爬虫的“盗亦有道”：

robot协议：存放于网站根目录下

搜索引擎关键词接口：

百度：<http://www.baidu.com/s?wd=>keyword

360: [http://www.so.com/s?q= keyword](http://www.so.com/s?q=keyword)

网站人机交互窗口的实质是连接，可以通过网页解析找到链接并用程序模拟

HTTP协议：Hypertext Transfer Protocol，超文本传输协议

基于“请求与响应”模式，无状态的应用层级协议

URL格式：<http://host[:port][path]>

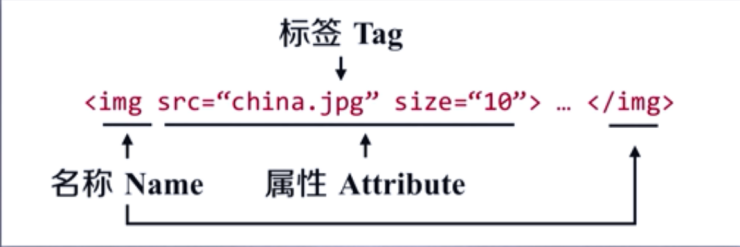
host:合法的internet主机域名或者IP地址

port：端口号，缺省端口为80

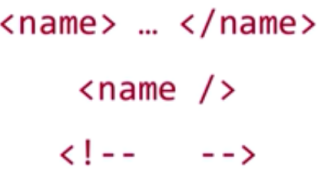
path：请求资源的路径

HTTP方法

：

信息组织与标记：

XML格式：类似于HTML格式



空元素缩写

注释

JSON格式：带有格式的键值对

“key”:”value”

“key”:[”value1”,”value2”,1234567890]

“key”:{

”value1”

“key”:”value”

}

YAML格式：无格式键值对（仅字符串）

所属关系： 缩进

key:value

key:value

并列关系：-开头

name：

-value1

-value2

|表达整体，#表达注释

text：|整块数据 #注释

Requests 库使用：

requests方法：



（基于request方法封装）

r=requests.get(url,params=None,\*\*kwargs)

url:资源地址

params ：url额外参数，字典或字节流格式，可选

\*\*kwargs: 12个控制访问的额外参数

r为response对象，其包含属性有：

r.status\_code:http请求的返回状态，200成功，404失败

r.text:http 相应内容的字符串形式

r.encoding:从http header中猜测响应内容的编码形式

r.apparent\_encoding:从容分析响应内容的编码方式（备选编码方式）

（但更准确）

r.content:http响应内容的二进制形式

requests库的异常：

requests.ConncetionError:网络连接错误异常，如DNS查询失败、拒绝连接

requests.HTTPError:HTTP错误异常

requests.URLRequired:URL缺失异常

requests.TooManyRedirects:超过最大重定向次数，产生重定向异常

（URL重定向次数超过requests库限制，常出现在复杂链接访问）

requests.ConnectionTimeout:连接远程服务器超时异常

requests.Timeout:请求URL超时

异常方法：

r.raise\_for\_status():若不是两百，返回异常 requests.HTTPError

requests库最主要方法：request

requests.request(‘METHOD’,’url’,\*\*kwargs)

