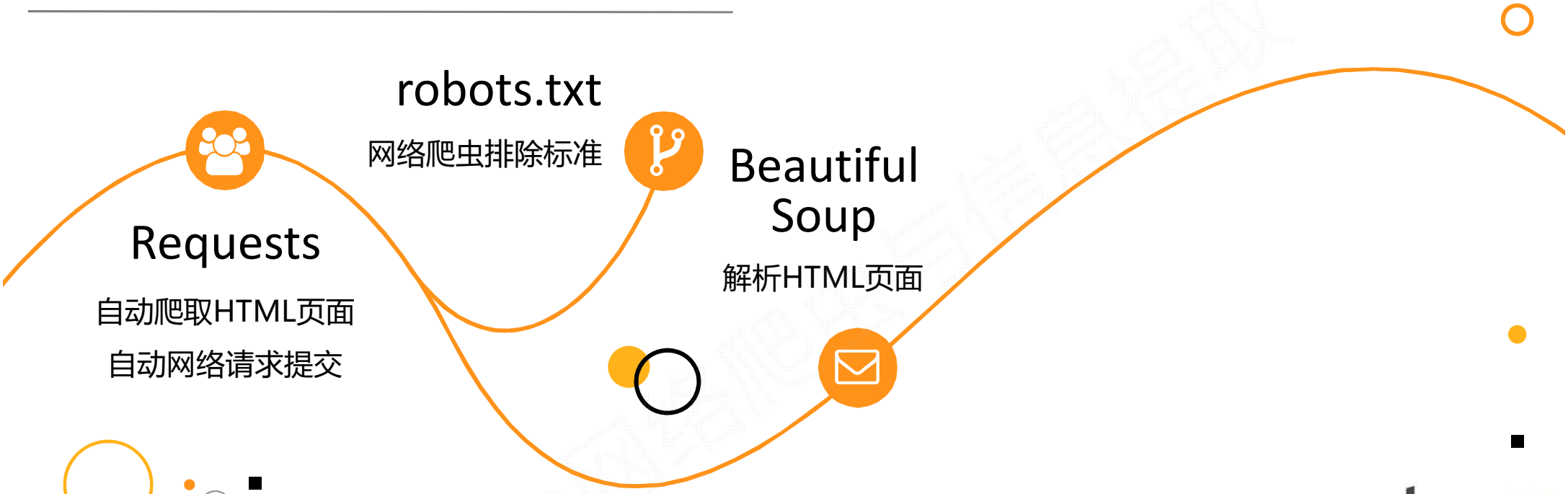
Beautiful Soup库入门

WS04

嵩天

www.python123.org

The Website is the API ...

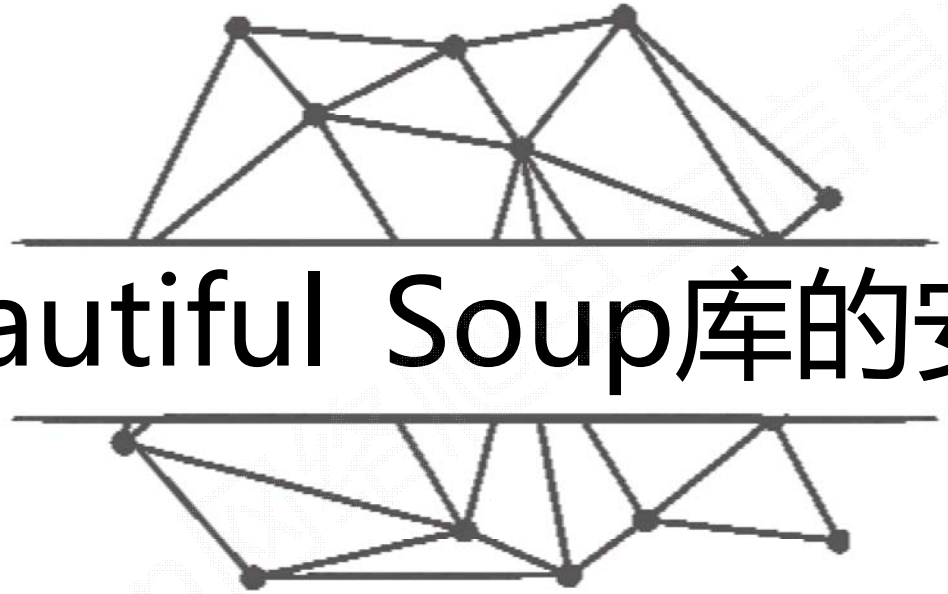


掌握定向网络数据爬取和网页解析的基本能力

python
弹指之间 · 享受创新

Python网络爬虫与信息提取

O4X -Tian



Beautiful Soup库的安装

https://www.crummy.com/software/BeautifulSoup/

You didn't write that awful page. You're just trying to get some data out of it. Beautiful Soup is here to help. Since 2004, it's been saving programmers hours or days of work on quick-turnaround screen scraping projects.

Beautiful Soup

"A tremendous boon." -- Python411 Podcast

[Download | Documentation | Hall of Fame | Source | Discussion group]

If Beautiful Soup has saved you a lot of time and money, the best way to pay me back is to check out *Constellation Games*, my sci-fi novel about alien video games.
You can read the first two chapters for free, and the full novel starts at 5 USD. Thanks!

If you have questions, send them to the discussion group. If you find a bug, file it.

Beautiful Soup is a Python library designed for quick turnaround projects like screen-scraping. Three features make it powerful:

