HTML Related

ZhangXu

2019 年 1 月 23 日

1 前言

上承接Python基本的获取数据方法。关于数据的解析提取工作，可使用re,bs4,xpath等，暂时不深究。

接下来便是对动态网页的提取，觉得有必要事先了解一下HTML相关的内容。

第一部分 HTML/CSS

w3school的教程

- HTML 指的是超文本标记语言 (Hyper Text Markup Language)
- HTML 不是一种编程语言，而是一种标记语言 (markup language)
- 标记语言是一套标记标签 (markup tag)
- HTML 使用标记标签来描述网页

2 基础

HTML标题 HTML标题（Heading）是通过 $< h1 >$ - $< h6 >$ 等标签进行定义的。

段落 $< p >$标签定义

链接 $< ahref = "" >$ content $< /a >$

图像 $< imgsrc = ""width = ""height = ""/ >$

3 元素

HTML元素指的是从开始标签（start tag）到结束标签（end tag）的所有代码。其定义HTML文档。

3.1 元素语法

- **开始标签**起始
- **结束标签**终止
- **元素的内容**是开始标签与结束标签之间的内容
- 某些 HTML 元素具有**空内容**（empty content）
- 空元素在**开始标签中进行关闭**（以开始标签的结束而结束）
- 大多数 HTML 元素可拥有**属性**

3.2 空元素

没有内容的 HTML 元素被称为空元素。空元素是在开始标签中关闭的。
$< br >$ 就是没有关闭标签的空元素（$< br >$ 标签定义换行）。
在 XHTML、XML 以及未来版本的 HTML 中，所有元素都必须被关闭。

在开始标签中添加斜杠，比如 $< br/ >$，是关闭空元素的正确方法，HTML、XHTML 和 XML 都接受这种方式。

即使 $< br >$ 在所有浏览器中都是有效的，但使用 $< br/ >$ 其实是更长远的保障。

3.3 TIPs

HTML 标签对大小写不敏感：$< P >$ 等同于 $< p >$。许多网站都使用大写的 HTML 标签。
万维网联盟（W3C）在 HTML 4 中推荐使用小写，而在未来 (X)HTML 版本中强制使用小写。

4 HTML属性

属性为 HTML 元素提供附加信息。

4.1 属性

属性总是以名称/值对的形式出现，比如：name="value"。
属性总是在 HTML 元素的开始标签中规定。

4.2 TIPs

属性和属性值对大小写不敏感。万维网联盟在其 HTML 4 推荐标准中推荐小写的属性/属性值。而新版本的 (X)HTML 要求使用小写属性。
始终为属性值加引号

4.3 属性参考手册

完整的 HTML 参考手册

- class classname 规定元素的类名（classname）
- id id 规定元素的唯一 id
- style style_definition 规定元素的行内样式（inline style）
- title text 规定元素的额外信息（可在工具提示中显示

5 标题

标题（Heading）是通过 $<h1>$ - $<h6>$ 等标签进行定义的。
搜索引擎使用标题为您的网页的结构和内容编制索引。
$<hr/>$ 标签在 HTML 页面中创建水平线。hr 元素可用于分隔内容。
注释: $<!--Thisisacomment-->$

6 段落

$<p>$标签定义。

6.1 输出

对于 HTML，您无法通过在 HTML 代码中添加额外的空格或换行来改变输出的效果。
当显示页面时，浏览器会移除源代码中多余的空格和空行。所有连续的空格或空行都会被算作一个空格。需要注意的是，HTML 代码中的所有连续的空行（换行）也被显示为一个空格。

7 样式

style 属性用于改变 HTML 元素的样式。

7.1 style属性

提供了一种改变所有 HTML 元素的样式的通用方法
通过 HTML 样式，能够通过使用 style 属性直接将样式添加到 HTML 元素，或者间接地在独立的样式表中（CSS 文件）进行定义。

8 文本格式化

HTML 可定义很多供格式化输出的元素，比如粗体和斜体字。.