method：即为七种主要方法

\*\*kwargs：控制访问参数

params:字典或字节序列，作为参数增加到url中

data:字典字节序列或文件对象，作为request的内容

json：JSON格式的数据，作为request的内容

headers：字典，HTTP定制头

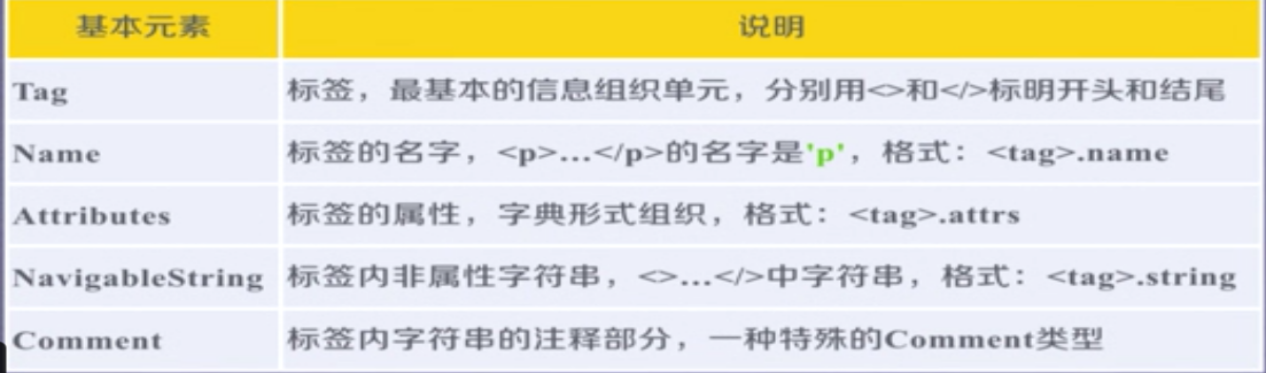
verify、cert、cookies、auth、files、timeout、proxies、allow\_redirect、stream

