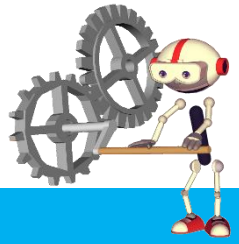


2021年第10届广东省创意机器人大赛培训

编程型机器人
网络与界面编程

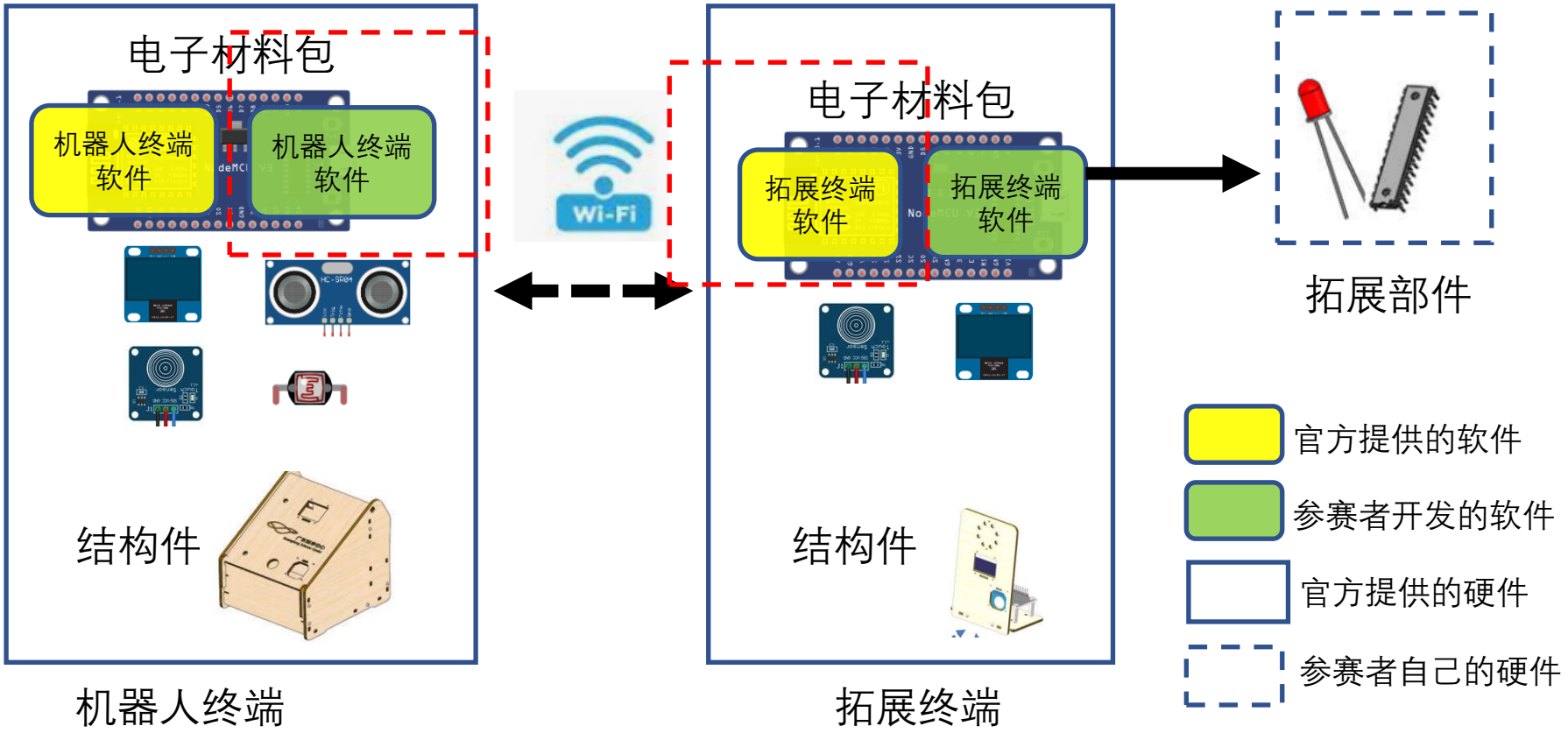
朱金辉

华南理工大学 软件学院

智能软件与机器人研究室

2021年7月

机器人的系统组成


电子材料包
机器人终端软件
机器人终端软件
Wi-Fi
电子材料包
拓展终端软件
拓展终端软件
拓展部件
结构件
结构件
机器人终端
拓展终端
官方提供的软件
参赛者开发的软件
官方提供的硬件
参赛者自己的硬件