9 引用Quotation

$< q >$元素定义**短的引用**。浏览器通常会为 $< q >$ 元素包围引号
$< blockquote >$定义被引用的节。浏览器通常会对 $< blockquote >$ 元素进行**缩进**处理。
$< abbr >$定义缩写或首字母缩略语。
$< dfn >$
$< address >$定义文档联系作者，通常以斜体显示。
$< cite >$定义著作的标题

10 计算机代码元素

代码格式 通常，HTML 使用可变的字母尺寸，以及可变的字母间距。在显示计算机代码示例时，并不需要如此。$< kbd >$, $< samp >$, 以及 $< code >$ 元素

全都支持固定的字母尺寸和间距。

键盘格式 $<kbd>$ 元素定义键盘输入

样本格式 $<samp>$ 元素定义计算机输出示例

代码格式 $<code>$ 元素定义编程代码示例（元素不保留多余的空格和折行，如需解决该问题，您必须在 $<pre>$ 元素中包围代码）

变量格式化 $<var>$ 元素定义数学变量

11 注释

注释标签 <! − −ffi − − > 用于在 HTML 插入注释

条件注释 $<!--[if IE 8]>$.... some HTML here ....$<![endif]-->$

12 CSS

所有的格式化代码均可移出 HTML 文档，然后移入一个独立的样式表。

12.1 使用样式

- **外部样式** $<head><link rel="stylesheet" type="text/css" href="mystyle.css"></head>$
- **内部样式表**$<head><style type="text/css">$ body background-color: red p margin-left: 20px $</style></head>$
- **内联样式** $<p style="color:red;margin-left:20px">$ This is a paragraph $</p>$

CSS教

13 链接

语法 $$Link text$$

target属性 使用 Target 属性，你可以定义被链接的文档在何处显示。target="_blank"会在新窗口打开。

13.1 name属性

对读者不可见。
当使用命名锚（named anchors）时，我们可以创建直接跳至该命名锚（比如页面中某个小节）的链接，这样使用者就无需不停地滚动页面来寻找他们需要的信息了。

命名锚的语法 $< aname = "label" >$锚（显示在页面上的文本）$< /a >$（您可以使用 id 属性来替代 name 属性，命名锚同样有效。）
将 # 符号和锚名称添加到 URL 的末端，就可以直接链接到 tips 这个命名锚了。

14 图像

语法 $< imgsrc = "url" / >$

替换文本 Alt在浏览器无法载入图像时，替换文本属性告诉读者她们失去的信息

15 表格

15.1 表格

表格由 $< table >$ 标签来定义。每个表格均有若干行（由 $< tr >$ 标签定义），每行被分割为若干单元格（由 $< td >$ 标签定义）。字母 td 指表格数据（table data），即数据单元格的内容。数据单元格可以包含文本、图片、列表、段落、表单、水平线、表格等等。

15.2 表格和边框属性

border="1"

表头 $< th >$ 标签进行定义。大多数浏览器会把表头显示为粗体居中的文本：

16 列表

无序列表 无序列表是一个项目的列表，此列项目使用粗体圆点（典型的小黑圆圈）进行标记。无序列表始于 $< ul >$ 标签。每个列表项始于 $< li >$。

有序列表 同样，有序列表也是一列项目，列表项目使用数字进行标记。有序列表始于 $< ol >$ 标签。每个列表项始于 $< li >$ 标签

定义列表 自定义列表不仅仅是一列项目，而是项目及其注释的组合。自定义列表以 $< dl >$ 标签开始。每个自定义列表项以 $< dt >$ 开始。每个自定义列表项的定义以 $< dd >$ 开始。