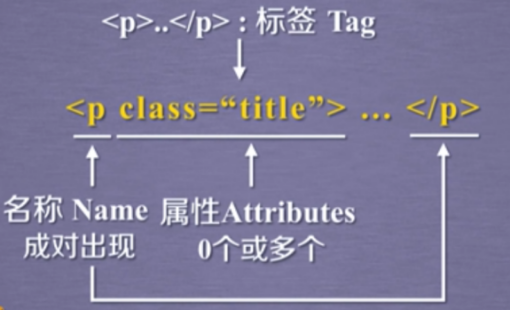
Beautiful Soup库使用

import bs4

可解析树形结构的网页源代码

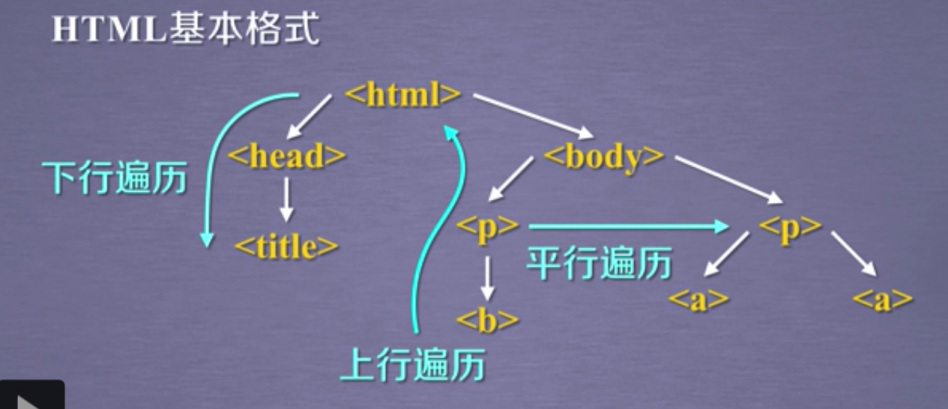
Beautiful Soup库的基本元素：





eg. p标签：

HTML基本格式以及遍历：



 下行遍历：

 上行遍历：

平行遍历：平行遍历必须在同一父亲节点下



<>find\_all(name,attrs,recursive,string,\*\*kwargs)，返回一个列表

name:对标签名称的检索字符串

（查找多个标签时用列表传入）

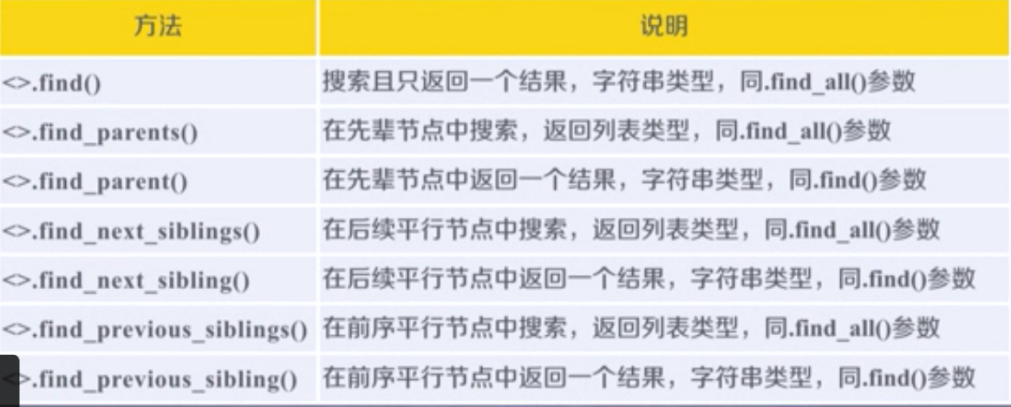
attrs：对标签属性值的检索字符串，可标注属性检索

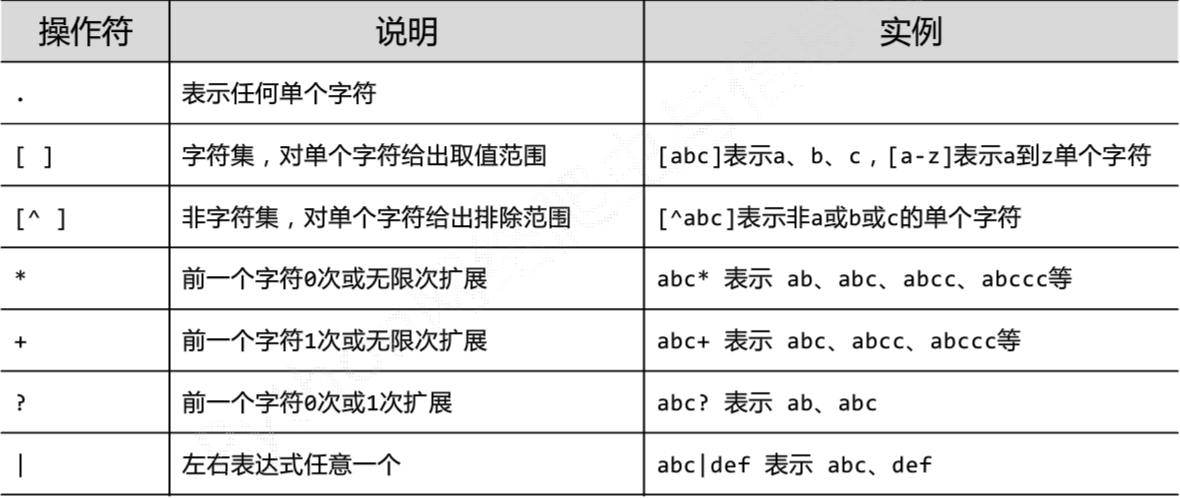
recursive：布尔型，默认True，是否对子孙全部搜索

string：对<>....</>中的内容进行检索

（以上参数直接赋值时必须完整赋值，若需模糊搜索需要正则表达式）

延伸方法：

 (参数类型与原型相同)

正则表达式：

常用操作符：



经典正则表达式实例：

^[A-Za-z]+$ 26个字母组成的字符串

^[A-Za-z0-9]+$ 26个字母和数字组成的字符串

^-?\d+$ 整数组成的字符串

^[0-9]\*[1-9][0-9]\*$ 正整数形式的字符串

[1-9]\d{5} 中国境内邮政编码

[\u4e00-\u9fa5] 中文字符串

\d{3}-\d{4}|\d{8}-d{7} 国内电话号码

IP地址的正则表达式匹配：

[1-9]?\d:0-99 1\d{2}:100-199

2[0-4]\d:200-249 25[0-5]:250-255

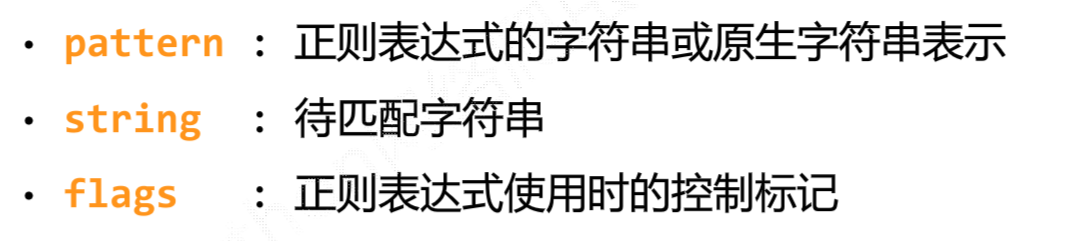
(([1-9]?\d|1\d{2}|2[0-4]\d|25[0-5]).){3}([1-9]?\d|1\d{2}|2[0-4]\d|25[0-5])

