

html和css

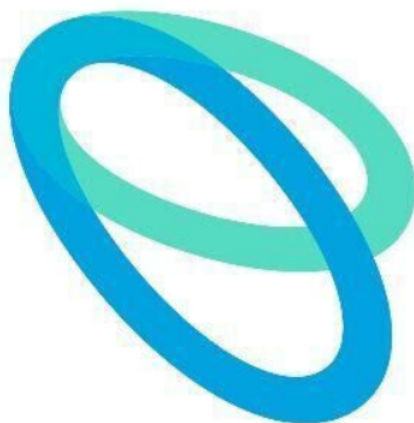
目标

- 学会 HTML 的基本知识
- 学会 CSS 的基本知识
- 可以独立完成课堂练习并做出最终练习 —— 个人名片

跟我做 - 第一张网页

这是我做的第一个网页

希望能在这次的学习中，收获很多的知识。



比特科技
BIT-TECHNOLOGY

HTML(Hyper Text Markup Language)

HTML 不是一门编程语言，而是一种用来告知浏览器如何组织页面的**标记语言**。HTML 可复杂、可简单，一切取决于开发者。它由一系列的**元素 (elements)** 组成，这些元素可以用来包围不同部分的内容，使其以某种方式呈现或者工作。一对标签 (tags) 可以为一段文字或者一张图片添加超链接，将文字设置为斜体，改变字号，等等。

元素(Element)

```
<p>我爱我的祖国</p>
```



- 开始标签：包含元素的名称（本例为 p），被左、右角括号所包围。表示元素从这里开始或者开始起作用。
- 与开始标签相似，只是其在元素名之前包含了一个斜杠。这表示着元素的结尾。
- 元素的内容，本例中就是所输入的文本本身。
- 开始标签、结束标签与内容相结合，便是一个完整的元素。

HTML 中事先规定了一些元素，每种元素都有其语义(semantic)，之后我们会学习其中一些场景的元素。这里的 p 是段落(paragraph)的语义。

[元素文档](#)

嵌套元素(Nesting Element)

可以把元素放到其它元素之中——这被称作嵌套。

```
<p>我爱我的<strong>祖国</strong></p>
```

这里使用 `` 标签，来**强调** 祖国 这一内容。

块级(block)元素和内联(inline)元素

```
<p>第一段</p>
<p>第二段</p>
<p>第三段</p>
<p>第四段</p>
```

```
<strong>第一句话</strong>
<strong>第二句话</strong>
<strong>第三句话</strong>
<strong>第四句话</strong>
```

- 块级元素在页面中默认以块的形式展现。更多的是表示一类段落的语义。

- 内联元素不会导致文本换行。更多的表示一个词、一句话等语义。

HTML5重新定义了元素的类别。尽管这些新的定义更精确，但却比上述的“块级元素”和“内联元素”更难理解，因此在之后的讨论中仍使用旧的定义。

在这篇文章中提到的“块”和“内联”，不应该与 CSS 中的同名术语相混淆。尽管他们默认是相关的，但改变CSS显示类型并不会改变元素的分类，也不会影响它可以包含和被包含于哪些元素。防止这种混淆也是HTML5摒弃这些术语的原因之一。

空元素(Empty Element)

不是所有元素都拥有开始标签，内容，结束标签。一些元素只有一个标签，通常用来在此元素所在位置插入/嵌入一些东西。例如：元素 `` 是用来在元素 `` 所在位置插入一张指定的图片。

