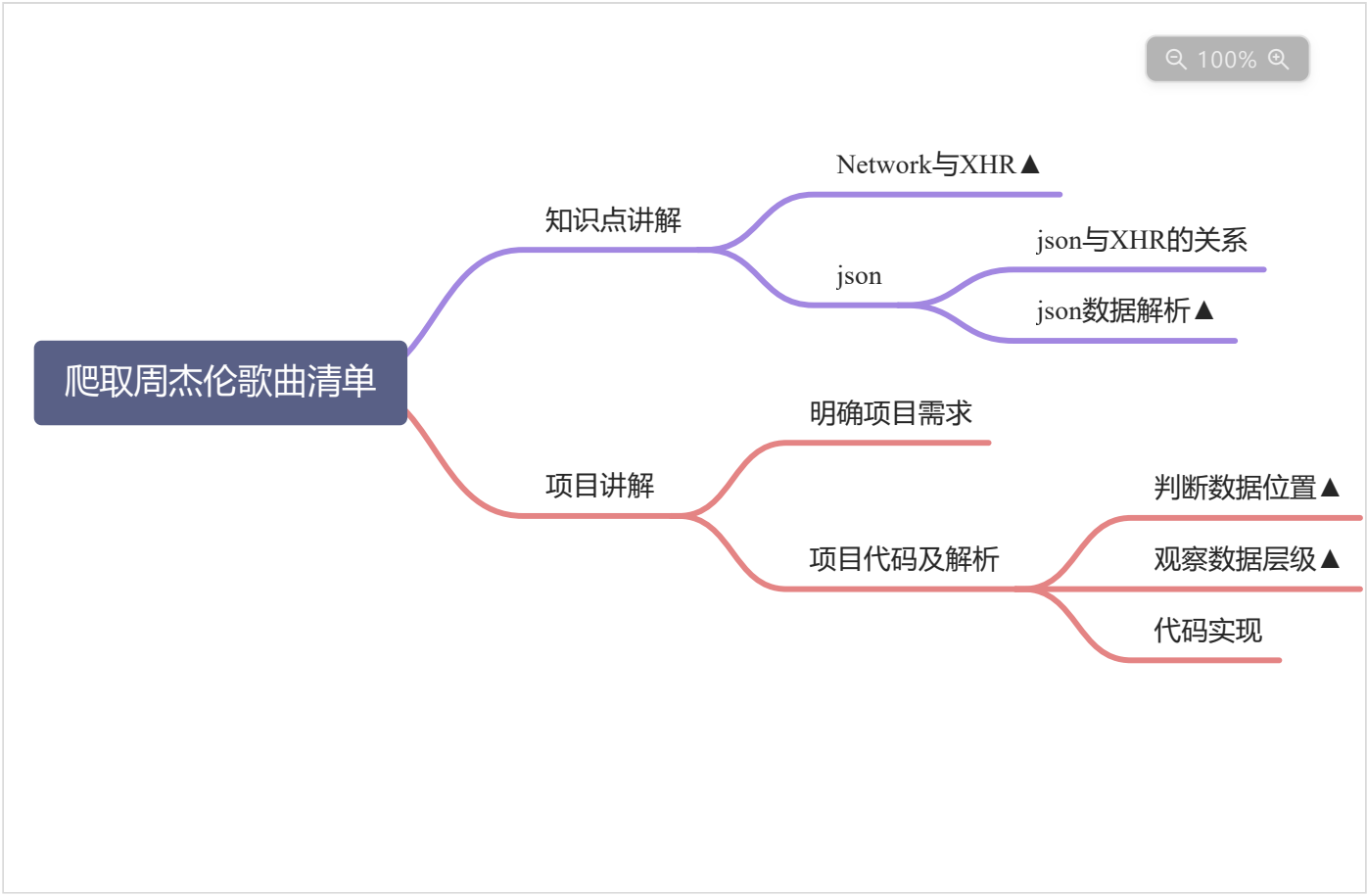
第4课 json

一、课程结构导图



注：▲为重点知识点。

二、知识点讲解

2.1 Network与XHR

概念：Network能够记录浏览器的所有请求资源信息（包括状态、资源类型、大小、所用时间等），我们最常用的是：ALL和XHR，XHR。XHR是一种数据传输对象，功能是在服务器和浏览器之间传输数据。