17 div span

可以通过 $< div >$ 和 $< span >$ 将 HTML 元素组合起来

17.1 块元素

- 大多数 HTML 元素被定义为块级元素或内联元素。
- 块级元素在浏览器显示时，通常会以新行来开始（和结束）。

17.2 内联元素

内联元素在显示时通常不会以新行开始。

17.3 $< div >$元素

- $< div >$ 元素是块级元素，它是可用于组合其他 HTML 元素的容器
- 如果与 CSS 一同使用，$< div >$ 元素可用于对大的内容块设置样式属性。
- $< div >$ 元素的另一个常见的用途是文档布局。它取代了使用表格定义布局的老式方法。使用 $< table >$ 元素进行文档布局不是表格的正确用法。$< table >$ 元素的作用是显示表格化的数据。

17.4 $< span >$元素

- $< span >$ 元素是内联元素，可用作文本的容器
- $< span >$ 元素也没有特定的含义
- 当与 CSS 一同使用时，$< span >$ 元素可用于为部分文本设置样式属性。

17.5 分组标签

div 定义文档中的分区或节（division/section）。

span 定义 span，用来组合文档中的行内元素

18 类

对 HTML 进行分类（设置类），使我们能够为元素的类定义 CSS 样式。
为相同的类设置相同的样式，或者为不同的类设置不同的样式。

分级块元素 $< div >$ 元素是块级元素。它能够用作其他 HTML 元素的容器。
设置 $< div >$ 元素的类，使我们能够为相同的 $< div >$ 元素设置相同的类。

19 布局

19.1 div元素布局HTML

$< div >$ 元素常用作布局工具，因为能够轻松地通过 CSS 对其进行定位。

- header 定义文档或节的页眉
- nav 定义导航链接的容器
- section 定义文档中的节
- article 定义独立的自包含文章
- aside 定义内容之外的内容（比如侧栏）
- footer 定义文档或节的页脚
- details 定义额外的细节
- summary 定义 details 元素的标题

19.1.1 使用表格的HTML布局

注释: $< table >$ 元素不是作为布局工具而设计的。$< table >$ 元素的作用是显示表格化的数据。使用 $< table >$ 元素能够取得布局效果，因为能够通过 CSS 设置表格元素的样式

20 响应式Web设计

20.1 概念

- RWD 指的是响应式 Web 设计（Responsive Web Design）
- RWD 能够以可变尺寸传递网页
- RWD 对于平板和移动设备是必需的

20.1.1 使用Bootstrap

Bootstrap 是最流行的开发响应式 web 的 HTML, CSS, 和 JS 框架。 Bootstrap 帮助您开发在任何尺寸都外观出众的站点：显示器、笔记本电脑、平板电脑或手机

21 框架

通过使用框架，你可以在同一个浏览器窗口中显示不止一个页面。每份HTML文档称为一个框架，并且每个框架都独立于其他的框架。

- 开发人员必须同时跟踪更多的HTML文档
- 很难打印整张页面
- 框架结构标签（$< frameset >$）定义如何将窗口分割为框架
- 每个 frameset 定义了一系列行或列
- rows/columns 的值规定了每行或每列占据屏幕的面积

21.1 Frame标签

Frame 标签定义了放置在每个框架中的 HTML 文档
假如一个框架有可见边框，用户可以拖动边框来改变它的大小。为了避免这种情况发生，可以在 $< frame >$ 标签中加入：noresize=”noresize”。

22 iframe

语法 $< iframe src = "URL" >< /iframe >$

22.1 iframe 高度和宽度

height 和 width 属性用于规定 iframe 的高度和宽度。
属性值的默认单位是像素，但也可以用百分比来设定（比如 "80%"）

22.2 删除边框

frameborder 属性规定是否显示 iframe 周围的边框。
设置属性值为 "0" 就可以移除边框

23 背景

应该使用层叠样式表（CSS）来定义 HTML 元素的布局和显示属性。

24 脚本

JavaScript 使 HTML 页面具有更强的动态和交互性

scrpt元素

- $< script >$ 标签用于定义客户端脚本，比如 JavaScript
- script 元素既可包含脚本语句，也可通过 src 属性指向外部脚本文件
- 必需的 type 属性规定脚本的 MIME 类型
- JavaScript 最常用于图片操作、表单验证以及内容动态更新
- $< noscript >$ 标签提供无法使用脚本时的替代内容，比方在浏览器禁用脚本时，或浏览器不支持客户端脚本时。
  noscript 元素可包含普通 HTML 页面的 body 元素中能够找到的所有元素。
  只有在浏览器不支持脚本或者禁用脚本时，才会显示 noscript 元素中的内容

