

python前端

笔记本: zymapple的笔记本
创建时间: 2018/12/10 2:32
作者: 519799178@qq.com
URL: about:blank

更新时间: 2018/12/14 1:13

1. HTML	(常用标签	网页的基本结构)
2. CSS	(常用样式	网页的显示效果)
3. JavaScript	(用户交互效果	动态效果)
4. jQuery	(JavaScript库	简化原生js操作)
5. Ajax	(异步数据交互	在页面不刷新的情况下进行数据交互)
6. Bootstrap	(前端UI框架	快速搭建静态页面并支持不同设备)

HTML概述

一、什么是HTML

HTML 是用来描述网页的一种语言。



超文本标记语言（英语：HyperText Markup Language，简称：HTML）是一种用于创建网页的标准标记语言。

HTML 不是一种编程语言，而是一种标记语言

标记语言是一套标记标签 (markup tag)

HTML 文档包含了HTML 标签及文本内容

HTML文档也叫做 web 页面

二，HTML是做什么的

可以使用HTML来建立自己的WEB站点，HTML运行在浏览器上，由浏览器来解析

三，建立HTML文件

- .html
- .htm

四，HTML注释

<!--注释内容-->

五，HTML网页的基本结构

<!DOCTYPE html>	声明为HTML 5 文档	
<html> </html>	元素是HTML页面的根	元素
<head> </head>	元素包含了文档的元	(meta) 数据
<body> </body>	元素包含了可见的页面	内容

六， HTML标签结构

HTML标记标签通常被称为 HTML标签（HTML tag）

- HTML标签是由尖括号包围的关键词
- HTML标签通常是成对出现的
- 标签对 中的第一个标签是开始标签，第二个标签是结束标签

<开始标签>内容</结束标签>

七， HTML元素

‘HTML标签’ 和 ‘HTML元素’ 通常都是描述同样的意思，一个HTML元素包含了开始标签与结束标签。

八， HTML属性

- HTML元素可以设置属性
- 属性一般添加在开始标签
- 属性一般以名称/值 对的形式出现，比如：name = " value "

注意：

- 属性值必须用双引号引起来
- 标签都要用小写字母
- 双标签必须要写闭合标签

HTML常用标签

一， HTML常用的块级标签（块级元素）

- 块级标签 ----> 独占一行

- 有语义的HTML块级标签 ----> 有默认样式
- 标题<Heading> ----> 通过H1~H6标签来定义
- 段落 ----> 通过标签p来定义
- 列表
 - 无序列表 ul, li
 - 是一个项目的列表, 列项目使用粗体圆点 (典型的小黑圆圈) 进行标记
 - 有序列表 ol, li (ol有 type= " 属性)
 - 也是一个项目的列表, 列表项目使用数字进行标记
 - 自定义列表 dl, dt, dd
 - dl 是定义一个列表项
 - dt 是定义标题
 - dd 是定义这个列表的解释项

-表格 table, tr (行), td (列), th (表头)

- table常用属性
 - border 边框
 - cellpadding 内容距离表框的距离
 - cellspacing 单元格和单元格之间的距离
 - rowspan 垂直合并 (跨行显示)
 - colspan 水平合并 (跨列显示)
 - align 内容水平对齐方式
 - 如果给tr设置, 会影响整行, 给td设置, 只影响单元格
 - valign 内容垂直对齐方式
 - 如果给tr设置, 会影响整行, 给td设置, 只影响单元格
 - width 设置宽度
 - height 设置高度

-无异议的块级元素 div

<div>元素没有特定的含义。除此之外, 由于它属于块级元素, 浏览器会在其前后显示换行。一般与css一同使用

<div>元素的另一个常见的用途是文档布局。他取代了使用表格定义布局的老式方法

--- <!--水平分割线-->

--- <hr>

web_1.html

<!DOCTYPE html>

```
<html lang="en">
<head>
<meta charset="UTF-8">
<title>常用HTML块级标签</title>
</head>
<body>
<!--常用的有语义的块级元素 有默认样式 独占一行-->
这是一个标题
<h1>这是一级标签</h1>
<h2>这是二级标题</h2>
<h3>这是三级标题</h3>
<h6>这是六级标签</h6>
<!--段落标签 有默认样式 独占一行-->
<!--<br>标签 强制换行-->
<p>段落<br>标签</p>
<p>第二个段落标签</p>
<!--列表-->
<!--无序列表 有默认样式 独占一行-->
<ul>
<li>列表项一</li>
<li>列表项二</li>
<li>列表项三</li>
</ul>
<!--有序列表 ol li 有自己的默认样式 独占一行-->
<!--有序列表当中有type属性 设置列表的标记类型 start 设置
列表项从第几个标记开始-->
<ol type="1" start="3">
<li>列表一</li>
<li>列表二</li>
<li>列表三</li>
</ol>
<!--自定义列表 有默认样式 独占一行-->
<dl>
<dt>这是自定义列表的标题</dt>
<dd>这是自定义列表</dd>
<dd>这是第二行的解释项</dd>
</dl>
<!--表格标签 table声明一个表格 th表头 tr行 td列-->
<!--声明一个表格-->
<!--
table 的属性: borde
cellpadding
。。。等
-->
<table border="1" cellpadding="10px">
<!--这是表格当中的行-->
<tr>
<!--这是表格当中的列-->
<td colspan="4">这是第一行中的第一列</td>
<!--<td>这是第一行中的第二列</td>-->
<!--<td>这是第一行中的第三列</td>-->
<!--<td>这是第一行中的第四列</td>-->
</tr>
<tr>
<td rowspan="2">这是第二行的第一列</td>
<td>这是第二行的第二列</td>
<td>这是第二行的第三列</td>
<td>这是第二行的第四列</td>
</tr>
<tr>
<td>这是第三行的第一列</td>
<td>这是第三行的第二列</td>
<td>这是第三行的第三列</td>
<td>这是第三行的第四列</td>
</tr>
</table>
<!--水平分割线-->
<hr>
```

