬取旅社信息

爬取木鸟民宿网站上的信息，利用while循环进行翻页，
爬取各城市所有旅社的信息

```
try:
    r = requests.get(url, headers=header)
    soup = BeautifulSoup(r.text, "html.parser")
    #超出范围便报错
    last_page = int(soup.find(class_ = "Lpage_li1")['title'])
except:
    break
else:
    page+=1
    #获取所有价格装入列表
    prices = soup.find_all(class_='s_mn_house_price2')
    prices_list.extend([int(float(re.sub("\D", '', price.get_text())) for price in prices])
```


实现代码： 旅社价格统计

```
#价格0-100 (含)
range0=0
#价格100-200 (含)
range1=0
#价格200-300 (含)
range2=0
#价格300-400 (含)
range3=0
#价格400-500 (含)
range4=0
#价格大于500
range5=0
for price in prices_list:
    if price<=100:
        range0+=1
    elif price<=200:
        range1+=1
    elif price<=300:
        range2+=1
    elif price<=400:
        range3+=1
    elif price<=500:
        range4+=1
    else:
        range5+=1
return range0, range1, range2, range3, range4, range5
```

划分6个价格区间，统计每个价格区间的旅社数量

实现代码： 旅社价格统计

```
def create_chart(self, *hotel_numbers):
    #定义图表形式与框架
    #使用NeonStyle, 渐变特效多
    my_style=pygal.style.RotateStyle("#FF0000",base_style=NeonStyle)
    my_style.title_font_size=20
    my_style.legend_font_size=16
    my_style.tooltip_font_size=21
    #tooltip_border_radius使得tip框四个角变圆滑
    pie_chart=pygal.Pie(style=my_style,inner_radius=0.5,tooltip_border_radius=10)
    pie_chart.width=1200
    pie_chart.legend_at_bottom=True
    pie_chart.legend_at_bottom_columns=6
    pie_chart.title=self.city+"旅社一宿价格分布"
    #标签部分加入提示和链接; 饼图部分加入链接
    pie_chart.add({'title': "100元（含）以下",'tooltip': 'Below ¥100','xlink':"/hostel-introduction/?city="+self.city+"&price=0-100"},
        [{"value":hotel_numbers[0],"xlink":"/hostel-introduction/?city="+self.city+"&price=0-100"}])
    pie_chart.add({'title': "100-200元（含）",'tooltip': '¥100-200','xlink':"/hostel-introduction/?city="+self.city+"&price=100-200"},
        [{"value":hotel_numbers[1],"xlink":"/hostel-introduction/?city="+self.city+"&price=100-200"}])
    pie_chart.add({'title': "200-300元（含）",'tooltip': '¥200-300','xlink':"/hostel-introduction/?city="+self.city+"&price=200-300"},
        [{"value":hotel_numbers[2],"xlink":"/hostel-introduction/?city="+self.city+"&price=200-300"}])
    pie_chart.add({'title': "300-400元（含）",'tooltip': '¥300-400','xlink':"/hostel-introduction/?city="+self.city+"&price=300-400"},
        [{"value":hotel_numbers[3],"xlink":"/hostel-introduction/?city="+self.city+"&price=300-400"}])
    pie_chart.add({'title': "400-500元（含）",'tooltip': '¥400-500','xlink':"/hostel-introduction/?city="+self.city+"&price=400-500"},
        [{"value":hotel_numbers[4],"xlink":"/hostel-introduction/?city="+self.city+"&price=400-500"}])
    pie_chart.add({'title': "500元以上",'tooltip': 'Above ¥500','xlink':"/hostel-introduction/?city="+self.city+"&price=500-"},
        [{"value":hotel_numbers[5],"xlink":"/hostel-introduction/?city="+self.city+"&price=500-"}])
    #保存图片
```