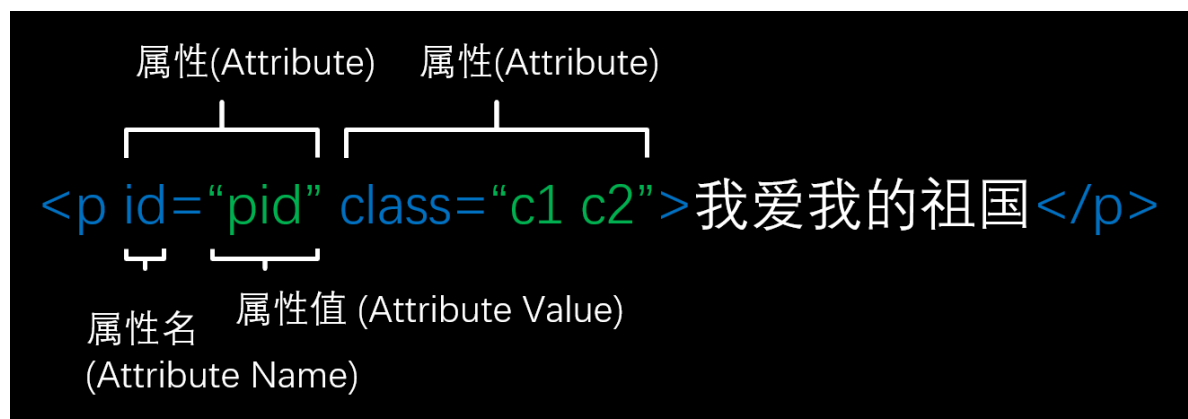
```

```

空元素 (Empty elements) 有时也被叫作 空元素 (Void elements).

属性(Attributes)

元素可以拥有属性



属性包含元素的额外信息，这些信息不会出现在实际的内容中。在上述例子中，我们指定了 id 这个属性，它的值是 pid，还指定了 class 这个属性，它的值是 c1 和 c2。

一个属性必须包含如下内容：

1. 一个空格，在属性和元素名称之间。(如果已经有一个或多个属性，就与前一个属性之间有一个空格。)
2. 属性名称，后面跟着一个等于号。
3. 一个属性值，由一对引号“”引起来。

元素的全局属性

当你浏览那些粗糙的web网站，你将会看见各种各样奇怪的标记风格，其中就有不给属性值添加引号。在某些情况下它是被允许的，但是其他情况下会破坏你的标记。例如，我们可以写一个只拥有一个href属性的链接，如下：

```
<a href=https://www.mozilla.org/>收藏页面</a>
```

然而，当我们再添加一个title属性时就会出错，如下：

```
<a href=https://www.mozilla.org/ title=The Mozilla homepage>收藏页面</a>
```

此时浏览器会误解你的标记，它会把title属性理解为三个属性——title的属性值为"The"，另外还有两个布尔属性“Mozilla”和“homepage”。看下面的例子，它明显不是我们所期望的，并且在这个编码里面它会报错或者出现异常行为。试一试把鼠标移动到链接上，看会显示什么title属性值！

我们建议始终添加引号——这样可以避免很多问题，并且使代码更易读。

布尔属性

一些属性，本身可以没有值，这类属性被称为布尔属性。我们更多关心的是这个元素是否有这个属性。

```
<input type="text" disabled>
<input type="text" disabled="true">
<input type="text" disabled="false">
<input type="text" disabled="随便写什么">
```

上述四种写法，其实效果是完全一样的。都是使得一个输入(input)禁止输入(disabled)。

大家可以和以下形式做个对比

```
<input type="text">
```

单引号或者双引号？

HTML 中，单引号和双引号的地位是相同的，以下两种情况都可以：

```
<a href="http://www.example.com">示例站点链接</a>
<a href='http://www.example.com'>示例站点链接</a>
```

在一个HTML中已使用一种引号，你可以在此引号中嵌套另外一种引号：

```
<a href="http://www.example.com" title="你觉得'好玩吗'?">示例站点链接</a>
<a href="http://www.example.com" title='你觉得"好玩吗"? '>示例站点链接</a>
```