数据分类：

ALL	查看全部
XHR	一种传输对象
Doc	一般是第0个请求在这里
Img	仅查看图片

Media	仅查看媒体文件
Other	其他
JS和CSS	前端代码
Font	字体
WS和Manifest	网络编程相关知识

2.2 json

2.2.1 json与XHR的关系

概念：

1. json是一种规范数据传输的格式数据的格式，格式形式有点像字典和列表的结合体，我们在XHR里查看到的列表/字典，严格来说其实它不是列表/字典，它是json。
2. json和XHR之间的关系是，XHR用于传输数据，其中大部分被传输的数据，都是json数据。
3. **在Python语言当中，json是一种特殊的字符串**，这种字符串特殊在于写法，它是用列表/字典的语法写成的。

示例：

```
a = '1,2,3,4' # 这是字符串
b = [1,2,3,4]  # 这是列表
c = '[1,2,3,4]'
# 这是字符串，但它是用json格式写的字符串

d = {'name':'小明','age':18,'height':180}# 这是字典
e = '{''name'':''小明'',''age'':18,''height'':180}'
# 这是字符串，但它是用json格式写的字符串
```

2.2.2 json数据解析

方法1：导入requests库，使用Response类的json()函数解析json数据。

示例1：

```python
# 使用requests库解析json数据

import requests

res_music = requests.get('https://c.y.qq.com/soso/fcgi-bin/client_search_cp?ct=24&qqmusic_ve

json_music = res_music.json()  # 使用json()方法，将response对象，转为列表/字典

print(type(json_music))  # 打印json_music的数据类型

# 打印结果为：<class 'dict'>
```

代码解析：

1. 第7行代码，requests库中有解析json数据的方法，也就是json()函数，能将json数据解析为字典/列表。
2. 解析出来的json_music变量，是一个字典，我们可以用for循环进行遍历，将数据进行逐层提取。

方法2：导入json模块，使用json.loads()函数解析json数据。

示例2：

```python
# 使用json模块
import requests,json

res_music = requests.get('https://c.y.qq.com/soso/fcgi-bin/client_search_cp?ct=24&qqmusic_ve

res_text = res_music.text  # 将Response对象,转为字符串数据

json_music = json.loads(res_text) # 使用json.loads()方法，将字符串数据，转为列表/字典

print(type(json_music))
# 打印结果为：<class 'dict'>
```

代码解析：

1. 第1行代码，json模块属于python内置模块，直接调用即可，无须另外安装和下载。
2. 第8行代码，json.loads()函数是将字符串类型，转为字典/列表，所以第6行代码，需要先转为字符串，而变量res_text，则是请求之后获取的json数据。

两种方法小结：

1. 相同点：两种方法都可以将json数据转换为字典/列表数据。

2. 不同点：
    a. Response.json()使用的requests模块，而json.loads()使用的是json模块。
    b. Response.json()是直接将json数据转为字典/列表，而json,loads()是将json数据先转为字符串，再转为字典/列表（如下图所示）。