利用pygal库制作饼图，并在饼图的各个部分加入链接，用户通过链接可以查看到对应区间的旅社的信息

实现代码： 旅社信息分类

```
# 将各民宿信息分装入字典，并装入列表
hotel_list = []
for title, price, intro, address, web in zip(titles_list, prices_list, intro_list, address_list, webs_list):
    if self.price == "0-100":
        if not price <= 100:
            continue
    elif self.price == "100-200":
        if not (price > 100 and price <= 200):
            continue
    elif self.price == "200-300":
        if not (price > 200 and price <= 300):
            continue
    elif self.price == "300-400":
        if not (price > 300 and price <= 400):
            continue
    elif self.price == "400-500":
        if not (price > 400 and price <= 500):
            continue
    else:
        if not price > 500:
            continue
    hotel_dict = {}
    hotel_dict["民宿名称"] = title
    hotel_dict["价格"] = price
    hotel_dict["基本信息"] = intro
    hotel_dict["地址"] = address.lstrip("地址：")
    hotel_dict["网址"] = web
    hotel_list.append(hotel_dict)
return hotel_list
```

利用if条件判断，将
对应价格区间的旅
社的信息装入字典，
再汇总入列表

实现代码： 爬取天气信息

```
# 获取网址
def getUrl(self):
    with open("app/templates/data/weather_city_id.csv", encoding="ANSI") as csvfile:
        city_ids = {}
        reader = csv.DictReader(csvfile)
        for row in reader:
            city_ids[row["city"]] = row["id"]
    address = city_ids[self.city]
    ak = '5n1C61vP4gFprRIojBSvG7uVVb1UA2a6'
    url = 'http://api.map.baidu.com/weather/v1/?district_id={0}&data_type=all&ak={1}'.format(address, ak)
    return url

# 获得天气预报
def getForecasts(self, url):
    res = requests.get(url)
    json_data = json.loads(res.text)
    dates = []
    high_values = []
```

利用百度地图开放平台，爬取城市未来五日的气温和天气类型信息

实现代码： 爬取天气信息

```
high_values = []
low_values = []
weather_day = []
weather_night = []

if json_data['status'] == 0:
    for day in json_data["result"]["forecasts"]:
        # 日期
        dates.append(day['date'])
        # 最高温度
        high_values.append(float(day['high']))
        # 最低温度
        low_values.append(float(day['low']))
        # 白天天气
        weather_day.append(day['text_day'])
        # 晚间天气
        weather_night.append(day["text_night"])
# 返回列表
return dates, high_values, low_values, weather_day, weather_night
```

利用百度地图开放平台，爬取城市未来五日的气温和天气类型信息

实现代码：天气信息可视化

```
# 生成气温图表
def getChart(self, dates, highs, lows):
    # 定义图表形式与框架
    # 使用NeonStyle, 渐变特效多
    my_style = pygal.style.RotateStyle("#FF69B4", base_style=NeonStyle)
    my_style.title_font_size = 20
    my_style.label_font_size = 12
    my_style.major_label_font_size = 14
    my_style.legend_font_size = 16
    my_style.tooltip_font_size = 21
    # tooltip_border_radius使得tip框四个角变圆滑
    # XY散点图中使用埃尔米特插值法, 使折线图变得圆滑; 相比多项式插值法, 埃尔米特插值法还考虑到各阶导数相同, 圆滑程度更高
    temperature_line = pygal.XY(style=my_style, tooltip_border_radius=10, interpolate='hermite')
    temperature_line.width = 1200
    temperature_line.title = self.city + "未来五天气温变化"
    temperature_line.x_title = "日期"
    temperature_line.y_title = "气温(℃)"
    temperature_line.x_label_rotation = 0
    # 主标签间隔
    temperature_line.x_labels_major_every = 1
    temperature_line.y_labels_major_every = 3
    # 添加数据
    date_high = []
    date_low = []
    x_date = []
```

利用pygal库画城市
未来五日最高温和
最低温的折线图

实现代码：天气信息可视化

利用pygal库画城市未来五日最高温和最低温的折线图

```
for j in range(len(dates)):
    # 因为采用XY散点图形式，所以新增实际值标签
    date_high.append({"value": (j + 1, highs[j]), "label": dates[j]})
    date_low.append({"value": (j + 1, lows[j]), "label": dates[j]})
    # 因为采用XY散点图形式，所以X轴的标签和实际值不同（但可一一对应）
    x_date.append({"label": dates[j], "value": j + 1})
# stroke意味着连线（即可采用埃尔米特插值法了）；同时自定义连线的虚实交替长度
temperature_line.add({"title": "最高温度", "tooltip": "Highs"}, date_high, stroke=True, dots_size=3.5,
                    stroke_style={'width': 1.8, 'dasharray': "15,7,5,5,5,7", 'linecap': 'round',
                                   'linejoin': 'round'})
temperature_line.add({"title": "最低温度", "tooltip": "Lows"}, date_low, stroke=True, dots_size=3.5,
                    stroke_style={'width': 1.8, 'dasharray': "10,5", 'linecap': 'round', 'linejoin': 'round'})
temperature_line.x_labels = x_date
# 保存图表
temperature_line.render_to_file(self.path)
```

Travel Web

Travel Web is a free website for searching travel destinations while getting useful information.