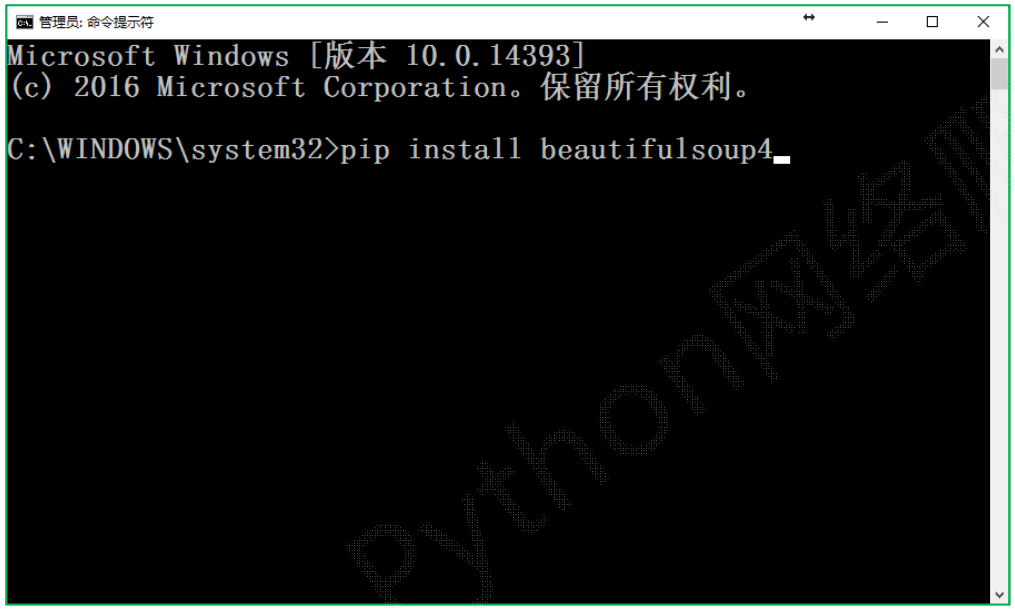
1. Beautiful Soup provides a few simple methods and Pythonic idioms for navigating, searching, and modifying a parse tree: a toolkit for dissecting a document and extracting what you need. It doesn't take much code to write an application
2. Beautiful Soup automatically converts incoming documents to Unicode and outgoing documents to UTF-8. You don't have to think about encodings, unless the document doesn't specify an encoding and Beautiful Soup can't detect one. Then you just have to specify the original encoding.
3. Beautiful Soup sits on top of popular Python parsers like lxml and html5lib, allowing you to try out different parsing strategies or trade speed for flexibility.

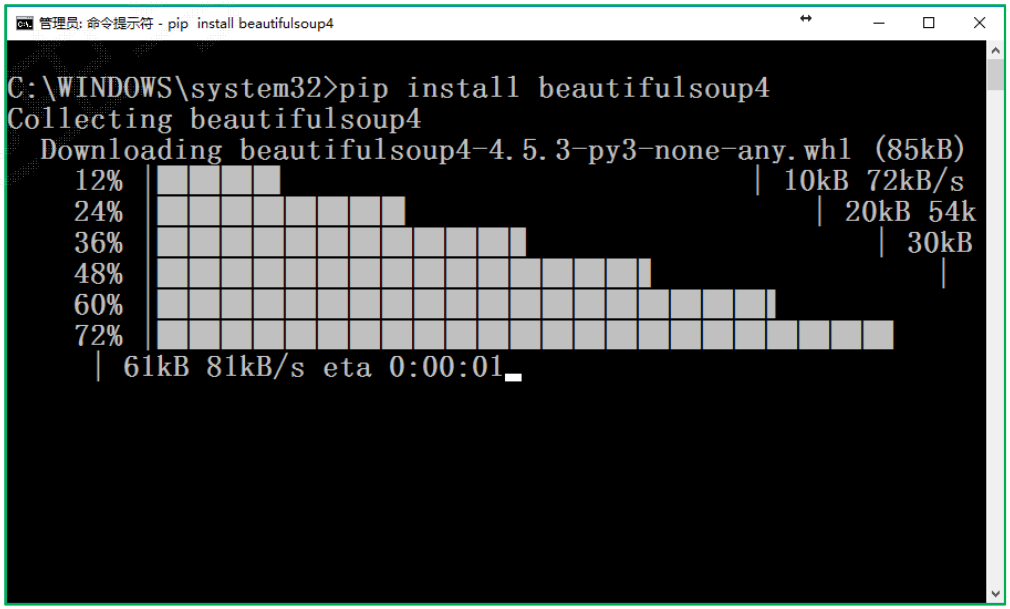
Beautiful Soup parses anything you give it, and does the tree traversal stuff for you. You can tell it "Find all the links", or "Find all the links of class externalLink", or "Find all the links whose urls match "foo.com", or "Find the table heading that's got bold text, then give me that text."

Beautiful Soup库的安装

Win平台：“以管理员身份运行”cmd

执行 pip install beautifulsoup4





Beautiful Soup库的安装小测

演示HTML页面地址：http://python123.io/ws/demo.html

文件名称：demo.html

Beautiful Soup库的安装小测

页面源代码：HTML 5.0格式代码

```
<html><head><title>This is a python demo page</title></head>
<body>
<p class="title"><b>The demo python introduces several python courses.</b></p>
<p class="course">Python is a wonderful general-purpose programming language. You can
learn Python from novice to professional by tracking the following courses:
<a href="http://www.icourse163.org/course/BIT-268001" class="py1" id="link1">Basic
Python</a> and <a href="http://www.icourse163.org/course/BIT-1001870001" class="py2"
id="link2">Advanced Python</a>.</p>
</body></html>
```

Beautiful Soup库的安装小测

手工获取demo.html源代码，浏览器右键“查看源代码”：

```
>>> demo = '''
<html><head><title>This is a python demo page</title></head>
<body>
<p class="title"><b>The demo python introduces several python courses.</b></p>
<p class="course">Python is a wonderful general-purpose programming language.
You can learn Python from novice to professional by tracking the following cou
rses:
<a href="http://www.icourse163.org/course/BIT-268001" class="py1" id="link1">B
asic Python</a> and <a href="http://www.icourse163.org/course/BIT-1001870001"
class="py2" id="link2">Advanced Python</a>.</p>
</body></html>'''
```

Beautiful Soup库的安装小测

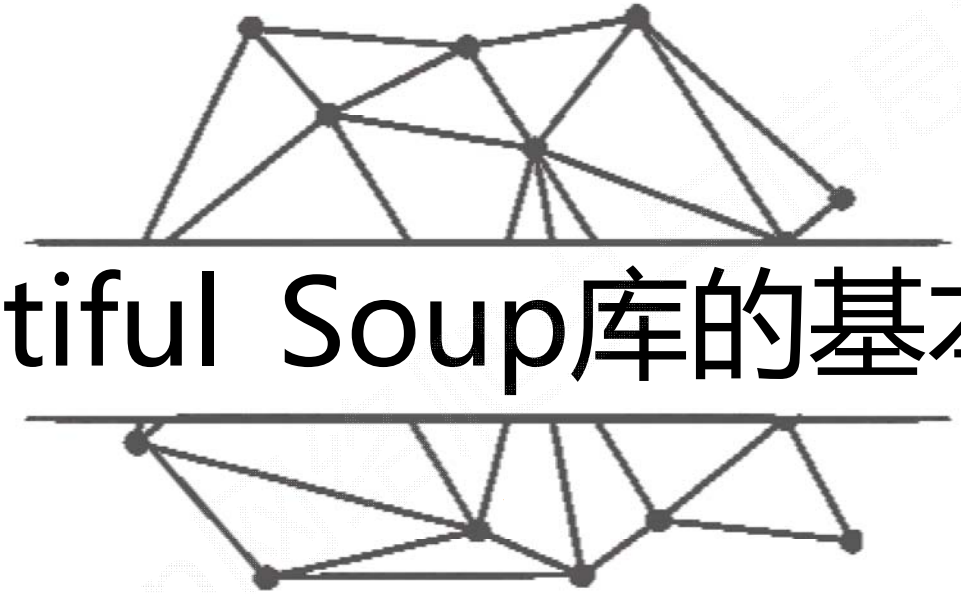
或者，用Requests库获取demo.html源代码：

```
>>> import requests
>>> r = requests.get("http://python123.io/ws/demo.html")
>>> r.text
'<html><head><title>This is a python demo page</title></head>\r\n<body>\r\n<p
class="title"><b>The demo python introduces several python courses.</b></p>\r\
n<p class="course">Python is a wonderful general-purpose programming language.
You can learn Python from novice to professional by tracking the following cou
rses:\r\n<a href="http://www.icourse163.org/course/BIT-268001" class="py1" id=
"link1">Basic Python</a> and <a href="http://www.icourse163.org/course/BIT-100
1870001" class="py2" id="link2">Advanced Python</a>.</p>\r\n</body></html>'
>>> demo = r.text
```