三、项目讲解

3.1 明确项目需求

需求：

1. 爬取QQ音乐中周杰轮歌单信息，包括歌曲名，专辑名，播放时长，播放链接。

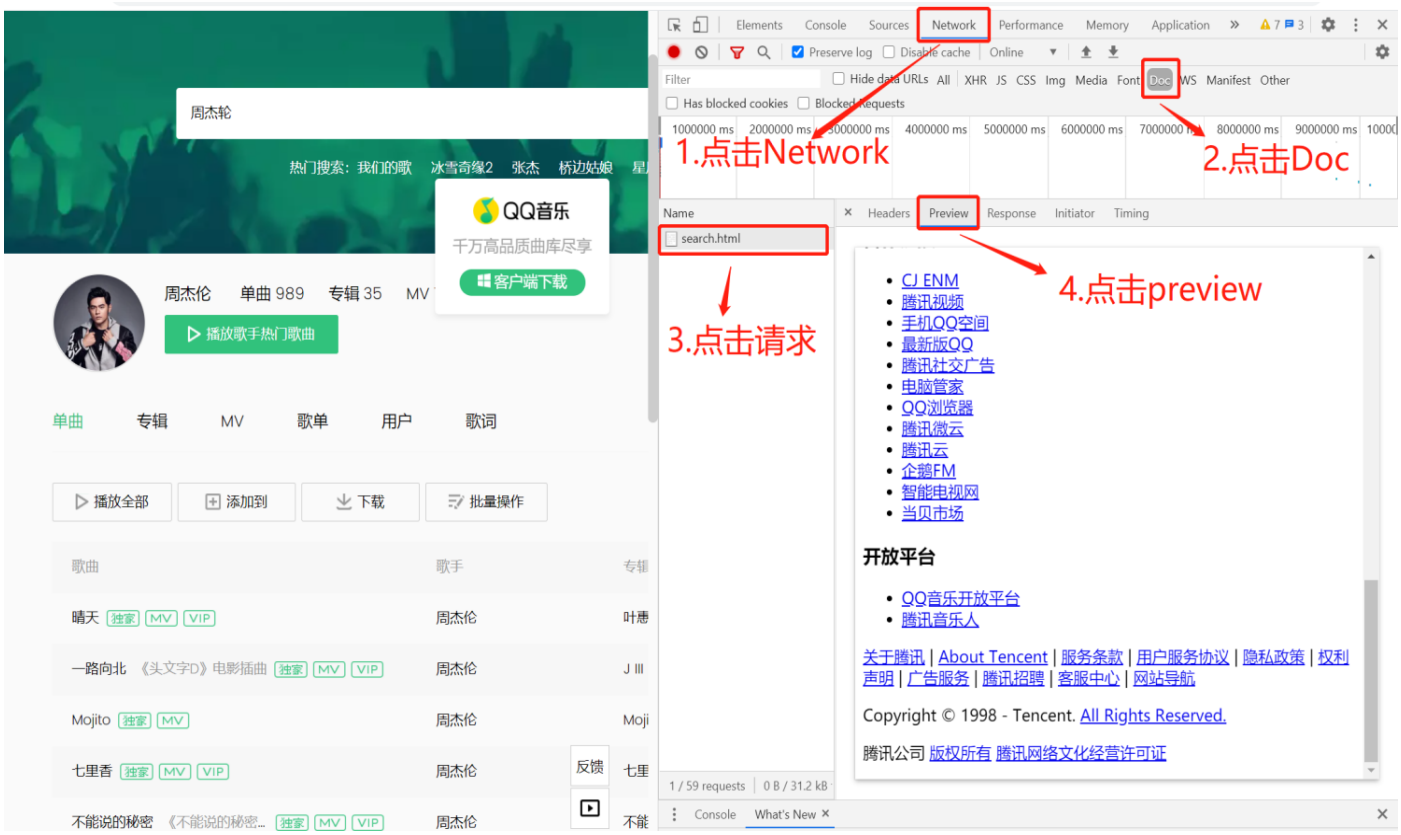
3.2 项目代码及解析

3.2.1 判断数据位置

目的1：判断数据是否在html中。

关键操作1：在搜索框输入周杰伦，打开检查中的Network-点击Doc-点击search.html请求-点击preview。

图示1：



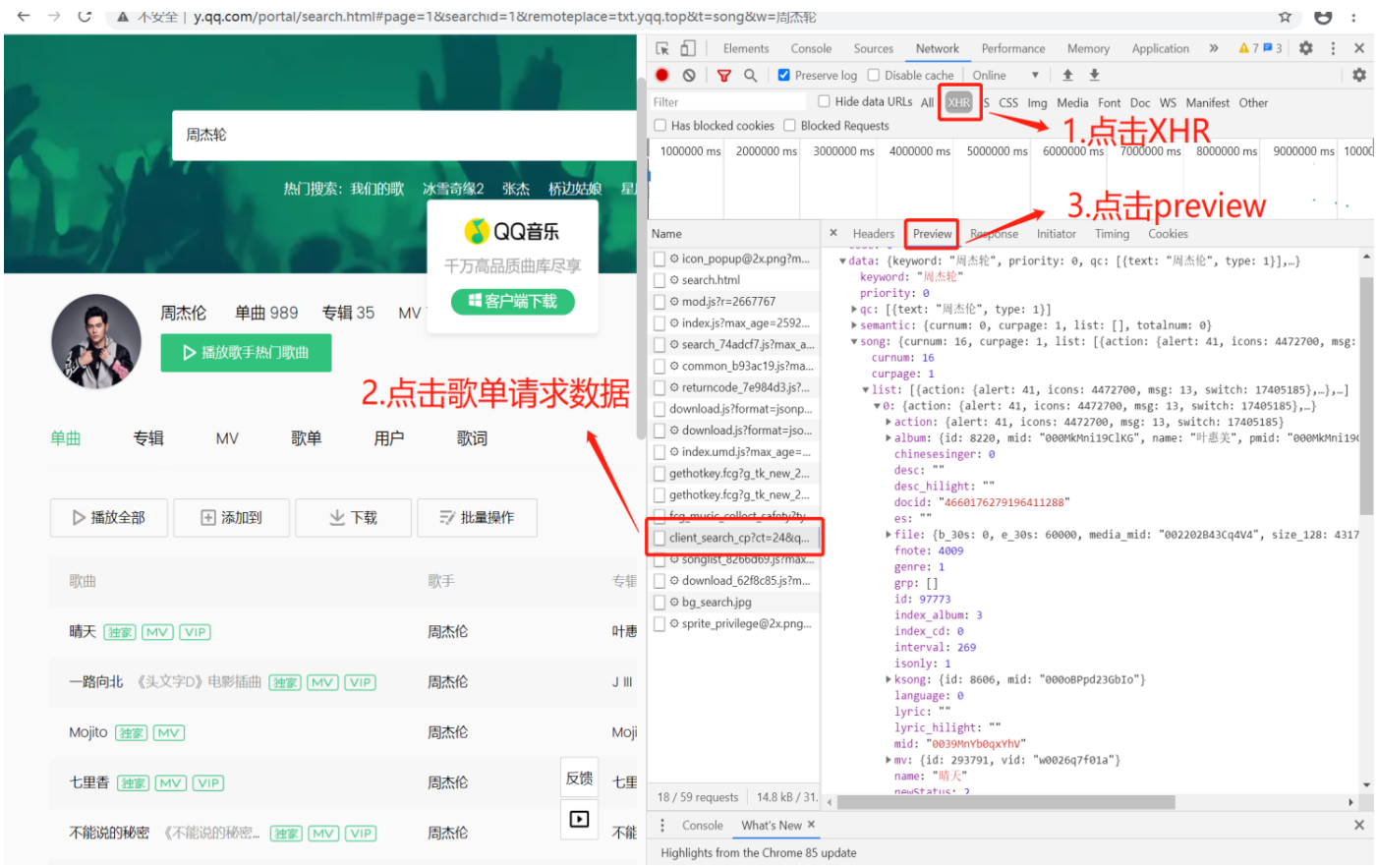
操作解析：

1. 我们在Elments看到的其实是html（网页框架）和Network（请求数据）加载完的结果显示，简单的来说，我们平时使用requests.get(url)发送请求时，请求的一般是第0个请求的数据，并没有请求到其它的请求，因此获取的也只是第0个请求的数据。
2. 我们在爬取网站时，先查看要爬取的数据是否存放在第0个请求中（点击preview可以查看），若在第0个请求中，则可以直接用BeautifulSoup解析数据和提取数据。若不在，则需要在其他请求中寻找数据。
3. 通过观察，歌单的信息并不在第0个请求中。

