

# python 前端 html

## 阅读目录

- [body 常用的标签](#)

**web 服务本质:** 浏览器发出请求--HTTP 协议--服务端接收信息----服务端返回响应---服务端把 HTML 文件发给浏览器--浏览器渲染页面.

**HTML:** 超文本标记语言是一种用于创建网页的标记语言.

**超文本:**就是指页面内可以包含图片、[链接](#), 甚至音乐、[程序](#)等非文字元素

**本质**是浏览器可识别的规则,我们按照规则写网页,浏览器根据我们写的代码渲染网页,对于不同的浏览器对于同一个标签渲染不同(兼容性问题),网页文件的扩展名: .HTML.

**HTML** 是一种标记语言而不是编程语言.

**标记语言**是一种将文本以及文本相关的其他信息结合起来, 展现出关于文档结构和数据处理细节的电脑文字编码

HTML 使用标签来描述网页.

基本的 HTML 文档

```
<!DOCTYPE html>
<html lang="zh-CN"> #lang 选择的语言
<head>
  <meta charset="UTF-8">
  <title>css 样式优先级</title>
</head>
<body>

</body>
</html>
```

解释:

<!DOCTYPE html> :用来声明为 HTML 文档, 声明必须是 HTML 文档的第一行, 位于 <html> 标签之前.

<html>, <\html>:是文档开始和结束的标志. 在他们之间是文件的头部和身体

<head>, </head> 是文档头部开始和结束的标志, 它们之间的内容不会在网页中显示, 里边必须包含标题 title

<meta>:提供了 HTML 的元数据, 元数据不会显示在客户端, 但是会被浏览器解析,

元数据 META 元素通常用于指定网页的描述, 关键词, 文件的最后修改时间, 作者及其他元数据。

<title> </title> 定义了网页的标题在浏览器标题栏中显示.

<body> </body> 之间的内容是网页可见的内容

## HTML 标签

HTML 标签是有尖括号包括着的关键字

HTML 标签的样式: 1. 成对出现的 <html>, </html>

2. 单独出现的 <br/>

标签里面可以有若干的属性, 也可以没有属性

标签的语法:

- <标签名 属性 1= "属性值 1" 属性 2= "属性值 2" .....>内容部分</标签名>
- <标签名 属性 1= "属性值 1" 属性 2= "属性值 2" ..... />

几个很重要的属性:

- id: 定义标签的唯一 ID, HTML 文档树中唯一
- class: 为 html 元素定义一个或多个类名 (classname) (CSS 样式类名)
- style: 规定元素的行内样式 (CSS 样式)

## HTML 注释 <!--注释内容-->

### head 常用的标签

meta 的属性有两种:

1. *http-equiv 属性*: 正规名字叫网页重定向 相当于 http 的文件头作用, 它可以向浏览器传回一些有用的信息, 以帮助正确地显示网页内容, 与之对应的属性值为 *content*, *content* 中的内容其实就是各个参数的变量值。

<meta http-equiv="refresh" content="5; https://www.baidu.com"> 页面 5 秒后连接到百度网页

2. *name 属性*: 主要用于描述网页, 与之对应的属性值为 content, content 中的内容主要是便于搜索引擎机器人查找信息和分类信息用的。

1、<meta name="Generator" content="">用以说明生成工具 (如Microsoft FrontPage 4.0) 等;

2、<meta name="KEYWords" content="">向搜索引擎说明你的网页的关键词;

3、<meta name="DEscription" content="">告诉搜索引擎你的站点的主要内容;

4、<meta name="Author" content="你的姓名">告诉搜索引擎你的站点的制作的作者

非 meta 标签

1. <title>oldboy</title> 标题

2. <link rel="icon" href="favicon.ico"> # link 连接到外部样式文件 icon 图标
3. <link rel="stylesheet" href="mystyle.css"> #样式文件
4. <script src="myscript.js"></script> js 文件

## body 常用的标签

### 块级标签(独占一行)

块级元素特点:

1. 独占一行, 默认宽度占 100%
2. 可以直接定义宽高。
3. 块级元素可以设置 margin 和 padding 的四个方向
4. 块级元素可以包含行内元素和块级元素
5. 嵌套(包含下),子块标签宽度在没有定义的情况下和父块标签宽度相同。