Beautiful Soup库的安装小测

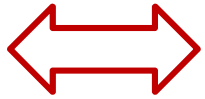
```
>>> from bs4 import BeautifulSoup
>>> soup = BeautifulSoup(demo, 'html.parser')
>>> print(soup.prettify())
<html>
 <head>
  <title>
   This is a python demo page
  </title>
 </head>
 <body>
  <p class="title">
   <b>
    The demo python introduces several python courses.
   </b>
  </p>
  <p class="course">
   Python is a wonderful general-purpose programming language. You can learn P
ython from novice to professional by tracking the following courses:
   <a class="py1" href="http://www.icourse163.org/course/BIT-268001" id="link1
">
```

```
from bs4 import BeautifulSoup

soup = BeautifulSoup('<p>data</p>', 'html.parser')
```



Beautiful Soup库的基本元素

Beautiful Soup库的理解

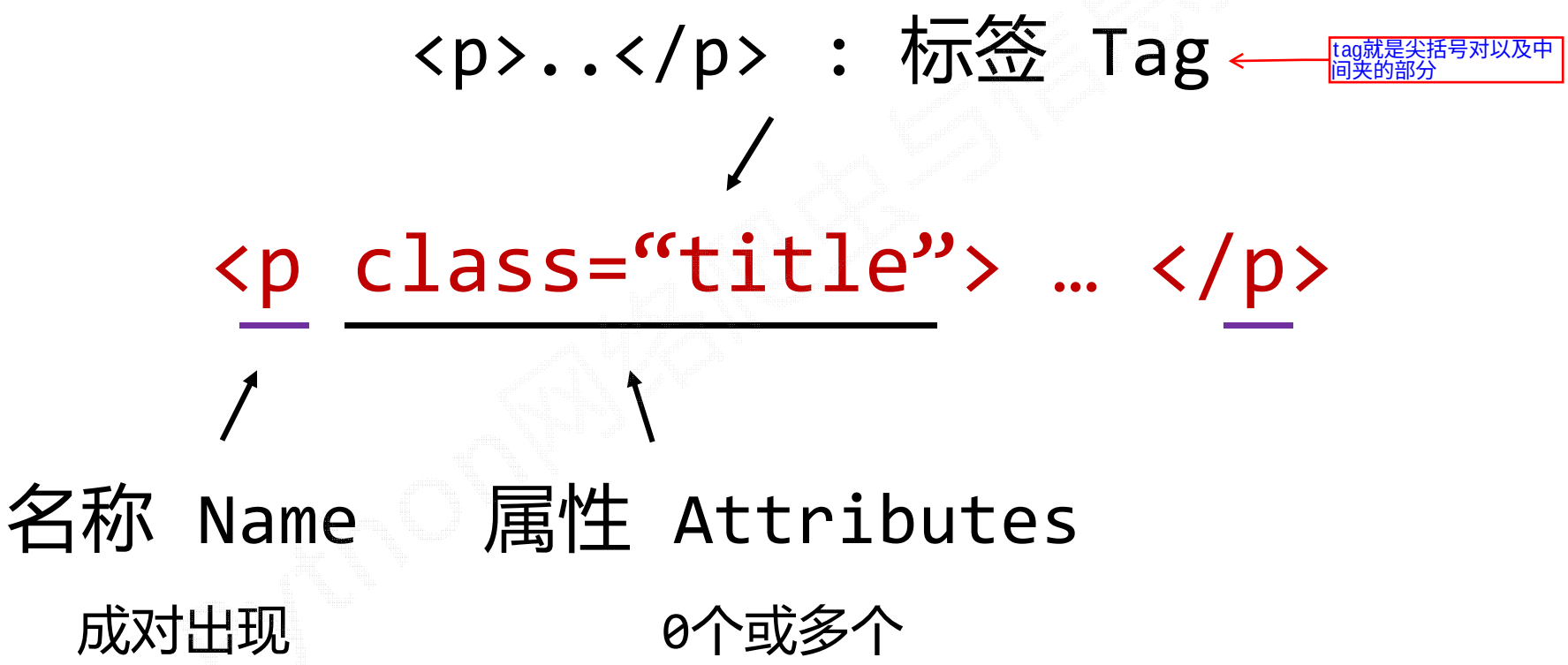


```
<html>
    <body>
        <p class="title"> … </p>
    </body>
</html>
```

标签树

Beautiful Soup库是解析、遍历、维护“标签树”的功能库

Beautiful Soup库的理解



Beautiful Soup库的引用

Beautiful Soup库，也叫beautifulsoup4 或 bs4

约定引用方式如下，即主要是用BeautifulSoup类

```
from bs4 import BeautifulSoup
import bs4
```

BeautifulSoup类

⇔ 标签树 ⇔ BeautifulSoup类

```
>>> from bs4 import BeautifulSoup
>>> soup  =  BeautifulSoup("<html>data</html>", "html.parser")
>>> soup2 =  BeautifulSoup(open("D://demo.html"), "html.parser")
```

BeautifulSoup对应一个HTML/XML文档的全部内容

Beautiful Soup库解析器

```
soup = BeautifulSoup('<html>data</html>' , 'html.parser')
```

解析器	使用方法	条件
bs4的HTML解析器	BeautifulSoup(mk,'html.parser')	安装bs4库
lxml的HTML解析器	BeautifulSoup(mk,'lxml')	pip install lxml
lxml的XML解析器	BeautifulSoup(mk,'xml')	pip install lxml
html5lib的解析器	BeautifulSoup(mk,'html5lib')	pip install html5lib