24.1 对付老式浏览器

如果浏览器压根没法识别 < *script* > 标签，那么 < *script* > 标签所包含的内容将以文本方式显示在页面上。为了避免这种情况发生，你应该将脚本隐藏在注释标签当中。那些老的浏览器（无法识别 < *script* > 标签的浏览器）将忽略这些注释，所以不会将标签的内容显示到页面上。而那些新的浏览器将读懂这些脚本并执行它们，即使代码被嵌套在注释标签内。

25 头部元素

< *head* >**元素** < *head* > 元素是所有头部元素的容器。< *head* > 内的元素可包含脚本，指示浏览器在何处可以找到样式表，提供元信息，等等。
以下标签都可以添加到 head 部分：< *title* >、< *base* >、< *link* >、< *meta* >、< *script* > 以及 < *style* >

title

- < *title* > 标签定义文档的标题
- title 元素在所有 HTML/XHTML 文档中都是必需的
- 定义浏览器工具栏中的标题
- 提供页面被添加到收藏夹时显示的标题
- 显示在搜索引擎结果中的页面标题
-
-
-
-

< *base* >**元素** 标签为页面上的所有链接规定默认地址或默认目标（target）

< *link* >**元素** 标签定义文档与外部资源之间的关系（标签最常用于连接样式表）

< *style* >**元素** 标签用于为 HTML 文档定义样式信息。

$< meta >$**元素** 元数据（metadata）是关于数据的信息。标签提供关于 HTML 文档的元数据。元数据不会显示在页面上，但是对于机器是可读的。
典型的情况是，meta 元素被用于规定页面的描述、关键词、文档的作者、最后修改时间以及其他元数据。$< meta >$ 标签始终位于 head 元素中。元数据可用于浏览器（如何显示内容或重新加载页面），搜索引擎（关键词），或其他 web 服务

针对搜索引擎关键词 一些搜索引擎会利用 meta 元素的 name 和 content 属性来索引您的页面

26 字符实体

中的预留字符必须被替换为字符实体

实体 如果希望正确地显示预留字符，我们必须在 HTML 源代码中使用字符实体（character entities）&entity_name;&#entity_number;

不间断空格 HTML 中的常用字符实体是不间断空格( )浏览器总是会截短 HTML 页面中的空格。如果您在文本中写 10 个空格，在显示该页面之前，浏览器会删除它们中的 9 个。如需在页面中增加空格的数量，您需要使用   字符实体

27 URL

- scheme - 定义因特网服务的类型。最常见的类型是 http
- host - 定义域主机（http 的默认主机是 www）
- domain - 定义因特网域名，比如 w3school.com.cn
- :port - 定义主机上的端口号（http 的默认端口号是 80）
- path - 定义服务器上的路径（如果省略，则文档必须位于网站的根目录中）。
- filename - 定义文档/资源的名称

（经试验，urlencode按照网页要求，一般是gbk编码或utf-8编码，然后每个非ascii码要添加百分号前缀。）

28 URL字符编码

URL 编码会将字符转换为可通过因特网传输的格式

URL编码 URL 只能使用 ASCII 字符集来通过因特网进行发送。由于 URL 常常会包含 ASCII 集合之外的字符，URL 必须转换为有效的 ASCII 格式。URL 编码使用 "%" 其后跟随两位的十六进制数来替换非 ASCII 字符。URL 不能包含空格。URL 编码通常使用 + 来替换空格。