Search for Hot Destinations »

Mafengwo

Mafengwo, the most popular travel website visited by the younger generation in China. Thanks to the core advantage of "content + trading", mafengwo understands the travel preferences of people better.

Learn more »

Baidu Map

Baidu Map is a platform to provide travel services including intelligent route planning, precise navigation, real-time traffic and public transportation, adhering to the mission of "technology makes travel easier".

Learn more »

Muniao

There are not only quadrangles, holiday apartments, sea view rooms, but also party villas, theme rooms and other special rooms, which not only makes you feel different from the hotel, but also greatly saves the cost of travel.

Learn more »

Search.

Select a city or region in China which you want to visit

Search

佛山市

佛山，简称“禅”，是广东省地级市，国务院批复确定的中国重要的制造业基地、珠江三角洲西翼经贸中心和综合交通枢纽。截至2018年，全市下辖5个区，总面积3797.72平方千米，常住人口790.57万人，城镇人口750.73万人，城镇化率94.96%。佛山地处中国华南地区、广东省中部、珠三角腹地，毗邻港澳、东接广州、南邻中山，与广州共同构成“广佛都市圈”，大力推进广佛同城化合作，打造国际大都市区，是珠江三角洲城市之一、粤港澳大湾区重要节点城市，“广佛肇经济圈”、“珠江—西江经济带”的重要组成部分。佛...

Search »

北京市

北京，简称“京”，古称燕京、北平，是中华人民共和国首都、省级行政区、直辖市、国家中心城市、超大城市，国务院批复确定的中国政治中心、文化中心、国际交往中心、科技创新中心。截至2018年，全市下辖16个区，总面积16410.54平方千米，2019年末，常住人口2153.6万人，城镇人口1865万人，城镇化率86.6%，常住外来人口达794.3万人。北京地处中国北部、华北平原北部，东与天津毗连，其余均与河北相邻，中心位置东经116°20′、北纬39°56′，是世界著名古都和现代化国际城市，也是中国共...

Search »

杭州市

杭州，简称“杭”，古称临安、钱塘，是浙江省省会、副省级市、杭州都市圈核心城市，国务院批复确定的中国浙江省省会 and 全省经济、文化、科教中心、长江三角洲中心城市之一。截至2019年，全市下辖10个区、2个县、代管1个县级市，总面积16853.57平方千米，常住人口1036万人，城镇人口813.26万人，城镇化率78.5%，常住外来人口达450.44万人。杭州地处中国华东地区、钱塘江下游、东南沿海、浙江北部、京杭大运河南端，是环杭州湾大湾区核心城市、沪嘉杭G60科创走廊中心城市、国际重要的电子商务中心...

Search »

← 首页

← 搜索页

网页效果展示

About.

Travel Web is a free website for searching travel destinations while getting useful information.

Considering the long-term and practical significance of the project, we chose the theme of tourism. The purpose of "going out for sightseeing" is to travel to a certain destination, that is to say, the purpose of traveling is to travel to a certain destination. As a way for people to broaden their horizons and relax their body and mind, tourism has a high universality. No matter children or middle-aged and old people can find their own tourism mode through different tourism modes, tourist attractions and tourist routes. Therefore, the theme of tourism can be said to be eternal.

We consider that many major tourism websites are about the information collection of scenic spots, hotels, transportation and tourism strategies, which is mainly applicable to the preliminary direction planning. However, due to the excessive information, it will take a long time to screen the information if we want to make a detailed planning. Therefore, our website reduces the presentation of traffic and strategy, adding the information of future weather conditions and the function of route planning, which solves the problem of finding the best route in the process of medium-term detailed planning to save the travel time to the greatest extent, and provides help for checking the weather in the next few days and adjusting the route in the process of later preparation and travel. We have searched and summarized the hotel information with good quality and low price, so as to facilitate users to understand the accommodation information.