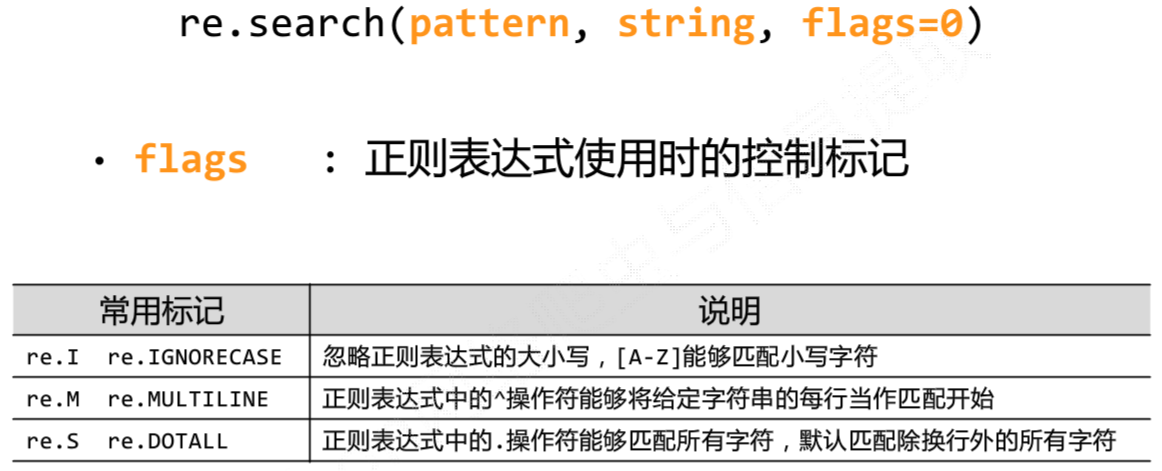
re库的使用：

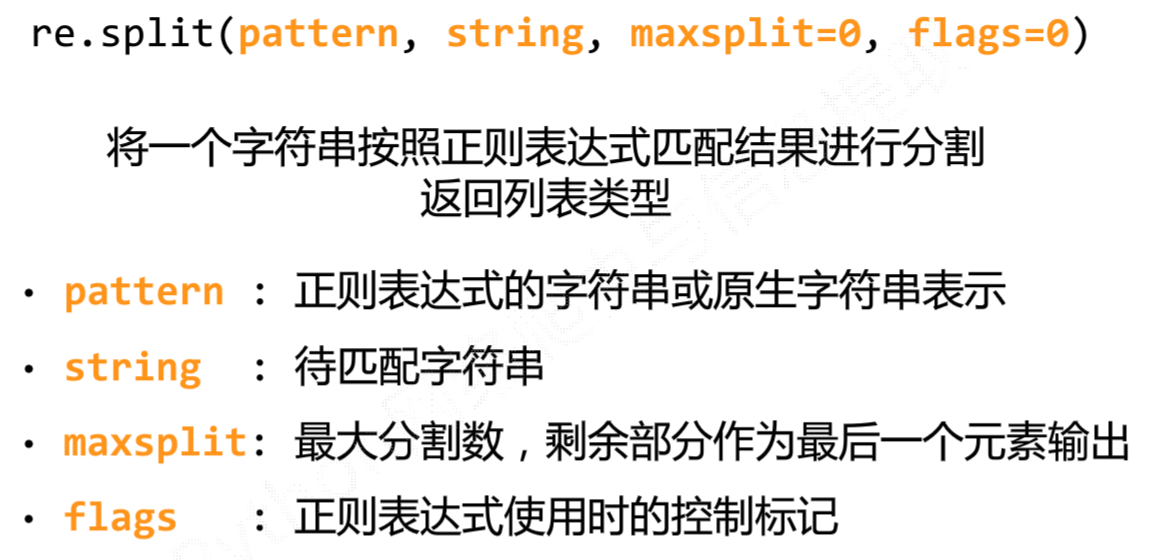
re库采用原生字符串表示正则表达式，表示为r’text’