<!-- 块元素 没有默认样式（没有默认的语义）div 独占一行 一般用来划定一块区域， 配合css在布局当中使用 -->

<div>这是一个div</div>

</body>

</html>

< == <

> == >

-
- 网易 #中间这个‘网易’也可以换成其他的嵌套其他的标签，比如嵌套上面的图片标签，点击图片后就会跳转到‘www.163.com’

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>锚点</title>
5  </head>
6  <body>
7
8      <b id="top">图片一</b>
9      
10
11     <b id="top2">图片二</b>
12     
13
14     <b id="top3">图片三</b>
15     
16
17     <b id="top4">图片四</b>
18     
19
20     <a href="#top">跳转到图片一</a>
21     <a href="#top2">跳转到图片二</a>
22     <a href="#top3">跳转到图片三</a>
23     <a href="#top4">跳转到图片四</a>
24
25 </body>
26 </html>
```

-HTML常用的行级标签（行内元素）

- 不独占一行

-有语义的行内元素

- HTML链接 a标签
 - `链接文本`
 - `target="_blank"` 在新窗口打开跳转页面
- HTML图像
 - ``
 - `src`: 图片地址（网络地址或者 本地地址）
 - `alt`: 用于图片加载失败的替代文字（必须存在）
 - `title`: 鼠标悬停在图片上的提示信息
 - `width`: 设置宽px
 - `height`: 设置高px
 - 注意：一般我们只给图片设置宽度或者高度。另一个属性会等比例放大或者缩小
- 文本标签（不独占一行）
 - `b`标签 `i`标签 `strong`标签 `em`标签
 - `b`标签: 加粗
 - `i`标签: 倾斜
 - `strong`标签: 加粗 具有强调意义
 - `em`标签: 倾斜 具有强调意义

-无语义的行内元素

- `span`标签 配合css使用 来设置文本中一部分内容
 - `是的`前面两个是红色的字

-常用的实体字符

- `gt;` `lt;` `copy;` `nbsp`

-表单标签

- 表单是一个包含表单元素的区域，通过form来定义表单区域
 - `action`: 提交地址
 - `method`: 提交方式 `get` 与 `post`

表单提交默认是以get方式进行数据传输的
get: 通过url地址传输数据, 数据会显示在url地址栏当中, 传输数据有大小限制, 根据浏览器的不同数据大小不同, 相对来说不安全
post: 数据不通过url地址传参, 相对来说较安全, 数据传输没有具体的大小限制, 一般表单指定使用post方式提交数据

- enctype: 如果表单当中有文件上传, 必须添加这个属性。指定属性值为 " multipart/form-data " 用于文件转码。
- type: 指定我们的表单类型
- 通过type属性定义不同类型的表单控件
 - text 普通文本输入框
 - password 密码输入框
 - radio 单选按钮
 - checked 多选按钮
 - select 下拉框
 - file 文件上传选框
 - textarea 文本域
 - submit 提交按钮
 - button 提交按钮 <button>提交</button>
 - reset 重置按钮
 - hidden 隐藏域, 要和表单一起提交的信息
- 常用属性
 - name 属性: 表单项名, 用于储存内容值
 - value属性: 输入的值
 - disabled属性: 禁用属性
 - readonly属性: 禁用属性 设置为只读, 不能写入
 - checked属性: 选择框指定默认选项
 - placeholder: 提示
- 注意
 - form 有两个必须存在的属性 action提交地址 method提交方式
 - post: 通过request body传参, 参数不可见, 参数没有大小限制
 - get: 通过url进行传参, 参数可见, 不安全, 大小有限制
 - 表单控件一定要有name属性
 - 单选框 type= " radio " 必须有name属性和value属性, 并且一组单元框name属性的值必须相同
 - 多选框 type= " checkbox " 必须指定name属性和value属性, 并且一组name属性的值要相同
 - 下拉框必须配合option来使用

- value: 设置默认值
- hidden 不需要用户查看和填写的信息