文档结构(Document Structure)

```
<!DOCTYPE html>
<html lang="zh-hans">
  <head>
    <meta charset="utf-8">
    <title>我的第一个网页</title>
  </head>
  <body>
    <h1>这是我做的第一个网页</h1>
    <p>希望能在本次的学习中，收获很多的知识。</p>
    
  </body>
</html>
```

1. `<!DOCTYPE html>`: 声明文档类型. 很久以前使用的, 现在大家都统一使用这个, 表示 HTML5 就可以。
2. `<html lang="zh-hans"></html>`: `<html>` 元素。这个元素包裹了整个完整的页面, 是一个根元素。
 1. `<html>` 是一个 HTML 文档的唯一根元素(Root Element)。虽然不遵守这个约定, 大多数浏览器也可以识别, 但我们还是希望大家遵循。
 2. `lang="zh-hans"` 属性表示这个页面是一个**简体中文**的内容。
3. `<head></head>`: `<head>` 元素。这个元素是一个容器, 它包含了所有你想包含在HTML页面中但不想在HTML页面中显示的内容。这些内容包括你想在搜索结果中出现的关键字和页面描述, CSS样式, 字符集声明等等。
4. `<meta charset="utf-8">`: 设置文档使用utf-8字符集编码。
5. `<title></title>`: 设置页面标题, 出现在浏览器标签上, 当你标记/收藏页面时它可用来描述页面。
6. `<body></body>`: 包含了你访问页面时所有显示在页面上的内容, 文本, 图片, 音频, 游戏等等。

HTML 注释

为了将一段HTML中的内容置为注释, 你需要将其用特殊的记号 `<!--` 和 `-->` 包括起来, 比如:

```
<p>我在注释外! </p>

<!-- <p>我在注释内! </p> -->
```

实体引用(Entity References)

在HTML中, 字符 `<`, `>`, `"`, `'` 和 `&` 是特殊字符. 它们是HTML语法自身的一部分, 那么你如何将这些字符包含进你的文本中呢, 比如说如果你真的想要在文本中使用符号&或者小于号, 而不想让它们被浏览器视为代码并被解释?

我们必须使用字符引用 —— 表示字符的特殊编码, 它们可以在那些情况下使用. 每个字符引用以符号&开始, 以分号(;)结束.

常见实体引入如下:

原义字符	等价字符引用
<	< lt 是小于(little) 的意思
>	> gt 是大于(great) 的意思
"	" quot 是引号(quotation mark) 的意思
'	' apos 是单引号(apostrophe)的意思
&	& amp 是 and 符(ampersand)的意思

示例：

```
<p>HTML 中用 <p> 来定义段落元素。</p>

<p>HTML 中用 &lt;p&gt; 来定义段落元素</p>
```

HTML中的空白

在上面的例子中，你可能已经注意到了在代码中包含了很多的空格——这是没有必要的；下面的两个代码片段是等价的：

```
<p>狗 狗 很 呆 萌。</p>

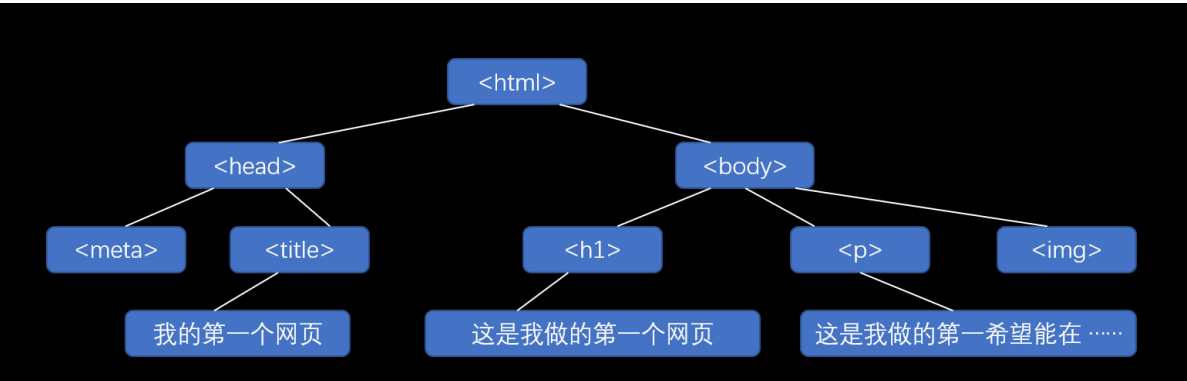
<p>狗 狗      很
    呆 萌。</p>
```