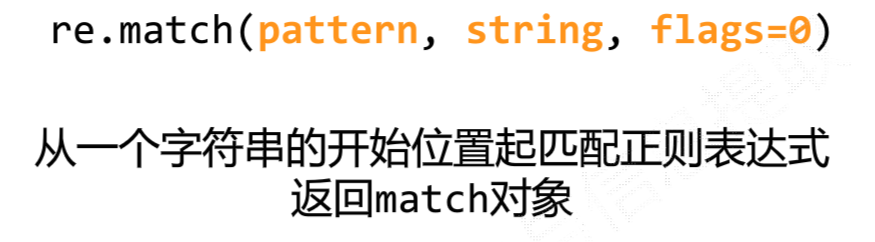
（原生字符串即不带转义字符的字符串）

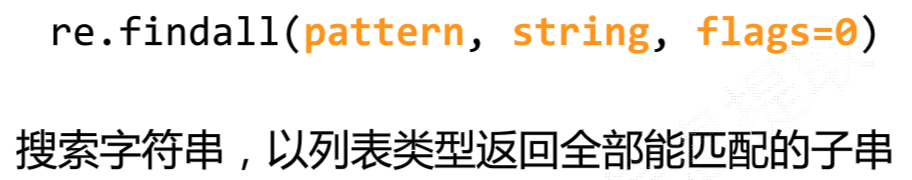
re库常用方法：

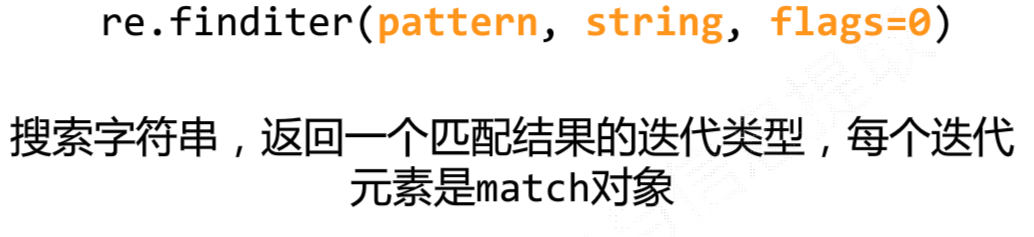


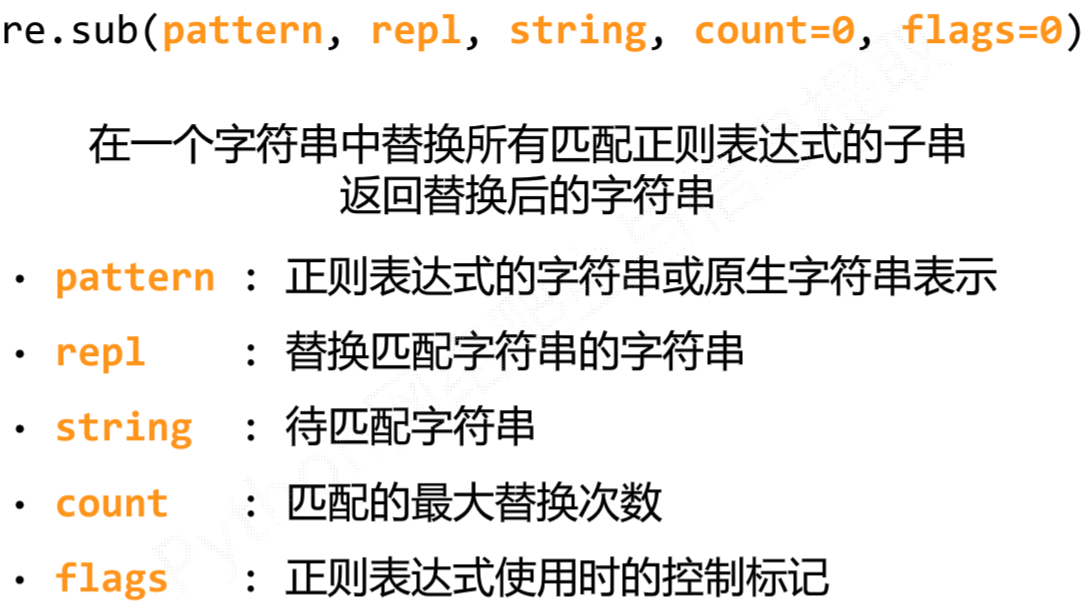
 1.