提纲

1. WiFi联网
2. 综合案例
3. 网络编程
4. 界面编程

编程型机器人
WiFi联网

基于ESP的WiFi方案

- WiFi热点（Access Point）
- WiFi工作站（Station）

无线路由器
Access Point

ESP控制板
Access Point

ESP控制板
Station

ESP控制板
Station

个人电脑
Station

ESP控制板
Station

个人电脑
Station

WiFi连接程序

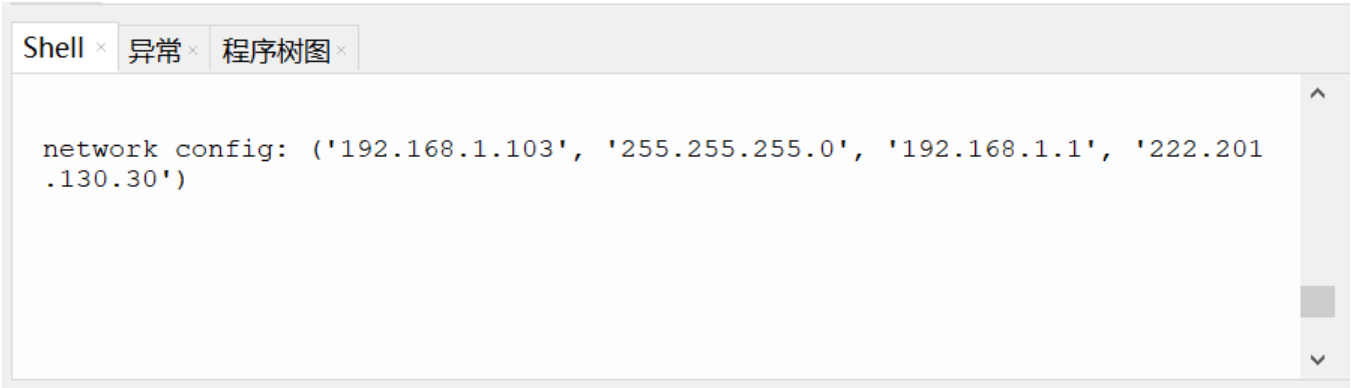
ESP控制板
Station

无线路由器
Access Point

wifi_name : 301

wifi_password : w6jhUJRW

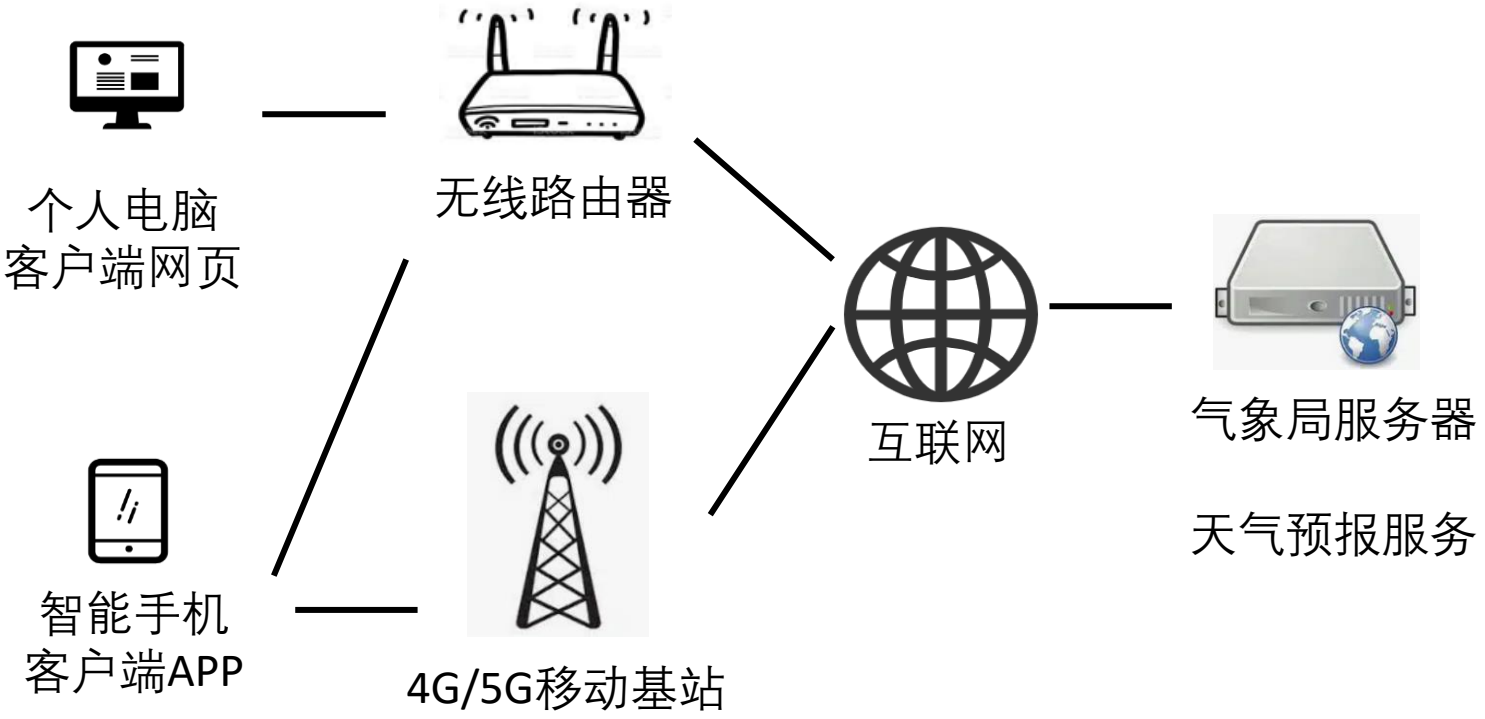
```
1  import network
2
3  sta_if = network.WLAN(network.STA_IF)
4  sta_if.connect("301", "w6jhUJRW")
5  print('network config:', sta_if.ifconfig())
```