There are some advantages of our website:

1. All modules are displayed on a website based on Django library, which is easy to operate and practical.
2. All information in this website is instant information, which is valuable for the users who are about to travel.
3. Hyperlinks are set up in the web pages of hotels and popular scenic spots to facilitate users to enter relevant websites to inquire more detailed information or make purchase decisions.
4. We give first priority to the interaction with users. Users can choose to view the relevant content of any city in the country; users can also choose the scenic spots they want to visit, and we provide them with the design of the best route; they can also choose to view the hotel information in a certain price range.
5. We also pay attention to the summary of information and statistics of data. Whether it is hot spots, hotel prices or weather conditions, we use scientific methods to make statistics, and use pygal and pyecharts libraries to design good-looking charts or maps for display.

© July 11, 2020, 7:47 p.m. - Travel Web

Contact.

Please contact us if you have any questions

Lingnan College, Sun Yat-sen University
No. 135, West Xingang Road
Haizhu District, Guangzhou
Tel: +86 2084114183

Zhijun He: hezhj9@mail2.sysu.edu.cn
Wanjia Chen: chenwj237@mail2.sysu.edu.cn
Yuanjing Lin: linyj59@mail2.sysu.edu.cn

© July 11, 2020, 7:49 p.m. - Travel Web

←关于我们页

←联系我们页

城市介绍

[Travel Web](#)[Home](#)[About](#)[Contact](#)[Search](#)

Search Result.