无论你在HTML元素的内容中使用多少空格(包括空白字符，包括换行)，当渲染这些代码的时候，HTML解释器会将连续出现的空白字符减少为一个单独的空格符。

那么为什么我们会在HTML元素的嵌套中使用那么多的空白呢? 答案就是为了可读性 —— 如果你的代码被很好地进行格式化，那么就很容易理解你的代码是怎么回事，反之就只有聚做一团的混乱。在我们的HTML代码中，我们让每一个嵌套的元素以两个空格缩进。你使用什么风格来格式化你的代码取决于你(比如所对于每层缩进使用多少个空格)，但是你应该坚持使用某种风格。

DOM(Document Object Model) 树

DOM（Document Object Model——文档对象模型）是载入到浏览器中的文档模型，以节点树的形式来表现文档，每个节点代表文档的构成部分（例如:页面元素、字符串或注释等等）。



常见的元素介绍

标题: `<h1>` 到 `<h6>`

```
<h1>学习 Java</h1>
<h2>学习 Java 基础</h2>
<h2>学习 数据结构</h2>
<h3>学习线性表</h3>
<h3>学习树</h3>
<h2>学习 Java Web</h2>
```

段落: `<p>`

```
<h1>为什么数据结构很重要</h1>
<p>因为在计算机的使用过程中，数据结构使用的非常频繁。</p>
```

列表:

无序列表(Unordered Lists): ``

有序列表(Ordered Lists): ``

列表项目(List Items): ``

```
<h1>早餐提供: </h1>
<ul>
  <li>豆浆</li>
  <li>油条</li>
  <li>豆腐脑</li>
  <li>米汤</li>
</ul>
```

```
<h1>学校怎么走? </h1>
<ol>
  <li>沿着条路走到头</li>
  <li>右转</li>
  <li>直行穿过第一个十字路口</li>
  <li>在第三个十字路口处左转</li>
  <li>继续走 300 米，学校就在你的右手边</li>
</ol>
```

嵌套列表:

```
<ol>
  <li>先用蛋白一个、盐半茶匙及淀粉两大匙搅拌均匀，调成“腌料”，鸡胸肉切成约一厘米见方的碎丁并用“腌料”搅拌均匀，腌渍半小时。</li>
  <li>用酱油一大匙、淀粉水一大匙、糖半茶匙、盐四分之一茶匙、白醋一茶匙、蒜末半茶匙调拌均匀，调成“综合调味料”。</li>
  <li>鸡丁腌好以后，色拉油下锅烧热，先将鸡丁倒入锅内，用大火快炸半分钟，炸到变色之后，捞出来沥干油汁备用。</li>
  <li>在锅里留下约两大匙油，烧热后将切好的干辣椒下锅，用小火炒香后，再放入花椒粒和葱段一起爆香。随后鸡丁重新下锅，用大火快炒片刻后，再倒入“综合调味料”继续快炒。
    <ul>
      <li>如果你采用正宗川菜做法，最后只需加入花生米，炒拌几下就可以起锅了。</li>
      <li>如果你在北方，可加入黄瓜丁、胡萝卜丁和花生米，翻炒后起锅。</li>
    </ul>
  </li>
</ol>
```