29 Web Server

如果希望向世界发布您的网站，那么您必须把它存放在 web 服务器上

- 硬件支出
- 软件支出
- 人工费
- 使用因特网服务提供商（ISP）

30 颜色

颜色由一个十六进制符号来定义，这个符号由红色、绿色和蓝色的值组成（RGB）。
每种颜色的最小值是0（十六进制：#00）。最大值是255（十六进制：#FF）。

31 $<!DOCTYPE>$

$<!DOCTYPE>$ 不是 HTML 标签。它为浏览器提供一项信息（声明），即 HTML 是用什么版本编写的。

HTML5 $<!DOCTYPEhtml>$

32 HTML表单

HTML 表单用于搜集不同类型的用户输入

32.1 $< form >$元素

元素定义 HTML 表单。HTML 表单包含表单元素。

$< input >$元素 最重要的表单元素。元素有很多形态，根据不同的 type 属性

- text:定义常规文本输入
- radio:定义单选按钮输入（选择多个选择之一）
- submit:定义提交按钮（提交表单）

32.2 Action属性

action 属性定义在提交表单时执行的动作

32.3 Method属性

method 属性规定在提交表单时所用的 HTTP 方法（GET 或 POST）

32.4 $< select >$元素

元素定义下拉列表

32.5 $< option >$元素

列表通常会把首个选项显示为被选选项。您能够通过添加 selected 属性来定义预定义选项。

32.6 $< textarea >$元素

元素定义多行输入字段（文本域）

32.7 $< button >$

定义可点击的按钮

32.8 HTML5表单元素

- $<datalist>$ 为 $<input>$ 元素规定预定义选项列表。$<input>$ 元素的 list 属性必须引用 ¡datalist¿ 元素的 id 属性。
- $<keygen>$
- $<output>$

33 HTML输入类型

- text
- password
- submit 定义提交表单数据至表单处理程序的按钮。表单处理程序（form-handler）通常是包含处理输入数据的脚本的服务器页面。在表单的 action 属性中规定表单处理程序（form-handler）
- radio
- checkbox 定义复选框
- button
- number 包括**输入限制**
- date 用于应该包含日期的输入字段
- color
- range
- month
- week
- time 用户选择时间（无时区）
- datetime 用户选择日期和时间（有时区）
- datetime-local 用户选择日期和时间（无时区）
- email

- search
- tel
- url

34 HTML Input属性

34.1 value属性

规定输入字段的初始值

34.2 readonly 属性

规定输入字段为只读（不能修改）

34.3 disabled 属性

规定输入字段是禁用的。被禁用的元素不会被提交。

34.4 size 属性

属性规定输入字段的尺寸（以字符计）

34.5 maxlength 属性

属性规定输入字段允许的最大长度

34.6 autocomplete属性

属性规定表单或输入字段是否应该自动完成。autocomplete 属性适用于 $< form >$ 以及如下 $< input >$ 类型：text、search、url、tel、email、password、datepickers、range 以及 color

34.7 novalidate 属性

novalidate 属性属于 ¡form¿ 属性。如果设置，则 novalidate 规定在提交表单时不对表单数据进行验证

34.8 autofocus 属性

autofocus 属性是布尔属性。如果设置，则规定当页面加载时 ¡input¿ 元素应该自动获得焦点

34.9 form 属性

属性规定< *input* >元素所属的一个或多个表单。输入字段位于 HTML 表单之外（但仍属表单）

34.10 formaction 属性

formaction 属性规定当提交表单时处理该输入控件的文件的 URL。
覆盖 ¡form¿ 元素的 action 属性。
适用于 type=”submit” 以及 type=”image”

34.11 formenctype 属性

属性规定当把表单数据（form-data）提交至服务器时如何对其进行编码（仅针对 method=”post” 的表单。覆盖 ¡form¿ 元素的 enctype 属性。适用于 type=”submit” 以及 type=”image”

34.12 formmethod 属性

34.13 formnovalidate 属性

34.14 formtarget 属性

34.15 height和width属性

height 和 width 属性规定 ¡input¿ 元素的高度和宽度。仅用于 ¡input type=”image”¿

34.16 list属性

list 属性引用的 ¡datalist¿ 元素中包含了 ¡input¿ 元素的预定义选项。

34.17 min和max属性

规定 ¡input¿ 元素的最小值和最大值。适用于如需输入类型：number、range、date、datetime、datetime-local、month、time 以及 week。