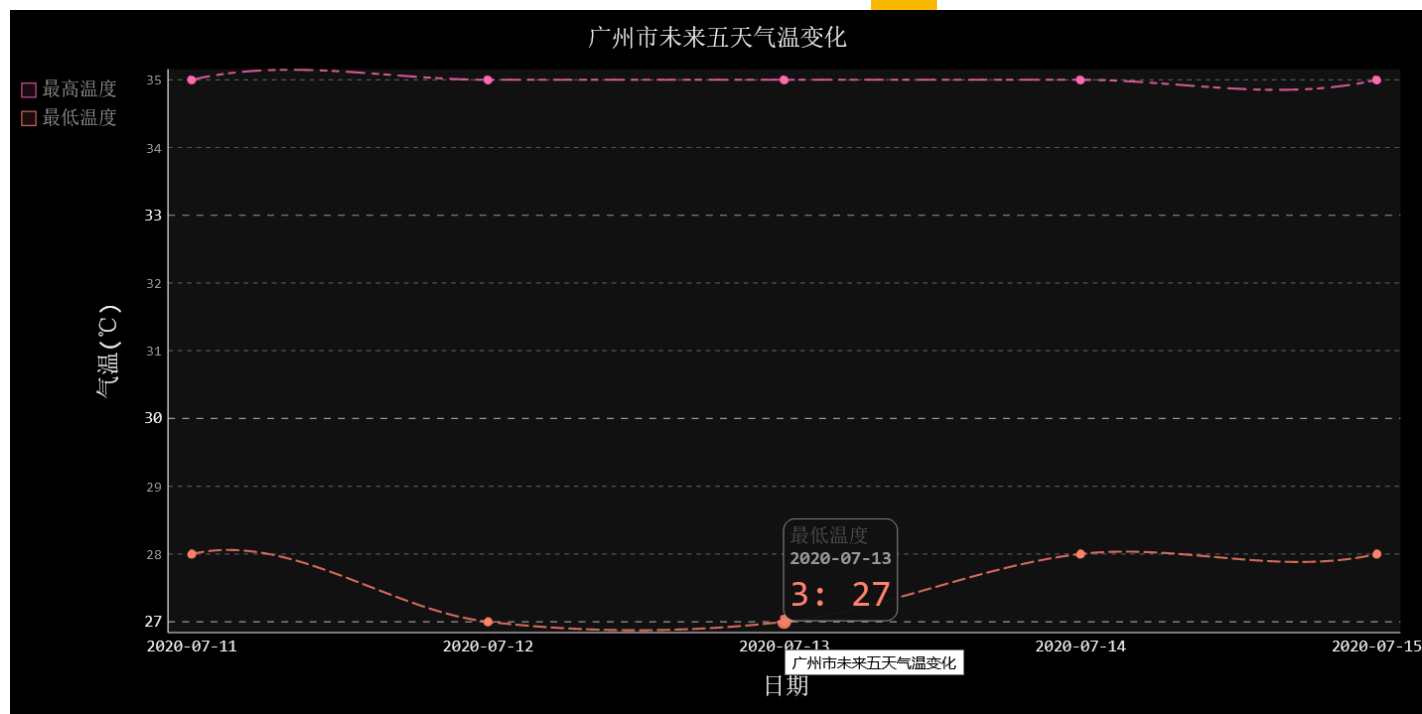
You are searching: 广州市

City Introduction

广州，简称“穗”，别称羊城、花城，是广东省省会、副省级市、国家中心城市、超大城市，国务院批复确定的中国重要的中心城市、国际商贸中心和综合交通枢纽。截至2018年，全市下辖11个区，总面积7434平方千米，建成区面积1249.11平方千米，常住人口1530.59万人，城镇化率86.46%。广州地处中国南部、珠江下游、濒临南海，是中国南部战区司令部驻地，国家物流枢纽，国家综合性门户城市，首批沿海开放城市，是中国通往世界的南大门，粤港澳大湾区、泛珠江三角洲经济区的中心城市以及一带一路的枢纽城市。广州是首批国家历史文化名城，广府文化的发祥地，从秦朝开始一直是郡治、州治、府治的所在地，华南地区的政治、军事、经济、文化和科教中心。从公元三世纪起成为海上丝绸之路的主港，唐宋时成为中国第一大港，是世界著名的东方港市，明清时是中国唯一的对外贸易大港，也是世界唯一两千多年长盛不衰的大港。广州被全球权威机构GaWC评为世界一线城市，每年举办的中国进出口商品交易会吸引了大量客商以及大量外资企业、世界500强企业的投资，国家高新技术企业达8700多家，总量居全国前三，集结了全省80%的高校、70%的科技人员，在校大学生总量居全国第一。广州人均住户存款均居全国前三位，人均可支配收入居全省第一位。广州人类发展指数居中国第一位，国家中心城市指数居中国第三位。福布斯2017年“中国大陆最佳商业城市排行榜”居第二位；中国百强城市排行榜居第三位。

[Learn more about weather »](#)[Learn more about hot destinations »](#)[Learn more about hostels »](#)

天气情况 (未来5天)



Travel Web Home About Contact Search

Search Result.

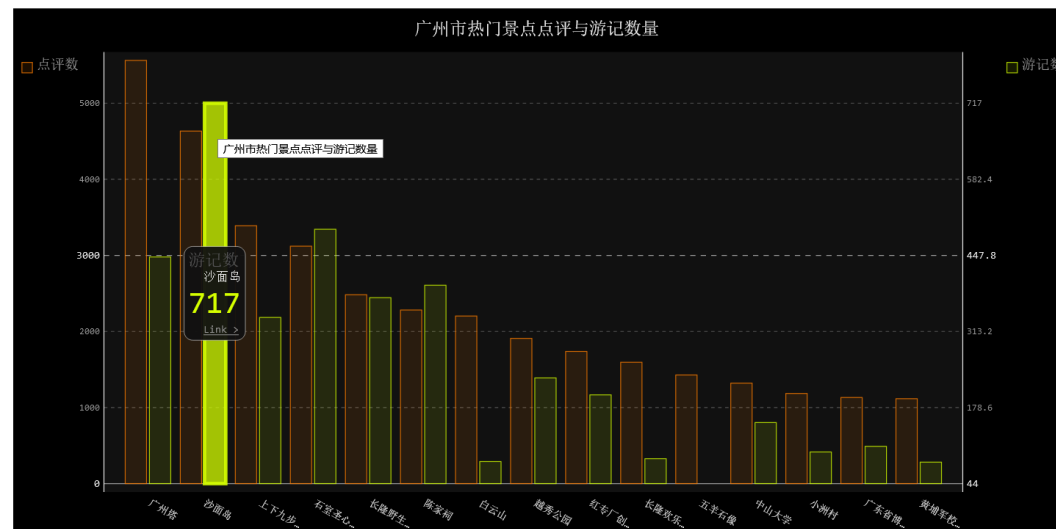
You are searching: 广州市

Weather Forecast

广州市未来五天天气

日期	白天天气	晚间天气
2020-07-11	多云	晴
2020-07-12	晴	晴
2020-07-13	晴	晴
2020-07-14	晴	多云
2020-07-15	多云	多云