 2.

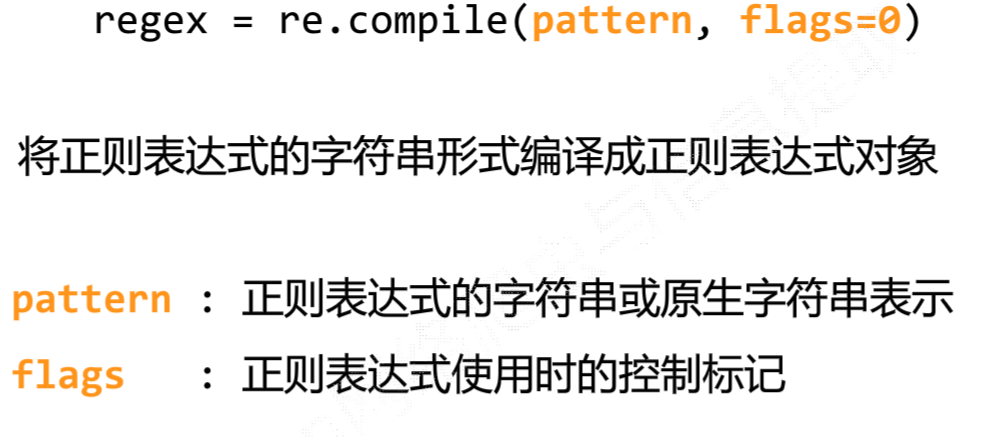
 3.

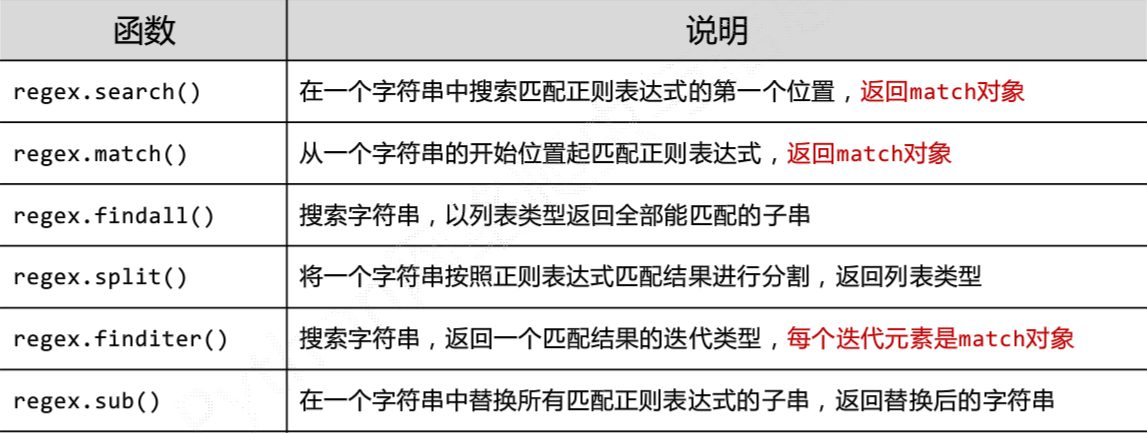
4.

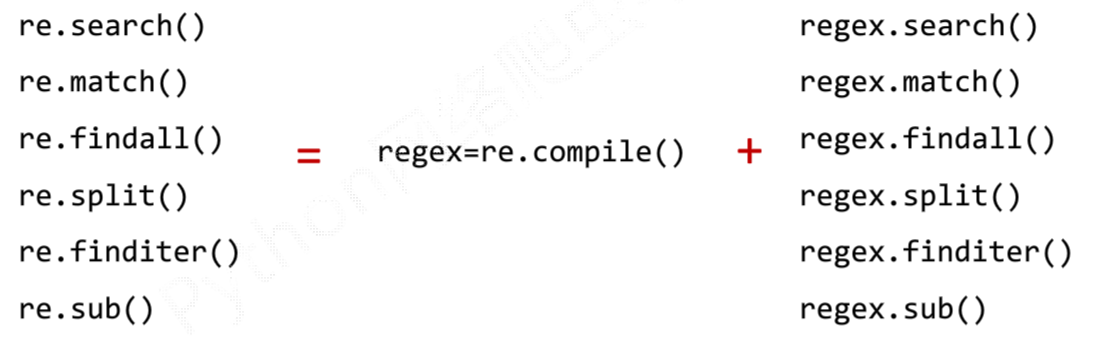
 5.