编程型机器人
综合案例——天气预报

天气预报系统

- 服务端：气象局服务器
- 客户端：个人电脑网页、手机客户端app

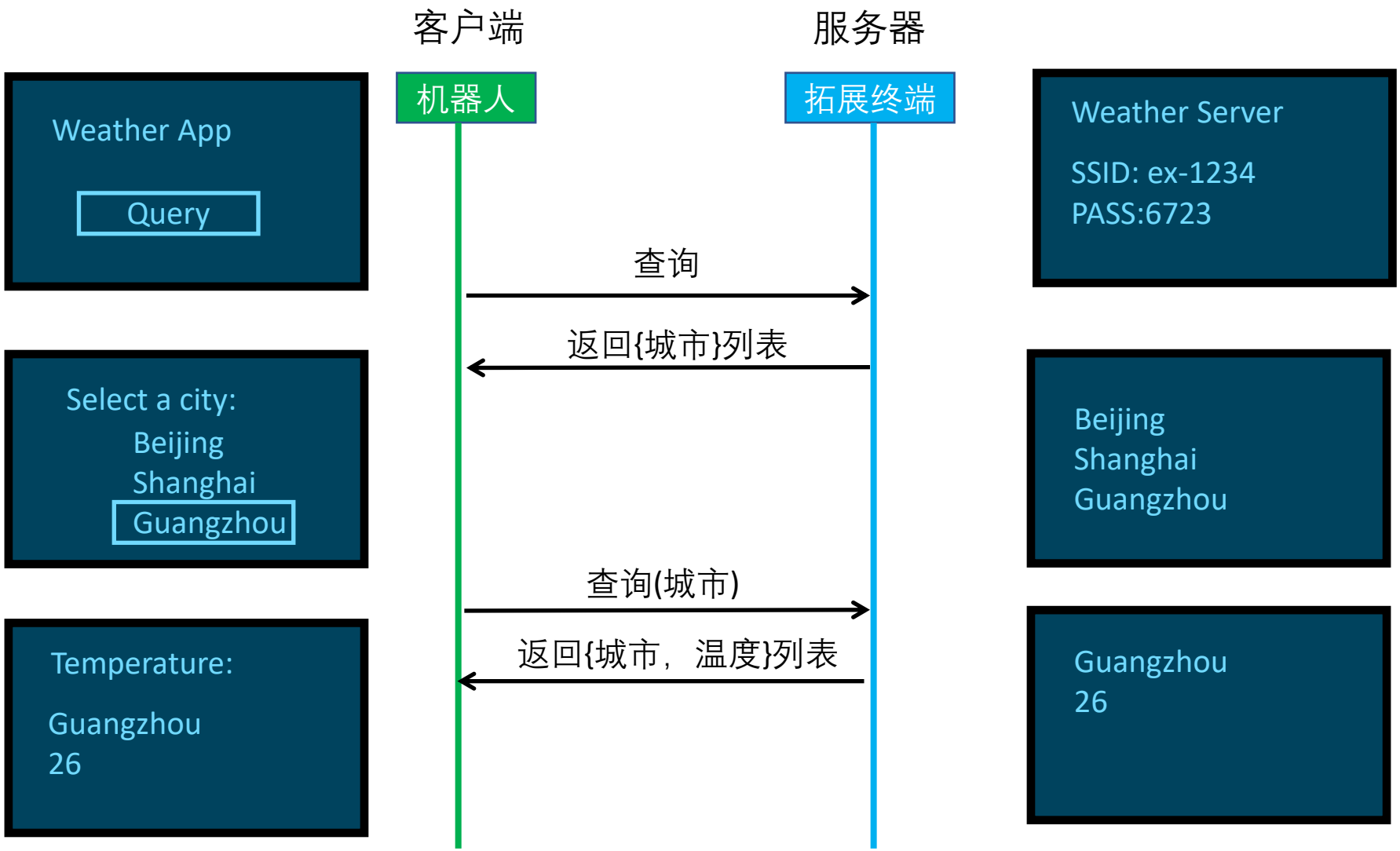


DIY天气预报系统

- HTTP协议是Hyper Text Transfer Protocol（超文本传输协议）的缩写，用于从万维网（WWW: World Wide Web）服务器传输超文本到本地浏览器的传送协议。
- HTTP协议工作于客户端-服务端架构。浏览器作为HTTP客户端通过URL向HTTP服务端即WEB服务器发送所有请求。Web服务器根据接收到的请求后，向客户端发送响应信息。

编程型机器人
网络编程

DIY系统的交互过程


客户端
服务器
机器人
拓展终端
Weather App
Query
Weather Server
SSID: ex-1234
PASS:6723
查询
返回{城市}列表
Select a city:
Beijing
Shanghai
Guangzhou
Beijing
Shanghai
Guangzhou
查询(城市)
返回{城市，温度}列表
Temperature:
Guangzhou
26
Guangzhou
26

交互协议

客户端 服务器

机器人 拓展设备

查询

返回{城市}列表

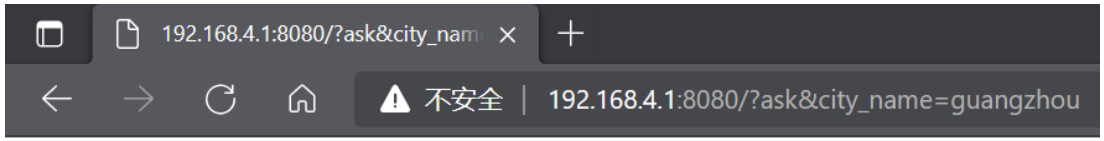
查询(城市)

返回{城市，温度}列表

请求: http://192.168.4.1:8080/?query
响应: {'result':['guangzhou', 'beijing', 'shanghai']}

{'result':['guangzhou', 'beijing', 'shanghai']}

请求: http://192.168.4.1:8080/?ask&city_name=guangzhou
响应: {'result':35}



{'result':35}

编程：Http请求

```
1  import ujson as json
2  import urequests as requests
3
4  r = requests.get("http://192.168.4.1:8080/?query")
5  print("\n r.status_code = "+str(r.status_code))
6  print("r.text = "+ r.text)
7  r.close()
```

URL

Shell × 异常 × 程序树图 ×

```
r.status_code = 200
r.text = {'result':['guangzhou', 'beijing', 'shanghai']}
```

JSON

- JSON(JavaScript Object Notation) 是一种轻量级的数据交换格式。
- 它有简洁和清晰的层次结构，易于人阅读和编写，也易于机器解析和生成。
- JSON是嵌套型键值对的集合，形如 key:value 的多层嵌套组合，如果key或value类型为string字符串，则需用单引号或双引号括起来。

```
{
    "beijing":{
        "temperature":20
    },
    "guangzhou":{
        "temperature":31
    }
}
```

```
1  import ujson as json
2  citylist={
3      "beijing":{
4          "temperature":20
5      },
6      "guangzhou":{
7          "temperature":31
8      }
9  }
10 print(citylist["beijing"]["temperature"])
```