BeautifulSoup类的基本元素

 …

基本元素	说明
Tag	标签，最基本的信息组织单元，分别用<>和</>标明开头和结尾
Name	标签的名字， … 的名字是'p'，格式：<tag>.name
Attributes	标签的属性，字典形式组织，格式：<tag>.attrs
NavigableString	标签内非属性字符串，<>…</>中字符串，格式：<tag>.string
Comment	标签内字符串的注释部分，一种特殊的Comment类型

回顾demo.html

演示HTML页面地址：http://python123.io/ws/demo.html

回顾demo.html

```
>>> import requests
>>> r = requests.get("http://python123.io/ws/demo.html")
>>> demo = r.text
>>> demo
'<html><head><title>This is a python demo page</title></head>\r\n<body>\r\n<p
class="title"><b>The demo python introduces several python courses.</b></p>\r\
n<p class="course">Python is a wonderful general-purpose programming language.
You can learn Python from novice to professional by tracking the following cou
rses:\r\n<a href="http://www.icourse163.org/course/BIT-268001" class="py1" id=
"link1">Basic Python</a> and <a href="http://www.icourse163.org/course/BIT-100
1870001" class="py2" id="link2">Advanced Python</a>.</p>\r\n</body></html>'
>>>
```

Tag 标签

基本元素	说明
Tag	标签，最基本的信息组织单元，分别用<>和</>标明开头和结尾

```
>>> from bs4 import BeautifulSoup
>>> soup = BeautifulSoup(demo, "html.parser")
>>> soup.title
<title>This is a python demo page</title>
>>> tag = soup.a
>>> tag
<a class="py1" href="http://www.icourse163.org/course/BIT-268001" id="link1">Bas
ic Python</a>
```

任何存在于HTML语法中的标签都可以用soup.<tag>访问获得

当HTML文档中存在多个相同<tag>对应内容时，soup.<tag>返回第一个

Tag的name（名字）

基本元素	说明
Name	标签的名字， … 的名字是'p'，格式：<tag>.name

```
>>> from bs4 import BeautifulSoup
>>> soup = BeautifulSoup(demo, "html.parser")
>>> soup.a.name
'a'
>>> soup.a.parent.name
'p'
>>> soup.a.parent.parent.name
'body'
>>>
```

每个<tag>都有自己的名字，通过<tag>.name获取，字符串类型

Tag的attrs（属性）

基本元素	说明
Attributes	标签的属性，字典形式组织，格式：<tag>.attrs

```
>>> tag = soup.a
>>> tag.attrs
{'id': 'link1', 'class': ['py1'], 'href': 'http://www.icours
e163.org/course/BIT-268001'}
>>> tag.attrs['class']
['py1']
>>> tag.attrs['href']
'http://www.icourse163.org/course/BIT-268001'
>>> type(tag.attrs)
<class 'dict'>
>>> type(tag)
<class 'bs4.element.Tag'>
```

一个<tag>可以有0或多个属性，字典类型

Tag的NavigableString

基本元素	说明
NavigableString	标签内非属性字符串，<>…</>中字符串，格式：<tag>.string

```
>>> soup.a
<a class="py1" href="http://www.icourse163.org/course/BIT-26
8001" id="link1">Basic Python</a>
>>> soup.a.string
'Basic Python'
>>> soup.p
<p class="title"><b>The demo python introduces several pytho
n courses.</b></p>
>>> soup.p.string
'The demo python introduces several python courses.'
>>> type(soup.p.string)
<class 'bs4.element.NavigableString'>
```

NavigableString可以跨越多个层次

Tag的Comment

基本元素	说明
Comment	标签内字符串的注释部分，一种特殊的Comment类型

```
>>> newsoup = BeautifulSoup("<b><!--This is a comment--></
b><p>This is not a comment</p>","html.parser")
>>> newsoup.b.string
'This is a comment'
>>> type(newsoup.b.string)
<class 'bs4.element.Comment'>
>>> newsoup.p.string
'This is not a comment'
>>> type(newsoup.p.string)
<class 'bs4.element.NavigableString'>
```

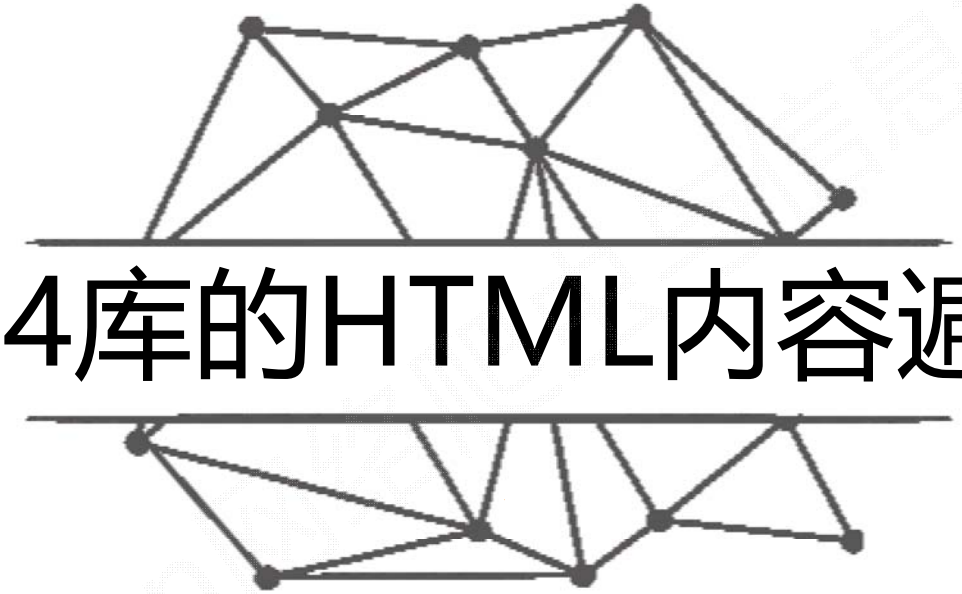
Comment是一种特殊类型

BeautifulSoup类的基本元素

 …

基本元素	说明
Tag	标签，最基本的信息组织单元，分别用<>和</>标明开头和结尾
Name	标签的名字， … 的名字是'p'，格式：<tag>.name
Attributes	标签的属性，字典形式组织，格式：<tag>.attrs
NavigableString	标签内非属性字符串，<>…</>中字符串，格式：<tag>.string
Comment	标签内字符串的注释部分，一种特殊的Comment类型