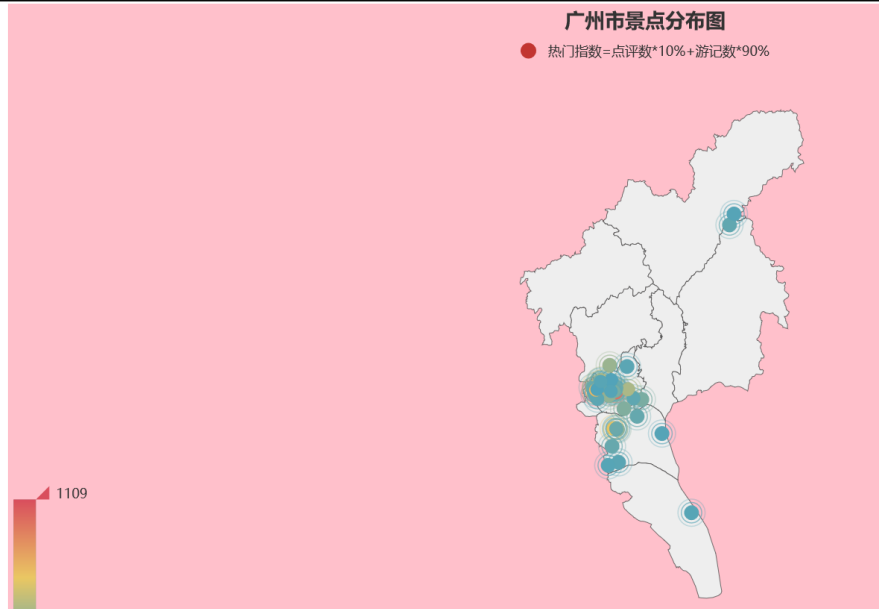
Hot Destinations & Links



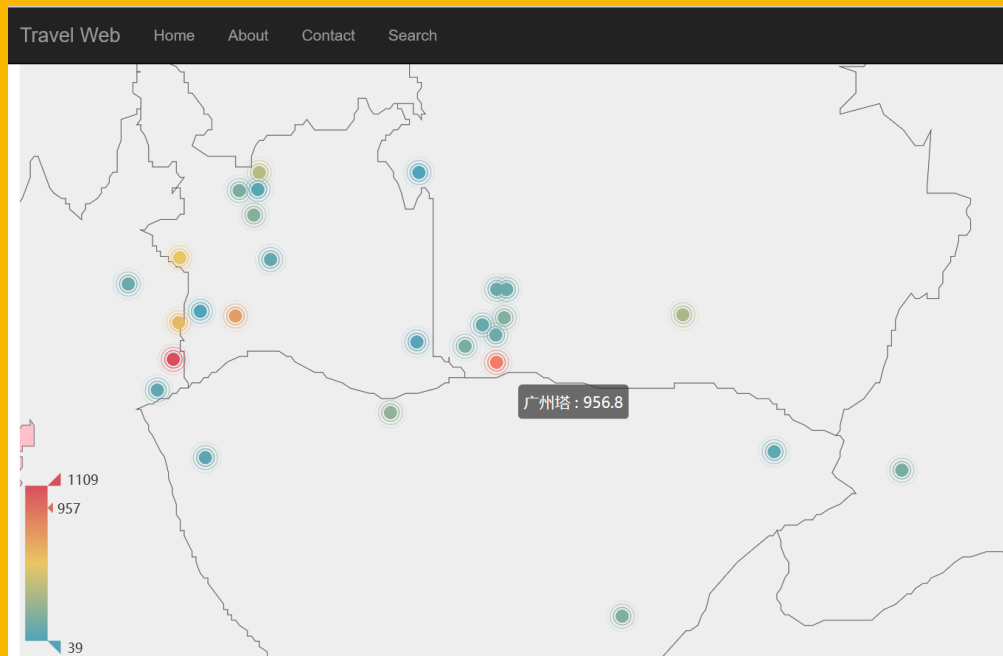
热门景点呈现 (柱状图+地图)