Shell × 异常 × 程序树图 ×

```
20
```

JSON与String

- JSON可以和String（字符串）互相转换
- json.dumps(JSON myJson) 函数可以把json转换成字符串
- json.loads(String myString) 函数可以把string转换为JSON

```
1  import ujson as json
2  citylist_str = "{'guangzhou': {'temperature': 31}}"
3  citylist_json=json.loads(citylist_str)
4  print(citylist_json["guangzhou"]["temperature"])
```

Shell × 异常 × 程序树图 ×

```
Traceback (most recent call last):
  File "main.py", line 3, in <module>
ValueError: syntax error in JSON
MicroPython v1.14 on 2021-02-02; ESP module with ESP8266
Type "help()" for more information.
```

- 程序报错了。因为citylist_str中的各属性是被单引号括起来的，但是json.loads()函数只能把被双引号括起来的string转换成json

数据解析

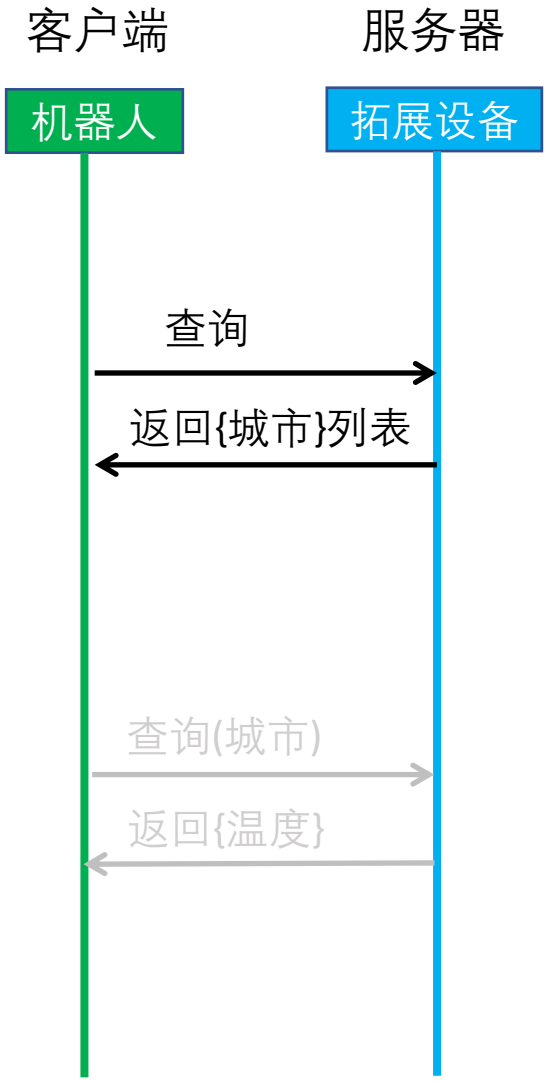
- string.replace(a,b)函数，把string中的所有a替换成b，返回替换后结果

```
1  import ujson as json
2
3  citylist_str = "{'guangzhou': {'temperature': 31}}"
4  print(citylist_str)
5
6  citylist_trans = citylist_str.replace("'", '"')
7  print(citylist_trans)
8
9  citylist_json = json.loads(citylist_trans)
   print(str(citylist_json["guangzhou"]))
```