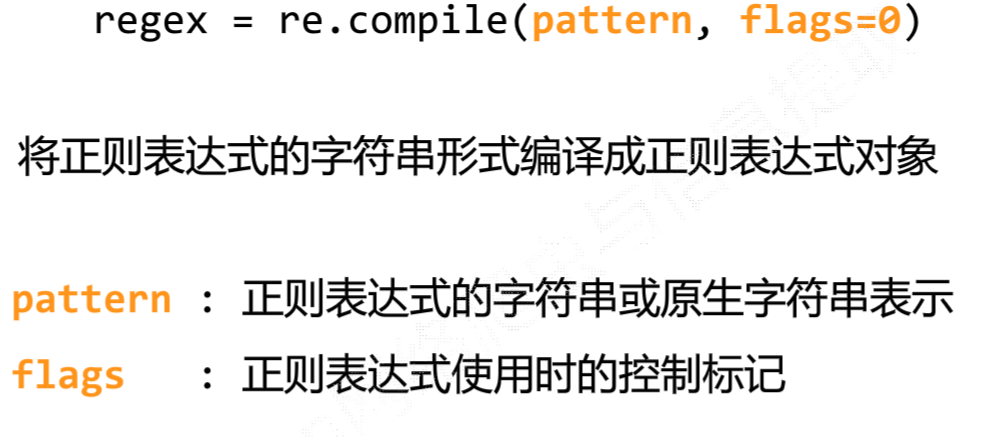
 6.