广州市景点分布图

● 热门指数=点评数*10%+游记数*90%



放大后
----->



Select some Destinations (no more than 8) & Learn more about their Best Route

--请选择起点--

广州塔

Best Route

- ☐ 沙面岛
- ☐ 石室圣心大教堂
- ☐ 越秀公园
- ☐ 黄埔军校旧址
- ☐ 中山纪念堂
- ☐ 黄埔古港古村
- ☐ 红专厂创意园
- ☐ 海心沙
- ☐ 南沙天后宫
- ☐ 石门国家森林公园
- ☐ 莲花山旅游区
- ☐ 广州长隆水上乐园
- ☐ 五羊石像
- ☐ 广州博物馆

- ☒ 广州塔
- ☒ 陈家祠
- ☒ 广东省博物馆
- ☒ 广州长隆旅游度假区
- ☒ 西汉南越王博物馆
- ☐ 小洲村
- ☐ 华南植物园
- ☐ 珠江新城
- ☐ 增城白水寨
- ☐ 海心沙广场
- ☐ 珠江长堤
- ☐ 北京路千年古道遗址
- ☐ 上下九广场
- ☐ 二沙岛

- ☐ 长隆野生动物世界
- ☐ 白云山
- ☐ 长隆欢乐世界
- ☐ 中山大学
- ☐ 岭南印象园
- ☐ 花城广场
- ☐ 沙湾古镇
- ☐ 荔湾湖公园
- ☐ 大夫山森林公园
- ☐ 太古仓码头
- ☐ 宝墨园
- ☐ 上下九步行街
- ☐ 广州动物园

最短路径规划



推荐路线

Learn more about Recommended Routes

--请选择天数--

Recommended Routes

Travel Web Home About Contact Search

Search Result.