目的2：判断数据是否在xhr中。

关键操作2：点击xhr-刷新网页-找到并点击请求名为client_search_cp？-点击perview。

图示2：



操作解析：

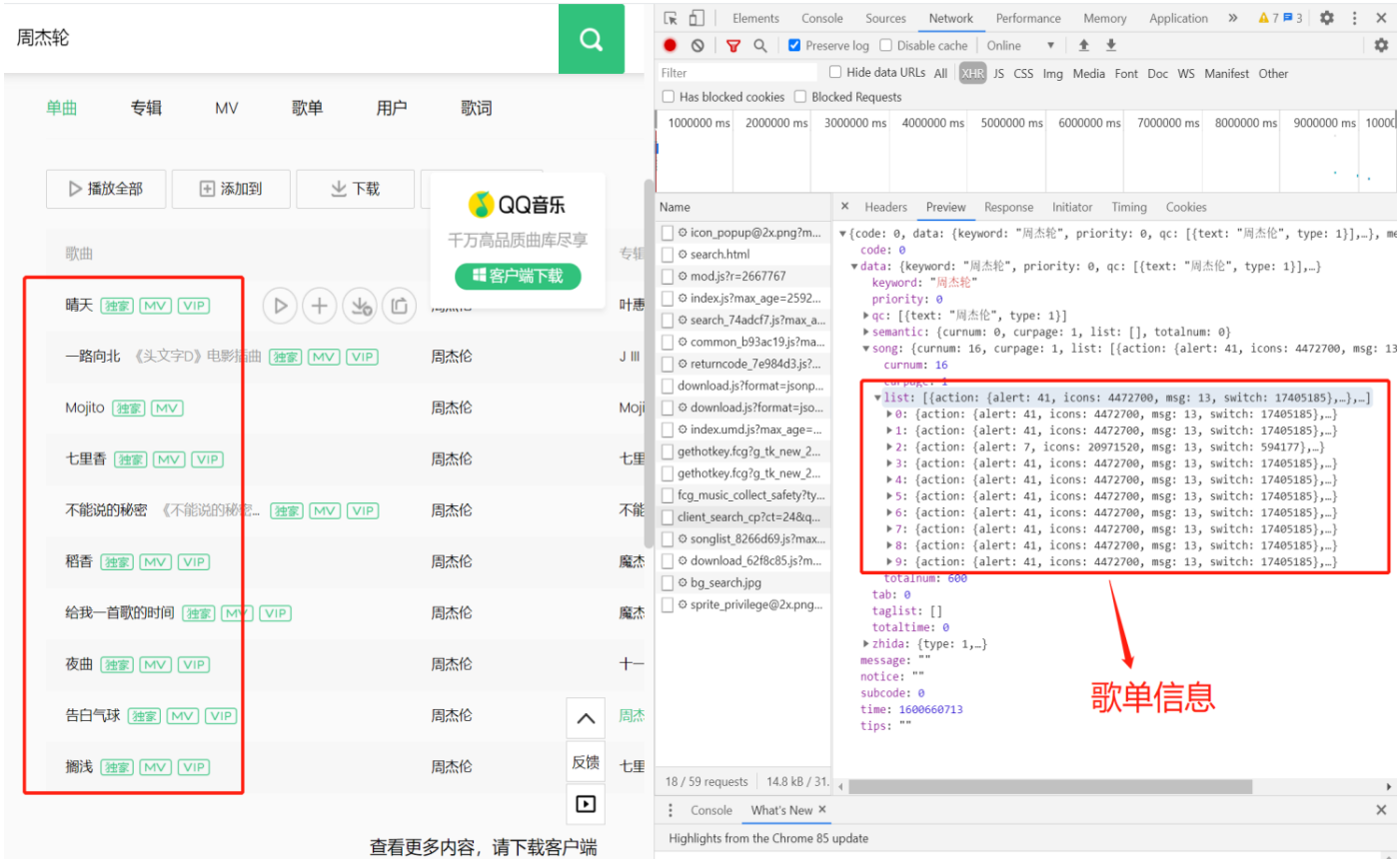
1. 之所以查看client_search_cp这个请求，是因为它的请求加载时间是最长的，而且文件大小也是最大的，歌单的信息比较多，可以推测出歌单的信息存放在该请求中。
2. 打开preview后，会发现并不是我们熟悉的html，而是一个字典与列表相互嵌套的格式，其实它就是json。
3. 通过观察，发现了歌单中的歌曲名，也就是name，可以判断出该请求存放着歌单的所有信息。