超链接：<a>

```
<p>创建了一个指向<a href="http://bitedu.vip/">比特论坛</a>的超链接。</p>
```

图片：

```
<h1>一张 比特 的图片</h1>

```

块级无语义元素：<div>

无语义元素，通常是用来组织内容，方便之后通过 css 或者 js 对其进行布局或操作。

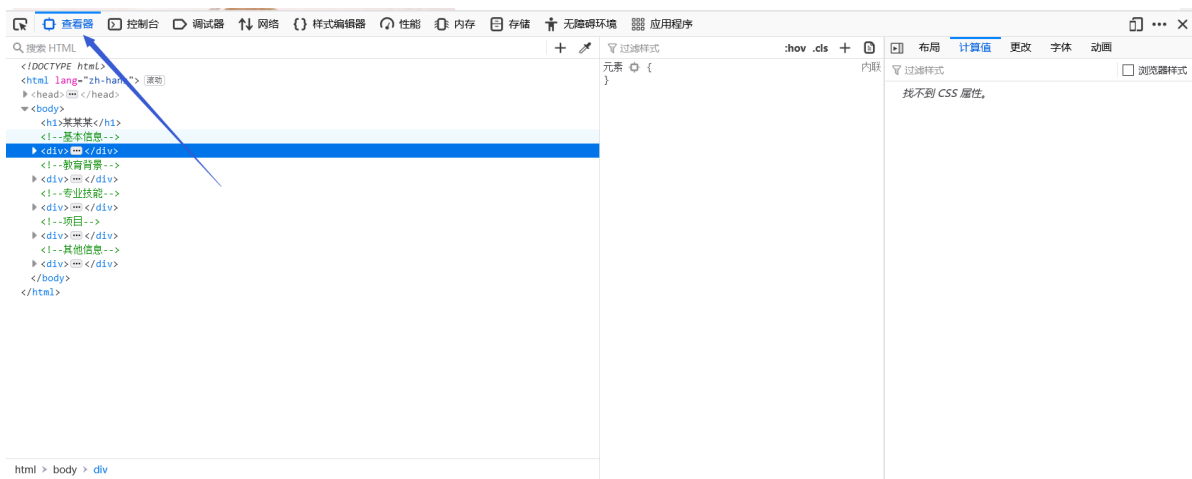
```
<!-- 某个版块 -->
<div>
  <h2>...</h2>
  <p>...</p>
</div>

<!-- 另一个版块 -->
<div>
  <h2>...</h2>
  <p>...</p>
</div>
```

内联无语义元素：

```
<p>大家一起往前<span>冲</span>啊！</p>
```

借助工具 —— 浏览器开发者工具



练一练 —— 完成个人名片

最终效果参考：

某某某

基本信息



求职意向: Java 开发工程师

联系电话: XXX-XXX-XXXX

邮箱: xxx@foxmail.com

[我的 github](#)

[我的 博客](#)

教育背景

1. 1990 - 1996 小葵花幼儿园 幼儿园
2. 1996 - 2002 小葵花小学 小学
3. 2002 - 2005 小葵花中学 初中
4. 2005 - 2008 小葵花中学 高中
5. 2008 - 2012 小葵花大学 计算机专业 本科

专业技能

- Java 基础语法扎实, 已经刷了 800 道 Leetcode 题;
- 常见数据结构都可以独立实现并熟练应用;
- 熟知计算机网络理论, 并且可以独立排查常见问题;
- 掌握 Web 开发能力, 并且独立开发了学校的留言墙功能。

我的项目

1. 留言墙

开发时间: 2008年9月 到 2008年12月

功能介绍：

- 支持留言发布
- 支持匿名留言

2. 学习小助手

开发时间：2008年9月 到 2008年12月

功能介绍：

- 支持错题检索
- 支持同学探讨

个人评价

在校期间，学习成绩优良，多次获得奖学金。

参考文档

[MDN HTML 介绍](#)

[MDN HTML 文档](#)

CSS(Cascading Style Sheets)

综合练习 - 个人名片最终效果参考