You are searching: 广州市

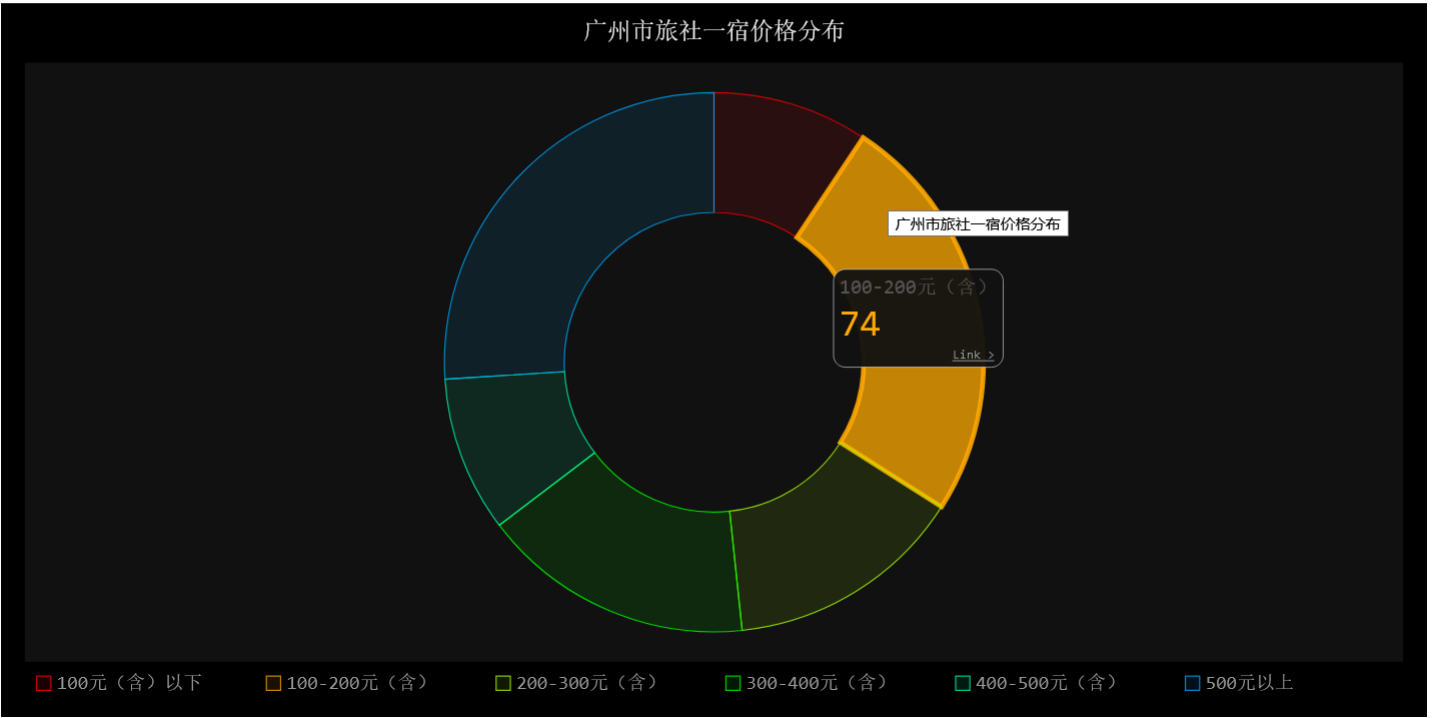
Recommended Routes

广州市推荐旅行路线（来源：途牛网站）

路线景点
长隆欢乐世界, 长隆野生动物世界, 珠海长隆国际海洋度假区
长隆野生动物世界, 长隆国际大马戏
珠海长隆海洋王国, 长隆野生动物世界, 长隆欢乐世界, 长隆旅游度假区, 长隆水上乐园
珠海长隆海洋王国, 长隆欢乐世界, 长隆野生动物世界, 长隆水上乐园
海心沙, 花城广场, 长隆野生动物世界, 长隆欢乐世界, 越秀公园, 长隆水上乐园, 五羊石雕

旅社信息

Hostel Price & Links



Search Result.

You are searching: 广州市

Hostel Introduction (Price: ¥ 100-200)

广州市旅社信息				
民宿名称	基本信息	价格	地址	网址
科学城美式乡村复古双人大床房整租	1室 整租 宜住2人	168	华观路	Learn more »
途驿*独栋、南站接送、地铁直达长隆及市区2房2厅	2室 整租 宜住5人	188	南边天路后街	Learn more »
天河五山华师附近（一只火烈鸟）房 一室	1室 整租 宜住2人	168	华观路中	Learn more »
银都公寓团一大地铁站精装一居室	1室 整租 宜住2人	178	德政南路50号	Learn more »
天河五山华观路万科米酷 小清新一居室	1室 整租 宜住2人	168	华观路中	Learn more »
火车站附近西村地铁站可接送电梯套房	1室 整租 宜住2人	139	同德围鹅掌坦西街9巷	Learn more »
临机场近地铁花园小区豪华大床房	1室 整租 宜住2人	188	平龙路19号	Learn more »
广州南站、临近长隆景区、琶洲会馆大床房	1室 整租 宜住2人	138	南站北路5号	Learn more »

亮点

- 所有模块集合在一个基于**django**库网站上进行展示，操作简单，实用性强。
- 本网站所有信息都是**即时信息**，对于即将出行的用户来说具有宝贵的价值。
- 在旅社和热门景点的网页中都设置了**超链接**，方便用户进入相关网站查询更详细的信息或进行购买决策。
- 注重与用户的**交互性**。用户自行选择查看全国任何城市的相关内容；用户还可以选择想去的景点，我们为其提供最佳路线的设计（尚未找到获得两地点间最短公里数的方法，因此以直线距离代替）；还可以选择查看一定价格区间的旅社信息。
- 注重信息的总结和数据的统计。无论是热门景点、旅社价格还是天气状况，我们都用科学的方式进行统计并采用pygal和pyecharts等库设计好看的**图表**或是**地图**进行展示。

- 理想很美好，现实很残酷，不动手永远都不知道会发生什么
- 主动出击，寻找资源
- 逻辑很重要，细节决定成败



收获

反思/改进

- 两点之间使用的直线距离，会存在和实际相差较大的情况（跨湖跨江？）
- 可以加入login in，为每一个用户记录历史搜索
- 更友好的界面（推荐路线可视化）
- 加入景点开放时间和客流量、车流量的考虑（避免拥挤、塞车等问题）
-

谢谢大家

Python2班 第N小组