标签 <tag>
 …
名称 .name
属性 .attrs
非属性字符串/注释
.string



基于bs4库的HTML内容遍历方法

回顾demo.html

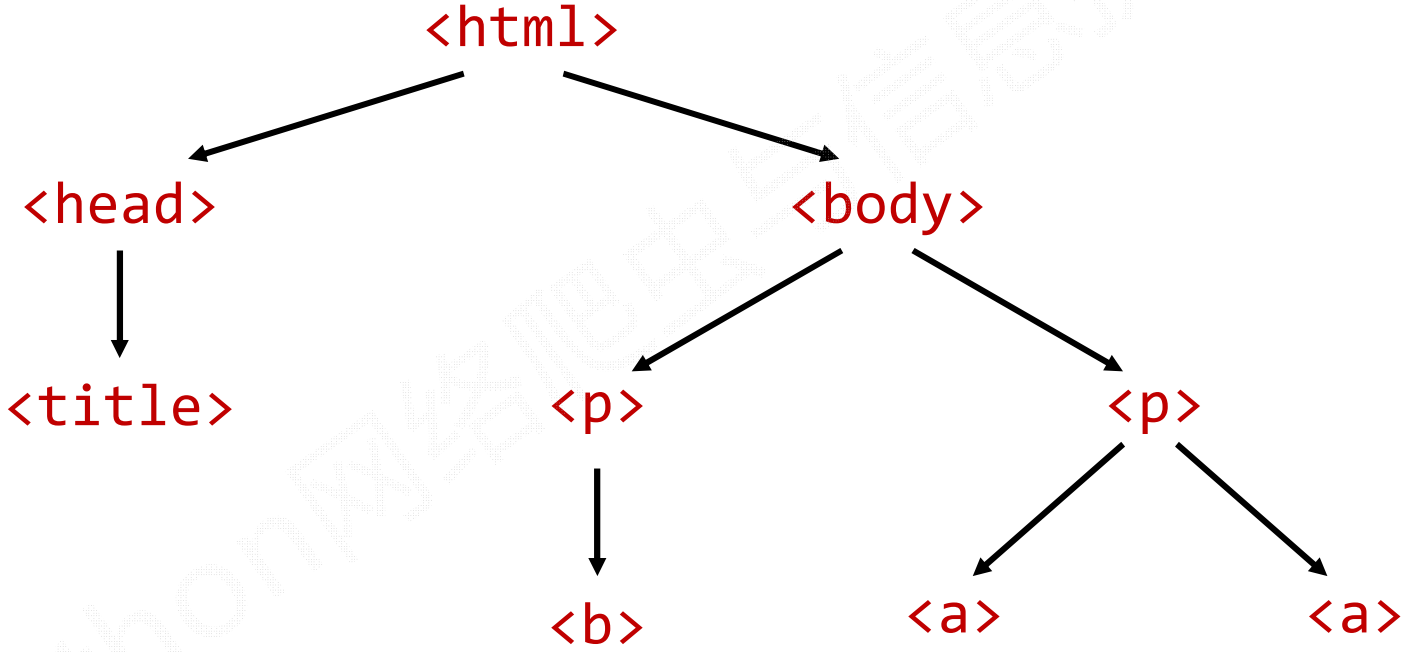
```
>>> import requests
>>> r = requests.get("http://python123.io/ws/demo.html")
>>> demo = r.text
>>> demo
'<html><head><title>This is a python demo page</title></head>\r\n<body>\r\n<p
class="title"><b>The demo python introduces several python courses.</b></p>\r\
n<p class="course">Python is a wonderful general-purpose programming language.
You can learn Python from novice to professional by tracking the following cou
rses:\r\n<a href="http://www.icourse163.org/course/BIT-268001" class="py1" id=
"link1">Basic Python</a> and <a href="http://www.icourse163.org/course/BIT-100
1870001" class="py2" id="link2">Advanced Python</a>.</p>\r\n</body></html>'
>>>
```

HTML基本格式

```
<html>
    <head>
        <title>This is a python demo page</title>
    </head>
    <body>
        <p class="title">
          <b>The demo python introduces several python courses.</b>
        </p>
        <p class=“course”>Python is a wonderful general-purpose programming language.
You can learn Python from novice to professional by tracking the following courses:
            <a href="http://www.icourse163.org/course/BIT-268001" class="py1"
id="link1">Basic Python</a> and
            <a href="http://www.icourse163.org/course/BIT-1001870001" class="py2"
id="link2">Advanced Python</a>.
        </p>
    </body>
</html>
```