**常见的块级元素有:**

#### 1.h1~h6 是用来定义标题的重要程度

<h1>标题 1</h1> 页面中的大标题, 只能出现一次在一个页面, 重要程度最高

<h2>标题 2</h2>

<h3>标题 3</h3>

<h4>标题 4</h4>

<h5>标题 5</h5>

<h6>标题 6</h6>重要程度最低

2.<hr> 在网页中加入一条横线

3.<p>段落标签</p>: 会上下生成空白行而 div 标签就没有

4.<div> </div>:div 标签用来定义一个块级元素, 并无实际的意义。主要通过 CSS 样式为其赋予不同的表现。

5.<ul> </ul> 无序列表

6 <li> </li> 无序列表中的内容

### 行内标签(长度根据内容确定)

特点:

1. 多个内联标签的内容在一行显示.
2. 宽度根据内容确定。
3. 不可以设置宽高。
4. 行内元素只可以设置 margin 和 padding 的左右两个方向, 上下不起作用。

- 5. 行内元素 display 默认为 inline
- 6. 行内元素一般只包含行内元素和文本。可以嵌套。

常见的行内标签有：

1. a :超链接标签所谓的超链接是指从一个网页指向一个目标的连接关系，这个目标可以是另一个网页，也可以是相同网页上的不同位置，还可以是一个图片，一个电子邮件地址，一个文件，甚至是一个应用程序。

```
<a href="http://www.oldboyedu.com" target="_blank" >点我</a>
```

href 属性指定目标网页地址。该地址可以有几种类型：

- 绝对 URL - 指向另一个站点（比如 href="http://www.jd.com"）
- 相对 URL - 指当前站点中确切的路径（href="index.htm"）
- 锚 URL - 指向页面中的锚（href="#top"） href 中可以跟 name 的值或 id 的值，但是 top 前边必须有#，因为这是锚点的标志

target:

- \_blank 表示在新标签页中打开目标网页
- \_self 表示在当前标签页中打开目标网页

2.<b>加粗</b> 和 <strong>加粗</strong> 效果一样但是 strong 更有语义,推荐 strong

3.图片标签<img>

```

```

4.<u>下划线</u> 和 .<ins>删除线</ins> 效果相同

5.<s>删除</s> 和 .<del>删除线</del> 效果相同

6.<i>斜体</i> 和 <em>斜体</em> 效果一样 em 更有语气强调,推荐用 em

7.span 标签用来定义内联行(行内元素)，并无实际的意义。主要通过 CSS 样式为其赋予不同的表现。

标签嵌套：

关于标签嵌套：通常块级元素可以包含内联元素或某些块级元素，但内联元素不能包含块级元素，它只能包含其它内联元素。

p 标签不能包含块级标签。

特殊字符

内容	对应代码
空格	&nbsp;sp

>	&gt;
<	&lt;
&	&amp;
¥	&yen;
版权	&copy;
注册	&reg;

## 列表

### 1.无序列表

ul 只能容有 li,li 里边可以有任意标签

```
<ul type="disc">
  <li>第一章</li>
  <li>第二章</li>
  <li>第三章</li>
  <li>第四章</li>
</ul>
```

结果:

- 第一章
- 第二章
- 第三章
- 第四章

type 属性:

- disc (实心圆点, 默认值)
- circle (空心圆圈)
- square (实心方块)
- none (无样式) /\*这个用 css 可以设置为: List-style-type:none\*/

## 2.有序列表

```
<ol type="1" start="2">  
  <li>第一章</li>  
  <li>第二章</li>  
  <li>第三章</li>  
  <li>第四章</li>  
</ol>
```

结果:

2. 第一章
3. 第二章
4. 第三章
5. 第四章

type 属性:

- 1 数字列表, 默认值
- A 大写字母
- a 小写字母
- I 大写罗马
- i 小写罗马

## 3 标题列表

```
<dl>
  <dt>标题 1</dt>
  <dd>内容 1</dd>
  <dt>标题 2</dt>
  <dd>内容 1</dd>
  <dd>内容 2</dd>
</dl>
```

结果:

```
标题1
  内容1
标题2
  内容1
  内容2
```

## 表格(table)

表格是一个二维数据空间，一个表格由若干行组成，一个行又有若干单元格组成，单元格里可以包含文字、列表、图案、表单、数字符号、预置文本和其它的表格等内容。

表格最重要的目的是显示表格类数据。表格类数据是指最适合组织为表格格式（即按行和列组织）的数据。

```
<body>
<table border="1" cellspacing="5" width="300px">
<caption> 表格题目 </caption>
  <thead>
  <tr>                                <!--行-->
    <th>序号</th>  <!--表格的表头使用 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-->
    <th>姓名</th>
```

```

    <th>爱好</th>
</tr>
</thead>
<tbody>
<tr>
    <td>1</td> <!--数据单元格 tabledata 的内容-->
    <td>egon</td>
    <td>杠娘</td>
</tr>
<tr>
<td>2</td>
<td>元昊</td>
<td>日天</td>
</tr>
</tbody>

```

- <thead> 标签定义表格的表头。该标签用于组合 HTML 表格的表头内容
- <tr> 行标签 table row 的缩写
- <th> 表格的表头的内容使用 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** table head 的缩写
- <td> 数据单元格(tabledata)的内容
- <caption> 表格题目 </caption>

属性:

- border: 表格边框的粗细.
- bordercolor 表格边框颜色
- cellpadding: 内边距
- cellspacing: 外边距.
- width: 宽度 :里边可以是(像素 百分比 等)。(最好通过 css 来设置长宽)
- rowspan: 单元格竖跨多少行
- colspan = "2": 单元格横跨多少列(即合并单元格), 两个单元格合并
- align="center| left|right" 文字居中, 左对齐, 右对其, **1 这项写在 table 中就是表格居中, 2 这个写在 tr 中就是表格中的文字居中**
- bgcolor 给表格配背景色
- valign = "center| top|bottom" 垂直对齐



结果：

序号	姓名	爱好
1	egon	杠娘
2	元昊	日天

## 表单(form)

作用:收集信息

功能：

- 表单用于向服务器传输数据，从而实现用户与 Web 服务器的交互
- 表单能够包含 input 系列标签，比如文本字段、复选框、单选框、提交按钮等等。
- 表单还可以包含 textarea、select、fieldset 和 label 标签。

表单的组成:

### 1 提示信息

昵称

密码

确认密码

性别

生日

所在地

手机号码

### 2 表单控件



### 3.表单域:表单输入的区域

#### form 元素的属性:

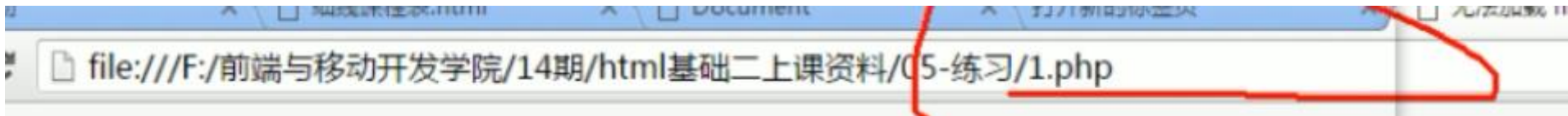
- action: url 用于指定表单的发送地址(服务器地址)
- method: 表单数据发送到服务器的方法 常用的方法有两种: post 和 get.
- 这里也可以添加 target="\_blank" 属性
- enctype : 表单发送到服务器的的编码格式

#### get 和 post 的区别:

- get 方法提交 , 通过地址栏来传递信息,数据会附在网址之后主动提交给服务器. get 是从服务器上获取数据, form 表单默认提交方式是 get .确定安全性差,用户名和密码会附在网址后面.