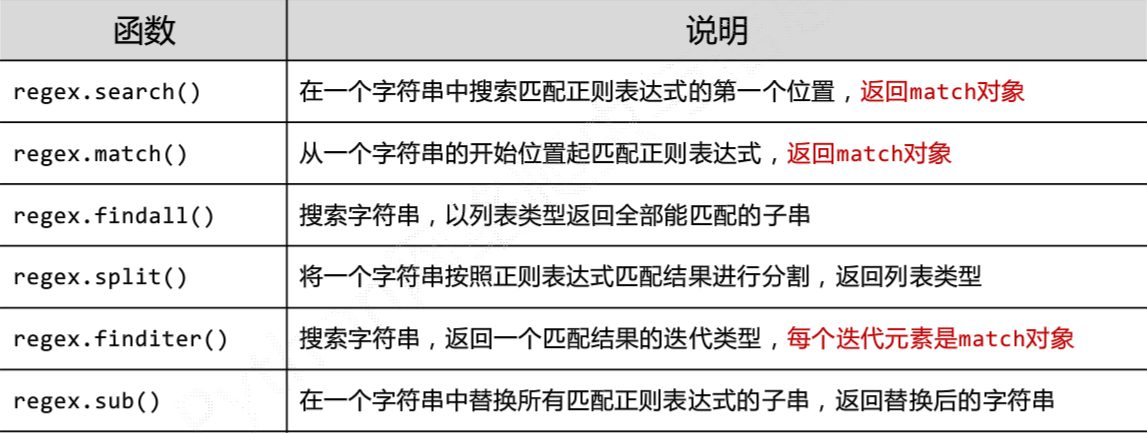
 re库的等价用法：







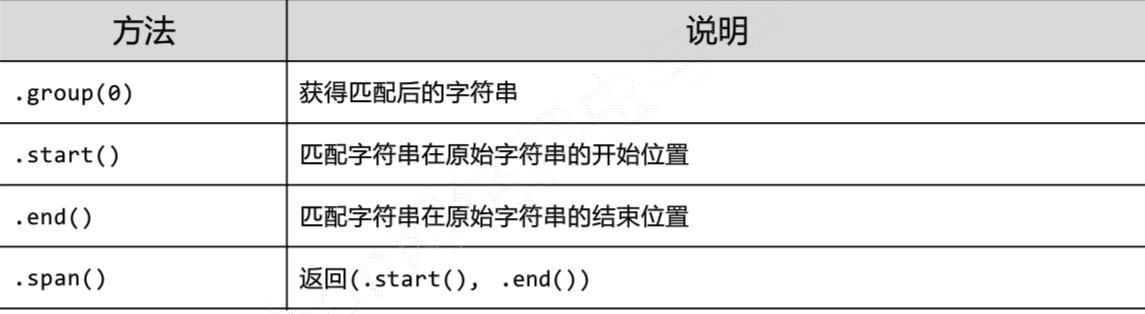


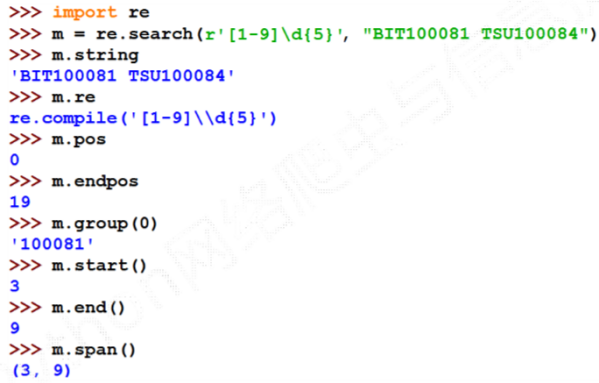


match对象：

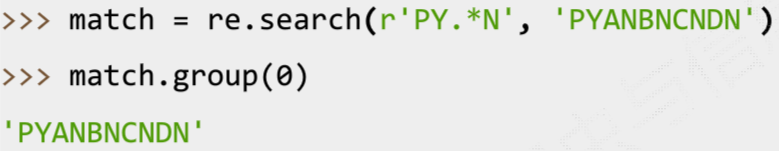
即正则表达式匹配一次返回的结果，其包含匹配的各种信息

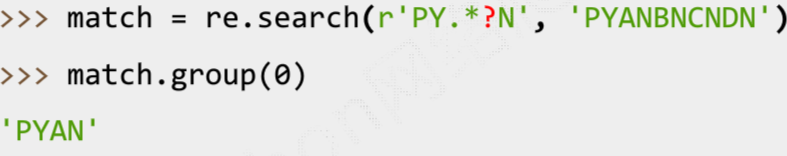
 match对象的属性：

 match对象的方法：

 操作实例：

re库默认贪婪匹配，即匹配最长的字符串

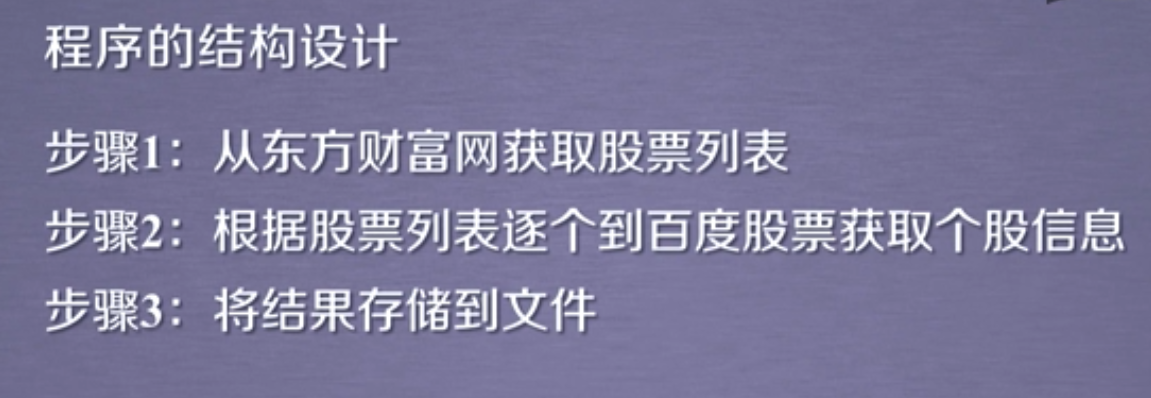






只要输出长度可能不同，都可以在操作符后增加？变成最小匹配

股票定向爬虫的结构框架：



1. 东方财富网股票列表下架，数据来源需要另行获取
2. 定向爬虫的基本结构，调试时寻找意外退出的部分

Scrapy框架：