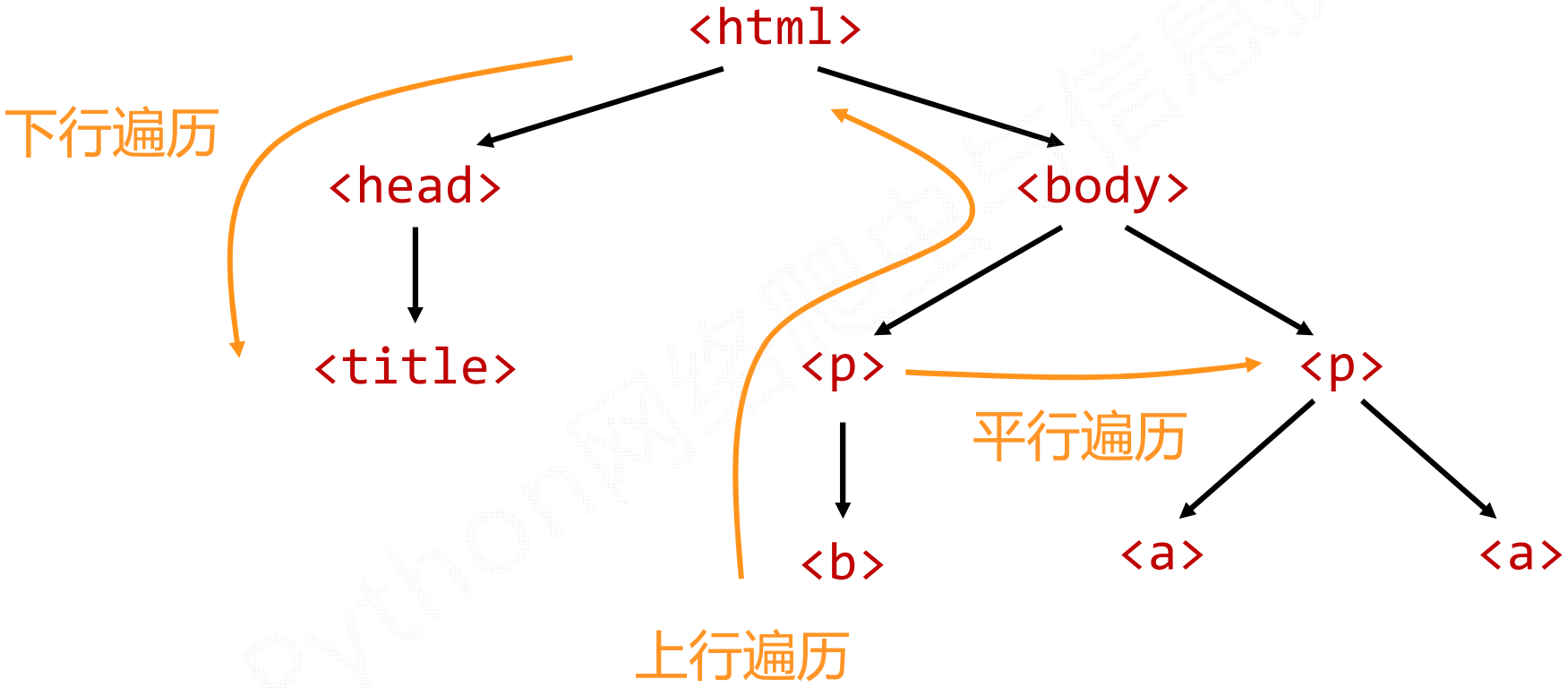
<>…</>构成了所属关系

HTML基本格式



<>…</>构成了所属关系，形成了标签的树形结构

HTML基本格式



标签树的下行遍历

属性	说明
.contents	子节点的列表，将<tag>所有儿子节点存入列表
.children	子节点的迭代类型，与.contents类似，用于循环遍历儿子节点
.descendants	子孙节点的迭代类型，包含所有子孙节点，用于循环遍历

BeautifulSoup类型是标签树的根节点

标签树的下行遍历

```
>>> soup = BeautifulSoup(demo, "html.parser")
>>> soup.head
<head><title>This is a python demo page</title></head>
>>> soup.head.contents
[<title>This is a python demo page</title>]
>>> soup.body.contents
['\n', <p class="title"><b>The demo python introduces several python cours
es.</b></p>, '\n', <p class="course">Python is a wonderful general-purpose
programming language. You can learn Python from novice to professional by
tracking the following courses:
<a class="py1" href="http://www.icourse163.org/course/BIT-268001" id="link
1">Basic Python</a> and <a class="py2" href="http://www.icourse163.org/cou
rse/BIT-1001870001" id="link2">Advanced Python</a>.</p>, '\n']
>>> len(soup.body.contents)
5
>>> soup.body.contents[1]
<p class="title"><b>The demo python introduces several python courses.</b>
</p>
```

标签树的下行遍历

```
for child in soup.body.children:
    print(child)
```

遍历儿子节点

```
for child in soup.body.descendants:
    print(child)
```

遍历子孙节点

标签树的上行遍历

属性	说明
.parent	节点的父亲标签
.parents	节点先辈标签的迭代类型，用于循环遍历先辈节点

标签树的上行遍历

```
>>> soup = BeautifulSoup(demo, "html.parser")
>>> soup.title.parent
<head><title>This is a python demo page</title></head>
>>> soup.html.parent
<html><head><title>This is a python demo page</title></head>
<body>
<p class="title"><b>The demo python introduces several python courses.</b></p>
<p class="course">Python is a wonderful general-purpose programming language. Yo
u can learn Python from novice to professional by tracking the following courses
:
<a class="py1" href="http://www.icourse163.org/course/BIT-268001" id="link1">Bas
ic Python</a> and <a class="py2" href="http://www.icourse163.org/course/BIT-1001
870001" id="link2">Advanced Python</a>.</p>
</body></html>
>>> soup.parent
>>>
```

标签树的上行遍历

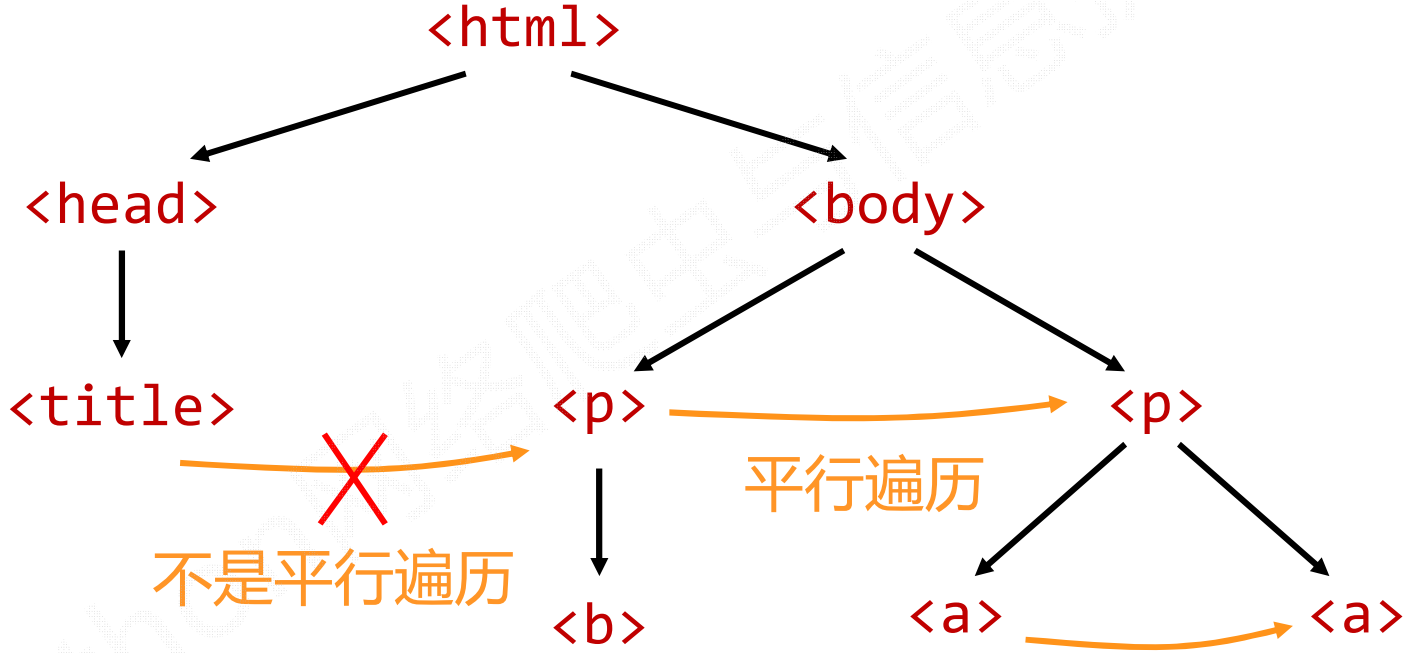
```
>>> soup = BeautifulSoup(demo, "html.parser")
>>> for parent in soup.a.parents:
        if parent is None:
                print(parent)
        else:
                print(parent.name)

p
body
html
[document]
```