3.2.2 观察数据层级

目的1：查找出歌单信息。

关键操作1：按data-song-list顺序，依次展开数据层级。

图示1：



操作解析：

1. 图示1左边显示的是10首歌曲，和右边的list中的元素一一对应，图示list中的0，1，2，3，4.....9，代表的其实是索引值，为了是方便观察数据。
2. 由于list中，每一个元素代表一首歌曲的信息，那么我们只需要展开观察其中一个元素中内容即可。
3. data-song-list是一个字典嵌套的层级关系，如下图所示。

第一层：**{data:{ }, message:'' '', notice:'' '', ...}**

嵌套

第二层：**{song:{ }, qc:[ ], semantic:{ }, ...}**

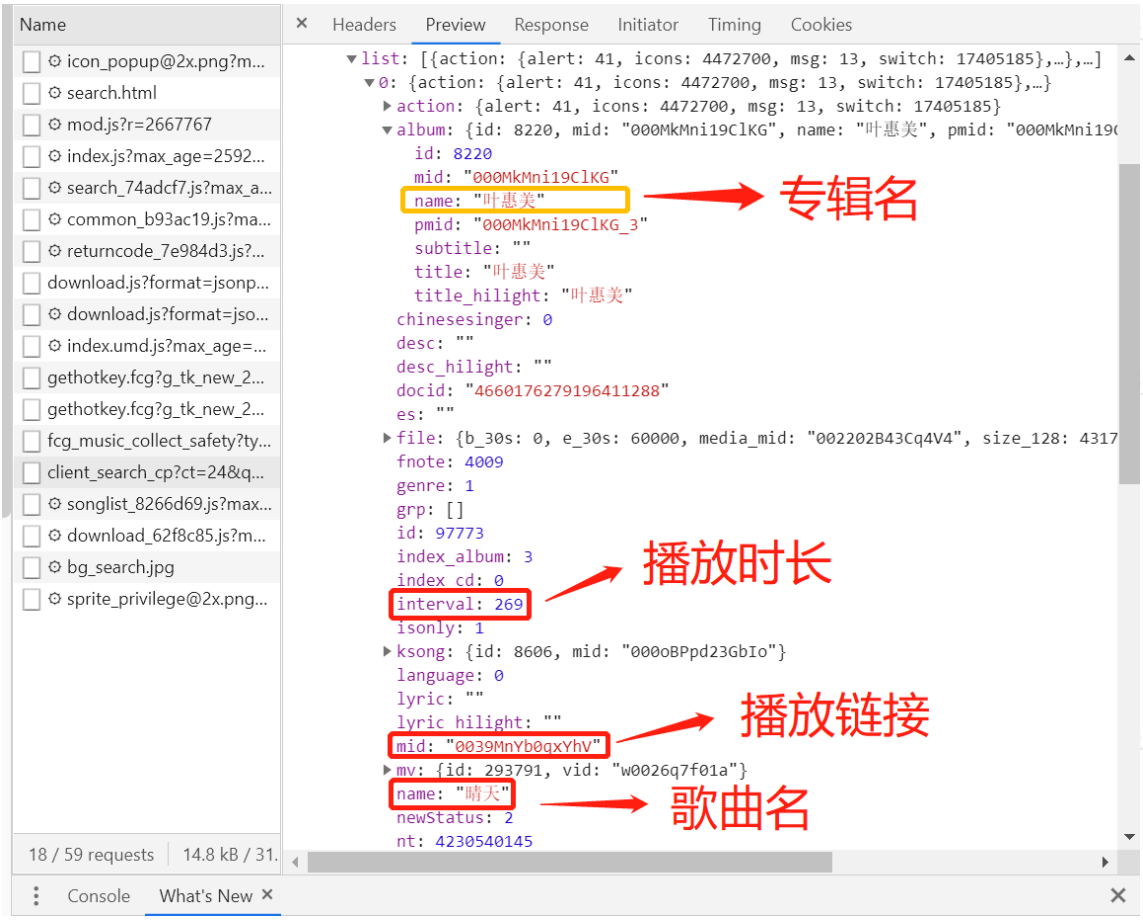
嵌套

第三层：**list:[list_music]**

目的2：查找出歌曲名，专辑名，播放时长，播放链接。

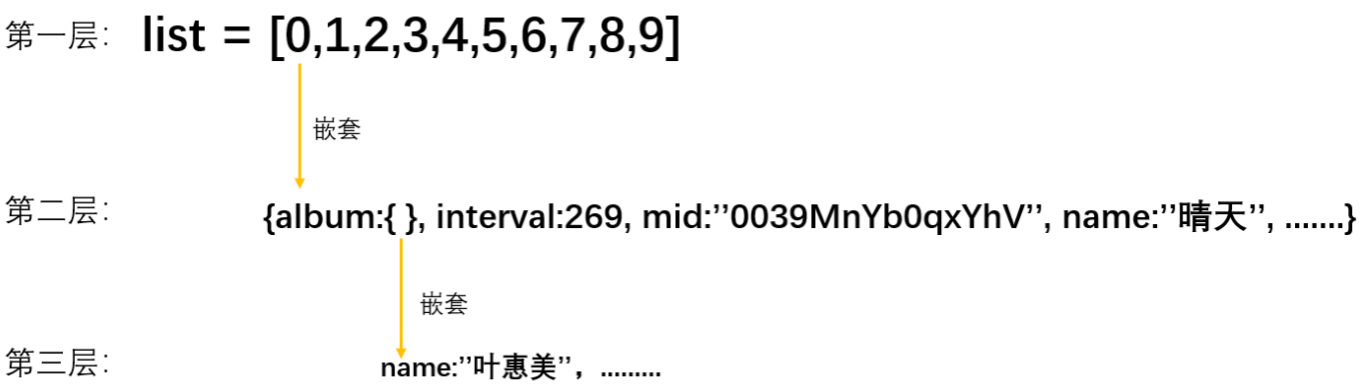
关键操作2：观察数据之间的层级嵌套关系。

图示2：



操作解析：

1. 图示2中的播放链接只属于播放链接组成的一部分，完整的播放链接需要进行字符串的拼接，例如：https://y.qq.com/n/yqq/song/ <https://y.qq.com/n/yqq/song/> +0039MnYb0qxYhV+html。
2. 通过观察，歌曲名，播放链接，播放时长属于同一层级，三者与album属于同一层级，都属于list列表中的第0个元素（如下图）。



3.2.3 代码实现

关键代码：

```
1  import requests
2  res_music = requests.get('https://c.y.qq.com/soso/fcgi-bin/client_search_cp?ct=24&qqmusic_ve
3
4  json_music = res_music.json() # 使用json()方法，将response对象，转为列表/字典
5
6  list_music = json_music['data']['song']['list'] # 一层一层地取字典，获取歌单列表
7
8  for music in list_music:
9  # list_music是一个列表，music是它里面的元素
10     print(music['name'])
11     # 以name为键，查找歌曲名
12     print('所属专辑：'+music['album']['name'])
13     # 查找专辑名
14     print('播放时长：'+str(music['interval'])+'秒')
15     # 查找播放时长
16     print('播放链接：https://y.qq.com/n/yqq/song/'+music['mid']+'.html\n\n')
17     # 查找播放链接
```

代码解析：

1. 第8行代码，由于提取出来的list_music是列表，所以需要进行for循环遍历，依次提取出列表里面的数据，并根据3.2.2中的图示2层级关系，使用字典取值方式，将歌曲名，专辑名，播放时长，播放链接逐一取出。
2. 当我们逐层提取数据时，我们需要看清楚提取之后的数据是字典还是列表。