34.18 multiple属性

如果设置，则规定允许用户在 ¡input¿ 元素中输入一个以上的值。multiple 属性适用于以下输入类型：email 和 file

34.19 pattern属性

规定用于检查 ¡input¿ 元素值的正则表达式。适用于以下输入类型：text、search、url、tel、email、and password

34.20 placeholder 属性

规定用以描述输入字段预期值的提示（样本值或有关格式的简短描述）。该提示会在用户输入值之前显示在输入字段中。placeholder 属性适用于以下输入类型：text、search、url、tel、email 以及 password

34.21 required 属性

required 属性是布尔属性。如果设置，则规定在提交表单之前必须填写输入字段

34.22 step 属性

规定 ¡input¿ 元素的合法数字间隔

35 HTML5 Related

35.1 figure figcaption元素

通过 HTML5，图片和标题能够被组合在 $< figure >$元素中。$< img >$ 元素定义图像，$< figcaption >$ 元素定义标题

35.2 Canvas

canvas 元素用于在网页上绘制图形

什么是Canvas？ canvas 元素使用 JavaScript 在网页上绘制图像。画布是一个矩形区域，您可以控制其每一像素。canvas 拥有多种绘制路径、矩形、圆形、字符以及添加图像的方法

Canvas创建 $< canvasid = "myCanvas" width = "200" height = "100" >< /canvas >$

通过JS来绘制 canvas 元素本身是没有绘图能力的。所有的绘制工作必须在 JavaScript 内部完成

36 内联SVG

HTML5支持内联SVG

36.1 SVG

- 可伸缩矢量图形 (Scalable Vector Graphics)
- 定义用于网络的基于矢量的图形
- 使用 XML 格式定义图形
- 在放大或改变尺寸的情况下其图形质量不会有损失
- 万维网联盟的标准
- 优势：
- SVG 图像可通过文本编辑器来创建和修改
- SVG 图像可被搜索、索引、脚本化或压缩
- 是可伸缩的
- 可在任何的分辨率下被高质量地打印
- 可在图像质量不下降的情况下被放大

37 Canvas vs. SVG

37.1 SVG

- SVG 是一种使用 XML 描述 2D 图形的语言
- SVG 基于 XML，这意味着 SVG DOM 中的每个元素都是可用的。您可以为某个元素附加 JavaScript 事件处理器

- 在 SVG 中，每个被绘制的图形均被视为对象。如果 SVG 对象的属性发生变化，那么浏览器能够自动重现图形

37.2 Canvas

- Canvas 通过 JavaScript 来绘制 2D 图形
- Canvas 是逐像素进行渲染的
- 在 canvas 中，一旦图形被绘制完成，它就不会继续得到浏览器的关注。如果其位置发生变化，那么整个场景也需要重新绘制，包括任何或许已被图形覆盖的对象

37.3 比较

Canvas

- 依赖分辨率
- 不支持事件处理器
- 弱的文本渲染能力
- 能够以 .png 或 .jpg 格式保存结果图像
- 最适合图像密集型的游戏，其中的许多对象会被频繁重绘

SVG

- 不依赖分辨率
- 支持事件处理器
- 最适合带有大型渲染区域的应用程序（比如谷歌地图）
- 复杂度高会减慢渲染速度（任何过度使用 DOM 的应用都不快）
- 不适合游戏应用

38 多媒体

Web 上的多媒体指的是音效、音乐、视频和动画。现代网络浏览器已支持很多多媒体格式

WAVE 是因特网上最受欢迎的无压缩声音格式，所有流行的浏览器都支持它。如果您需要未经压缩的声音（音乐或演讲），那么您应该使用 WAVE 格式

MP3 是最新的压缩录制音乐格式。MP3 这个术语已经成为数字音乐的代名词。如果您的网址从事录制音乐，那么 MP3 是一个选项。