- post 方法提交:数据不会附在网址之后,会将数据打包发给服务器,等候服务器读取数据来进行处理.post 是向服务器传送数据。以后学数据库时会用到. 安全性高,网址后不会附你的用户名和密码



- **enctype 属性:**

熟悉表单元素<form>的小伙伴，对其中的属性 enctype 一定不会陌生，就是它规定了对表单提交给服务器时表单数据编码的内容类型(Content Type)。

知道了表单编码由 enctype 决定的，那么它究竟有多少可选的取值呢？是不是所有的 MIME 类型它都能用呢？

实际上，根据 [HTML5 规范](#)中所叙述的，enctype 具有以下三种选项，其中最后一项 text/plain 是相比 4.01 新增的。

- application/x-www-form-urlencoded
- multipart/form-data
- text/plain

### application/x-www-form-urlencoded

这是**默认的编码类型**，使用该类型时，会将表单数据中非字母数字的字符转换成转义字符，如"%HH"，然后组合成这种形式 key1=value1&key2=value2；所以后端在取数据后，要进行解码。

**注意:**

- 若表单中有文件，则只留文件名；

method 为 get 时候，浏览器用 x-www-form-urlencoded 的编码方式把 form 数据转换成一个字串 (name1=value1&name2=value2...)，然后把这个字串 append 到 url 后面，用?分割，加载这个新的 url。

### multipart/form-data

当 method 为 post 时候，浏览器把 form 数据封装到 http body 中，然后发送到 server。如果没有 type=file 的控件，用默认的 application/x-www-form-urlencoded 就可以了。但是如果有 type=file 的话，就要用到 multipart/form-data 了。浏览器会把整个表单以控件为单位分割，并为每个部分加上 Content-Disposition(form-data 或者 file),Content-Type(默认为 text/plain),name(控件 name)等信息，并加上分割符(boundary)。

- 一般来说，method 和 enctype 是两个不同的互不影响的属性，但在传文件时，method 必须要指定为 POST，否则文件只剩下 filename 了；
- 当没有传文件时，enctype 会改回默认的 application/x-www-form-urlencoded。

**text/plain:** 表单数据中的空格转换为 "+" 加号，但不特殊字符编码。(get 方式会这样，post 时不会)

## input :

- 用来设置表单中的内容项,比如表单中的文本输入
- 还可以在表单外使用

## input 元素的属性:

1. type 指定输入内容的类型,默认为 text 单行文本框.
2. name:输入内容的识别名称,传递参数时的参数名称,在服务器中对应着 value,相当于 value 的 key, 用来区分不同的文本框.
3. value: 表单提交时对应项传的值
  - type="button", "reset", "submit"时, 为按钮上显示的文本内容
  - type="text","password","hidden"时, 为输入框的默认值意思是不用输入就会显示
  - type="checkbox","radio", "file", 为输入相关联的值.
- 4.maxlength:设置最大字数的限制,汉字字符数字相同.
- 5.checked=" checked" 设置默认选中项,针对单选框,多选框有用.
6. selected="selected" : 设置下拉选中项,针对 select 有用
- 7 multiple='multiple': 设置下拉多选项

’,’,

### <1> 表单类型

type:            text 文本输入框  
                 password 密码输入框  
                 radio 单选框 只有将 name 值设置相同的时候才会出现单选效果, 否则会出现多选效果  
                 checkbox 多选框  
                 submit 提交按钮  
                 button 按钮(需要配合 js 使用.) button 和 submit 的区别?  
                 file 提交文件: form 表单需要加上属性 enctype="multipart/form-data"

上        传文件注意两点:

            请求方式必须是 post  
            enctype="multipart/form-data"

### <2> 表单属性

name:        表单提交项的键.

            注意和 id 属性的区别: name 属性是和服务器通信时使用的名称;

            而 id 属性是浏览器端使用的名称, 该属性主要是为了方便客户端编程, 而在 css 和 javascript 中使用的

value:        表单提交项的值. 对于不同的输入类型, value 属性的用法也不同:

            type="button", "reset", "submit" - 定义按钮上的显示的文本

type="text", "password", "hidden" - 定义输入字段的初始值

type="checkbox", "radio", "image" - 定义与输入相关联的值

checked: radio 和 checkbox 默认被选中

readonly: 只读. text 和 password

disabled: 对所用 input 都好使. 输入框内的信息不能输入时灰色  
,,,

## select 标签:下拉列表

### 下拉列表语法

```
<select>  
  
<option> 下拉列表选项 </option>  
  
<option> 下拉列表选项 </option>  
  
</select>
```

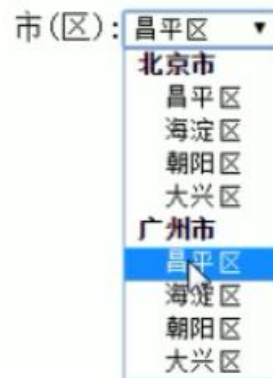
下拉分组:

代码:

```
市(区):<select>
    <optgroup label="北京市">
    <option>昌平区</option>
    <option>海淀区</option>
    <option>朝阳区</option>
    <option>大兴区</option>
    </optgroup>
    <optgroup label="广州市">
    <option>昌平区</option>
    <option>海淀区</option>
    <option>朝阳区</option>
    <option>大兴区</option>
    </optgroup>
</select>
```

Label:分组名称

效果:



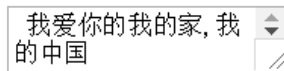
```
<select> 下拉选标签属性
name: 表单提交项的键.
size: 选项个数
multiple: multiple
    <optgroup>为每一项加上分组
    <option> 下拉选中的每一项 属性:
        value: 表单提交项的值.
        selected: selected 下拉选默认被选中
'''
```

## <textarea> 多行文本框标签

```
'''
<form id="form1" name="form1" method="post" action="">
    <textarea cols= "控制输入每行的字符" rows= "控制输入的行数" name= "名称">
        默认内容
    </textarea>
</form>

'''*
```

## 效果

A screenshot of a web browser window showing a text area. The text area contains the text "我爱你的我的家, 我的中国". The text area has a vertical scrollbar on the right side.

## 文件上传控件

```
<input type="file" name="file">
```

效果:

A screenshot of a web browser window showing a file upload button. The button is labeled "选择文件" (Select File). To the right of the button, the filename "xdd1.jpeg" is displayed.

## <label> 标签

定义：<label> 标签为 input 元素定义标注（标记）。如果想点击框内的内容或按钮,直接点击 label 标签的内容就可以了

说明：

1 label 元素不会向用户呈现任何特殊效果。

2 <label> 标签的 for 属性值应当与相关元素的 id 属性值相同。

```
'''  
<form method="post" action="">  
    <label for="username">用户名</label>  
    <input type="text" name="username" id="username" size="20" />  
</form>  
'''
```

效果：

用户名

## 按钮系列控件

提交按钮 :<input type="submit">:

普通按钮: 不能实现提交功能, 配合 JS 使用

```
<input type="button" value="普通按钮">
```

效果：

普通按钮

**图片按钮:** 可以是实现提交功能

```
<input type="image" src="p1.jpg">
```

效果：





重置按钮: 把内容还原到默认状态

```
<input type="reset" >
```

## iframe (inner Frame) 内联框架

iframe 元素会创建包含另外一个文档的内联框架（即行内框架）。iframe 是属于内联框架，它是 body 的子集，和 body 是父子关系，iframe 作为一个普通元素放在 body 里。说白了就是在当前页面中插入另一个页面。

属性：

1. width 可设置内联框架的宽。
2. height 可设置内联框架的高。
3. name 可设置框架的名称。
4. src 可设置页面的路径
5. scrolling 规定是否在 iframe 中显示滚动条 (yes,no,auto)
6. frameborder 规定是否显示框架周边的边框 (1 默认有边框, 0 无边框)

代码：

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge"> <!--IE 浏览器最高渲染-->
  <meta name="viewport" content="width=device-width, initial-scale=1"> <!--为了确保适当的绘制和缩放-->
  <title>iframe 框架的使用</title>
  <link rel="stylesheet" href="../bootstrap-3.3.7-dist/css/bootstrap.min.css">
</head>
<body>
<address>
```

北京海淀区。西三旗桥东。神州科技园

```
</address>
<hr>
<iframe src="https://www.baidu.com/" name="topifame" frameborder="0" width=100% height="300px"></iframe>
<hr>
<a href="https://www.baidu.com/" target="topifame">百度</a>
<a href="https://www.sina.com.cn/" target="topifame">新浪</a>
</body>
</html>
```

效果图



当你点击新浪时，iframe 的框架内会显示新浪的网页，反之会显示百度的页面