遍历所有先辈节点，包括soup本身，所以要区别判断

标签树的平行遍历

属性	说明
.next_sibling	返回按照HTML文本顺序的下一个平行节点标签
.previous_sibling	返回按照HTML文本顺序的上一个平行节点标签
.next_siblings	迭代类型，返回按照HTML文本顺序的后续所有平行节点标签
.previous_siblings	迭代类型，返回按照HTML文本顺序的前续所有平行节点标签

标签树的平行遍历



平行遍历发生在同一个父节点下的各节点间

标签树的平行遍历

```
>>> soup = BeautifulSoup(demo, "html.parser")
>>> soup.a.next_sibling
' and '
>>> soup.a.next_sibling.next_sibling
<a class="py2" href="http://www.icourse163.org/course/BIT-1001870001" id="
link2">Advanced Python</a>
>>> soup.a.previous_sibling
'Python is a wonderful general-purpose programming language. You can learn
Python from novice to professional by tracking the following courses:\r\n'
>>> soup.a.previous_sibling.previous_sibling
>>> soup.a.parent
<p class="course">Python is a wonderful general-purpose programming langua
ge. You can learn Python from novice to professional by tracking the follo
wing courses:
<a class="py1" href="http://www.icourse163.org/course/BIT-268001" id="link
1">Basic Python</a> and <a class="py2" href="http://www.icourse163.org/cou
rse/BIT-1001870001" id="link2">Advanced Python</a>.</p>
```

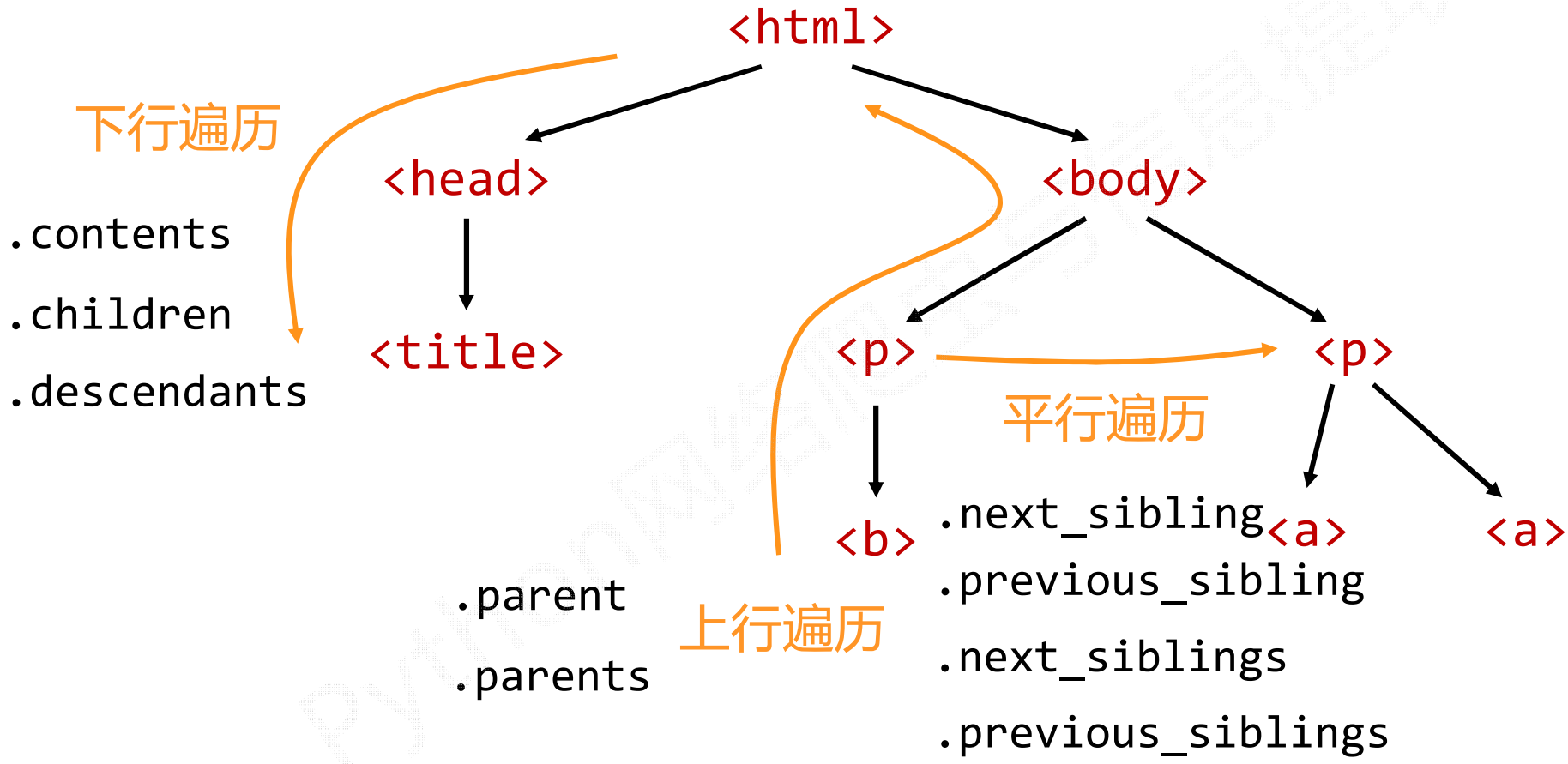
标签树的平行遍历

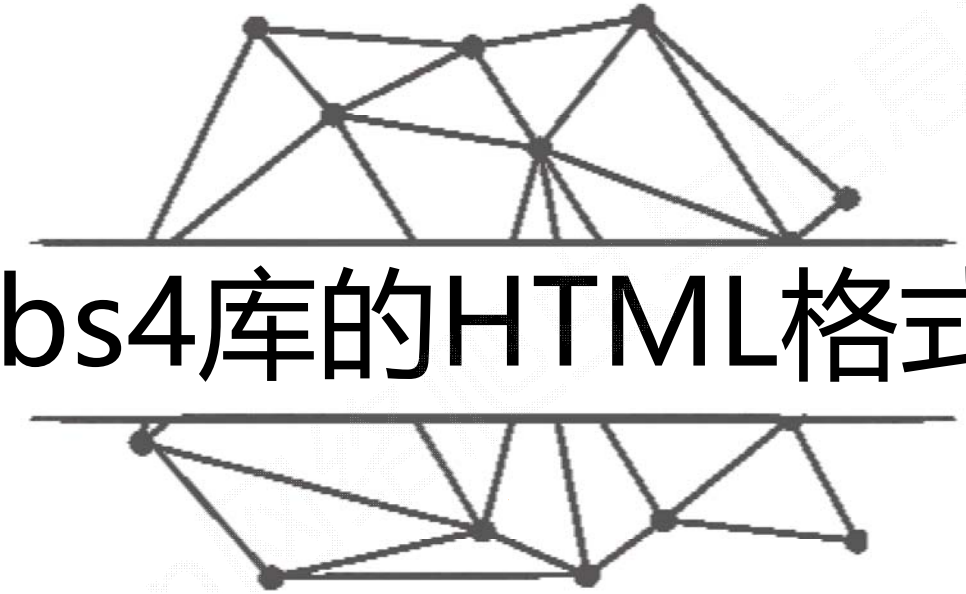
```
for sibling in soup.a.next_sibling:
    print(sibling)
```