```
<form action="" method="">
  <!--
    表单控件
      type:指定我们吃的表单类型(普通文本 密码框 单选框 多选框)
      必须要填写 name属性
    {username:1234, sex:0, love:[1, z, p]}-->
  用户名: <input type="text" name="username" value="1234"> <br>
  密 码: <input type="password" name="password"> <br>
  <!--单选框 type="radio" 必须有name属性和value属性, 并且一组单选框name属性的值必须相同-->
  性别: <input type="radio" name="sex" value="1"> 男
        <input type="radio" name="sex" value="0"> 女
  <br>
  <!--
    多选框 type="checkbox" 必须指定name属性和value属性, 并且一组多选框name属性的值要相同
  -->
  爱好: <br>
        <input type="checkbox" name="love[]" value="1"> 篮球 <br>
        <input type="checkbox" name="love[]" value="z"> 足球 <br>
        <input type="checkbox" name="love[]" value="p"> 乒乓球 <br>
</form>
```

-CSS

-CSS注释*-

- css中注释 /* 注释内容 */

-CSS基本语法及页面引用

- css基本语法
- css的定义方法是: 选择器 (属性: 值 属性: 值 属性: 值) 选择器是将样式和页面元素关联起来的名称, 属性时希望设置的样式属性每个属性有一个或者多个值

-css页面的引入方法

- 外链式: 通过link标签, 链接到外部样式表到页面中
 - <link rel="stylesheet" type="text/css" href="css/main.css">
- 嵌入式: 通过style标签, 在网页上创建嵌入的样式表
 - <style type="text/css">


```
div{width:10px; height:10px; color:red}
.....
</style>
```

- 内链式：通过标签的style属性，在标签上直接写样式
 - <div style="width:10; height:10px; color:red">

-注意：

- 1.外联式，通过link标签引入css文件 要在head标签当中引入
- 2.嵌入式，通过style标签来写css，也要写在head标签当中
- 3.内联式，通过style属性来设置css样式

- 三种方式同时使用时，有优先级问题，页面从上向下加载，离元素越近的优先级更高*

-CSS选择器

- 1.标签选择器
 - 标签选择器，此种选择器影响范围大，建议尽量应用在层级选择器中
 - *{margin:0;padding:0}

```
div{color:red}
<div>.....</div>  <!--对应以上两条样式-->
<div class="box">.....</div>  <!--对应以上
两条样式-->
```

- id选择器
 - 通过id名来选择元素，元素的id名称不能重复，所以一个样式设置项只能对应页面上一个元素，不能复用，id名一般给程序使用，所以不推荐使用id作为选择器
 - #box{color:red}

```
<div id="box">....</div>  <!--对应以上一条
样式，其他元素                    不允许应用此
                                样式-->
```

- 类选择器
 - 通过类名来选择元素，一个类可应用于多个元素，一个元素上也可以使用多个类，应用灵活，可复用，是css中应用最多的一种选择器
 - red{color:red}

```
big{font-size:20px}
nt10{nargin-top:10px}
<div class="red"> .....</div>
<h1 class="red big nt10">.....</h1>
<p class="red nt10">.....</p>
```

- 1.标签选择器 通过标签来设置元素的样式 影响范围最大
- 2.类选择器 通过class类名来设置元素 影响范围相对较小
- 3.id选择器 通过id属性的值来设置元素的样式，影响范围最小 注意id在html当中具有唯一性，不能重名
- 影响范围越大，优先级越低

-关系型选择器（可以同时给多个元素设置相同的样式）

- 层级选择器
 - 主要应用在选择父元素下的子元素，或者子元素下面的子元素，可与标签元素结合使用，减少命名，同时也可以通过层级，防止命名冲突
 - box span{color:red}

```
box.red{color:pink}
red{color:red}
<div class="box">
  <span>....</span>
  <a href="#" class="red">.....</a>
</div>
<h3 class="red">.....</h3>
```

- 组选择器
 - 多个选择器，如果有同样的样式设置，可以使用组选择器。也称为并列选择器

```
.box1,.box2,.box3{width:100px;height:100px}
.box1{background:red}
.box2{background:pink}
.box2{background:gold}
<div class="box1">....</div>
<div class="box2">....</div>
<div class="box3">....</div>
```

- 伪类及伪元素选择器
 - 常用的伪类选择器有hover，表示鼠标悬浮在元素上时的状态，伪元素选择器有before和after，他们可以通过样式在元素中插入内容
 - hover 设置鼠标悬停在元素上时的状态

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>伪类选择器</title>
  <style>
    /*hover 设置鼠标悬停在元素上时的状态*/
    .box{
      width:200px;
      height:200px;
      background-color:green;
    }
    .box:hover{
      /*鼠标悬停上之后的样式*/
      background-color:red;
      width:500px;
      height:500px;
    }

    /* after 在元素里面的尾部插入内容*/

  </style>
</head>
<body>
  <div class="box"></div>
```

- after 在元素尾部插入内容

- 层级选择器 可以和多种选择器混合使用