Shell

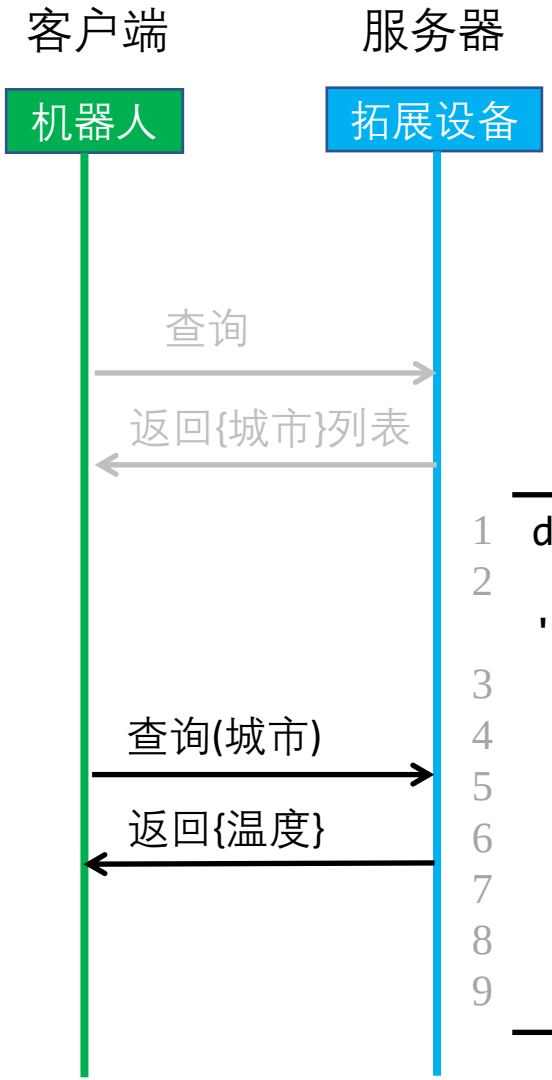
```
>>> %Run -c $EDITOR_CONTENT
{'guangzhou': {'temperature': 31}}
{"guangzhou": {"temperature": 31}}
{'temperature': 31}
```

查询城市列表




```
1  import urequests as requests
2
3  def query_city_list():
4    # 连接WiFi
5    ... 省略
6
7    # 获取城市列表
8    city_list=None
9    URL = 'http://192.168.4.1:8080/?query'
10   r = requests.get(URL)
11   if r.status_code == 200:
12     r_trans = r.text.replace("'", '"')
13     r_json = json.loads(r_trans)
14     city_list = r_json["result"]
15     r.close()
16
17   # 打开新窗口
18   ... 省略
```

查询城市气温



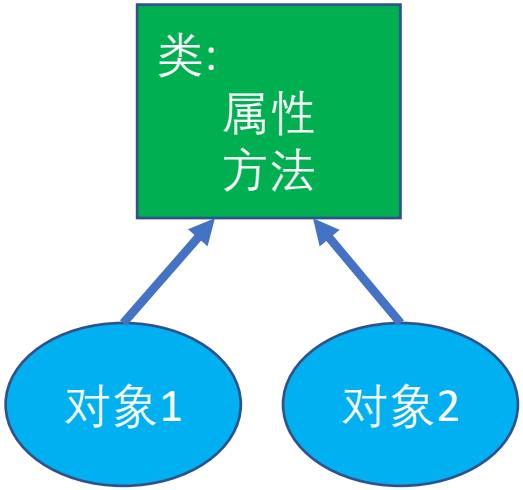

```
1  def query_city_weather(city_name):
2    URL=
   'http://192.168.4.1:8080/?ask&city_name='+ city_name
3    r = requests.get(URL)
4    if r.status_code == 200:
5      r_trans = r.text.replace("'", '"')
6      r_json = json.loads(r_trans)
7      temperature = r_json["result"]
8      r.close()
9      return temperature
```

编程型机器人
界面编程

面向对象

- 类：具有相同的属性和方法的对象的集合。
- 类的数据成员：类中的一个变量（属性）。
- 类的函数成员：类中的一个行为（方法）。
- 对象：用具体的数据成员和成员函数去实例化一个类，就得到一个对象。

```
1  class Person:
2      def __init__(self, name):
3          self.name = name
4
5      def say(self, sentence):
6          print(self.name + " says: " + sentence)
7
8  zhangsan = Person('zhang san')
9  lisi = Person('li si')
10
11 zhangsan.say('hello')
12 lisi.say('ni hao')
```

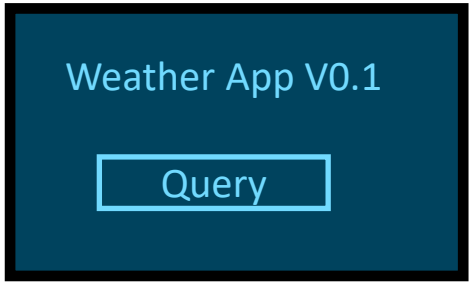


Shell

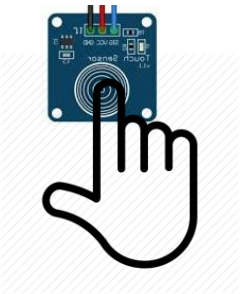
```
>>> %Run -c $EDITOR_CONTENT
zhang san says: hello
li si says: ni hao
```