遍历后续节点

```
for sibling in soup.a.previous_sibling:
    print(sibling)
```

遍历前续节点


<html>
下行遍历
<head>
<body>
.contents
.children
.descendants
<title>
<p>
<p>
平行遍历
<b>
.next_sibling
<a>
<a>
.previous_sibling
.parent
上行遍历
.next_siblings
.parents
.previous_siblings



基于bs4库的HTML格式输出

回顾demo.html

```
>>> import requests
>>> r = requests.get("http://python123.io/ws/demo.html")
>>> demo = r.text
>>> demo
'<html><head><title>This is a python demo page</title></head>\r\n<body>\r\n<p
class="title"><b>The demo python introduces several python courses.</b></p>\r\
n<p class="course">Python is a wonderful general-purpose programming language.
You can learn Python from novice to professional by tracking the following cou
rses:\r\n<a href="http://www.icourse163.org/course/BIT-268001" class="py1" id=
"link1">Basic Python</a> and <a href="http://www.icourse163.org/course/BIT-100
1870001" class="py2" id="link2">Advanced Python</a>.</p>\r\n</body></html>'
>>>
```

能否让HTML内容更加“友好”的显示？

bs4库的prettify()方法

```
>>> import requests
>>> r = requests.get("http://python123.io/ws/demo.html")
>>> demo = r.text
>>> demo
'<html><head><title>This is a python demo page</title></head>\r\n<body>\r\n<p
class="title"><b>The demo python introduces several python courses.</b></p>\r\
n<p class="course">Python is a wonderful general-purpose programming language.
You can learn Python from novice to professional by tracking the following cou
rses:\r\n<a href="http://www.icourse163.org/course/BIT-268001" class="py1" id=
"link1">Basic Python</a> and <a href="http://www.icourse163.org/course/BIT-100
1870001" class="py2" id="link2">Advanced Python</a>.</p>\r\n</body></html>'
>>>
```

bs4库的prettify()方法

```
>>> from bs4 import BeautifulSoup
>>> soup = BeautifulSoup(demo, "html.parser")
>>> soup.prettify()
'<html>\n <head>\n  <title>\n   This is a python demo page
\n  </title>\n </head>\n <body>\n  <p class="title">\n   <
b>\n    The demo python introduces several python courses.
\n   </b>\n  </p>\n  <p class="course">\n   Python is a wo
nderful general-purpose programming language. You can lear
n Python from novice to professional by tracking the follo
wing courses:\n   <a class="py1" href="http://www.icourse1
63.org/course/BIT-268001" id="link1">\n    Basic Python\n
 </a>\n   and\n   <a class="py2" href="http://www.icourse1
63.org/course/BIT-1001870001" id="link2">\n    Advanced Py
thon\n   </a>\n   .\n  </p>\n </body>\n</html>'
```

bs4库的prettify()方法

```
>>> print(soup.prettify())
<html>
 <head>
  <title>
   This is a python demo page
  </title>
 </head>
 <body>
  <p class="title">
   <b>
    The demo python introduces several python courses.
   </b>
  </p>
```

bs4库的prettify()方法

.prettify()为HTML文本<>及其内容增加更加'\n'

.prettify()可用于标签，方法：<tag>.prettify()

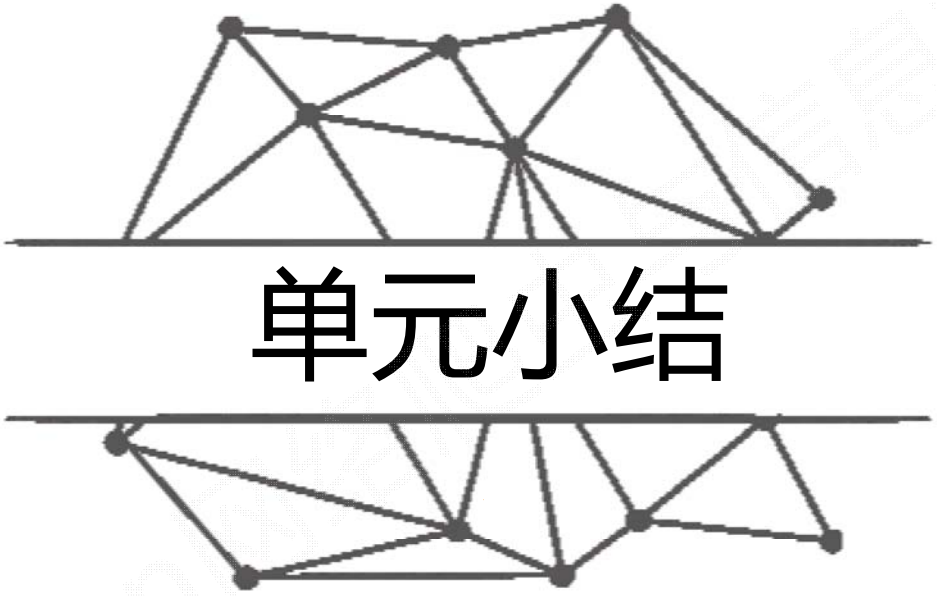
```
>>> print(soup.a.prettify())
<a class="py1" href="http://www.icourse163.org/course/BIT-
268001" id="link1">
 Basic Python
</a>
```

bs4库的编码

bs4库将任何HTML输入都变成utf-8编码

Python 3.x默认支持编码是utf-8,解析无障碍

```
>>> soup = BeautifulSoup("<p>中文</p>", "html.parser")
>>> soup.p.string
'中文'
>>> print(soup.p.prettify())
<p>
 中文
</p>
```



单元小结

Beautiful Soup库入门

```
from bs4 import BeautifulSoup

soup = BeautifulSoup('<p>data</p>', 'html.parser')
```

Tag　Name　Attributes NavigableString　Comment	.contents .children .descendants
bs4库的基本元素	bs4库的遍历功能

.parent

.parents

.next_sibling

.previous_sibling

.next_siblings

.previous_siblings