用户界面编程

- Window类
  - 窗口可包含多个组件（例如button）
  - 统一接收触摸开关的输入事件
  - 触发事件对应的回调函数
- Button类

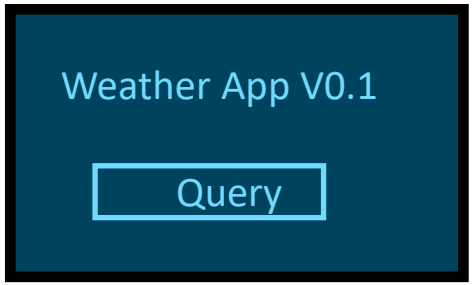



```
1  from window import Window
2  from questionWindow import Button
```



MainWindow

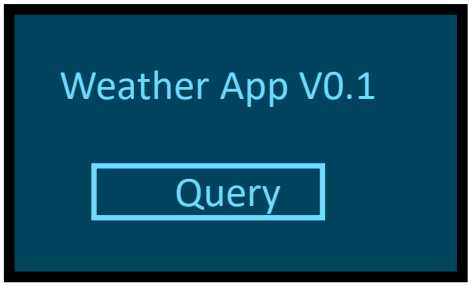
- MainWindow继承Window
- MainWindow初始化过程包括:
  - 设置窗口标题title
  - 新建和添加button
  - 指定button的回调函数
  - 指定显示回调函数




```
1  # main.py
2  from window import Window
3  from questionWindow import Button
4  from util import display
5  class MainWindow(Window):
6    def __init__(self,title="Weather App V0.1"):
7      Window.__init__(self)
8      self.title = title
9      self.btn_query = Button("Query",50,40,40,12)
10     self.btn_weather.cb_hold = self.query_city_list
11     Window.add(self,self.btn_weather)
12     Window.setDrawCB(self,self.cb_draw)
13   def cb_draw(self):
14     display.text(self.title,2,2,1)
```

查询城市

```
def query_city_list(self):
    # 第一步，连接wifi
    sta_if = network.WLAN(network.STA_IF)
    sta_if.active(True)
    sta_if.connect("JC-0877ab","12345678")
    print('network config:', sta_if.ifconfig())
    # 第二步，通过http请求获取城市列表
    city_list=None
    r = requests.get(
            “http://192.168.4.1:8080/?query”)
    r_trans = r.text.replace("'", '"')
    r_json = json.loads(r_trans)
    city_list = r_json["result"]
    r.close()
    # 第三步，新建CityWindow
    from cityWindow import CityWindow
    w = CityWindow(city_list)
    w.run()
    # 第四步（可选），回收资源，避免内存泄露del(w)
    # del(CityWindow)
    # del(sys.modules['cityWindow'])
```



CityWindow

```
1  # cityWindow.py
2
3  from window import Window
4  from questionWindow import Button
5  from util import display
6
7  class CityWindow(Window):
8    def __init__(self, city_list):
9      Window.__init__(self)
10     self.city_list = city_list
11     Window.setDrawCB(self,self.cb_draw)
12     # 创建button，每个button的回调函数相同，但button的title
13     for i in range(len(self.city_list)):
14         btn_city = Button(self.city_list[i], 40, 30 + i * 10, 80, 12)
15         btn_city.cb_hold = self.cb_query_weather
16         Window.add(self, btn_city)
17   def cb_draw(self):
18     display.text("City:", 2, 2, 1)
```

查询气温

```
1  def cb_query_weather(self):
2    city_name = self.hold_title
3    print(“city name is ” + str(city_name))
4    temperature = self.query_city_weather(city_name)
5    display.fill(0)
6    display.text("Temperature:", 10, 10)
7    display.text(city_name + ":" +
                      str(temperature), 20, 30)
8    display.show()
9    time.sleep(3)
10   Window.close(self)
11
12 def query_city_weather(self, city_name):
13   URL = 'http://192.168.4.1:8080/?ask&city_name=' + city_name
14   r = requests.get(URL)
15   r_trans = r.text.replace("'", '"')
16   r_json = json.loads(r_trans)
17   temperature = r_json["result"]
18   r.close()
19   return temperature
```

谢谢！