

# HTML&CSS

无08 王知衡 [wangzhih20@mails.tsinghua.edu.cn](mailto:wangzhih20@mails.tsinghua.edu.cn)

这里介绍的知识非常基础，因为网站组的同学们在后面写网站的时候很少直接写 HTML&CSS（尤其是HTML），网站组的前端框架一般只需要写JS和TS即可，但是HTML和CSS作为基础的前端框架，还是需要了解的，以后的工作中，还是会用的一些HTML的语法。这里讲只是为了让大家简单了解下网页的机制。

因为HTML和CSS没有什么难度，大多数都是一些细节的知识，所以大家在听基本概念的时候了解即可，细节问题可以实际写代码时再看，没有必要去死记硬背一些属性，在实践中一边用一边查一边记即可。

本文档参考[MDN](#)、[菜鸟教程](#)完成，希望深入学习的同学可以参考（MDN上有关于web比较详细的教程和文档），也感谢上一届电子系科协张凯学长的讲义，我的讲义是基本按照他的来改的。

## HTML

### 基础

超文本标记语言（英语：**Hypertext Markup Language**，简称：HTML）是一种用来结构化 Web 网页及其内容的标记语言。

HTML 不是一门编程语言，而是一种用于定义内容结构的 *标记语言*。HTML 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。

举个简单的例子：

欢迎大家加入科协

可以将这行文字封装成一个段落（paragraph）元素来使其在单独一行显示：

```
<p>欢迎大家加入科协</p>
```

HTML 标签不区分大小写。不过，从一致性、可读性来说，最好仅使用小写字母。

这个元素的主要部分有：

1. **开始标签**（Opening tag）：包含元素的名称（本例为 p），被大于号、小于号所包围。表示元素从这里开始或者开始起作用——在本例中即段落由此开始。
2. **结束标签**（Closing tag）：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——在本例中即段落在此结束。初学者常常会犯忘记包含结束标签的错误，这可能会产生一些奇怪的结果。
3. **内容**（Content）：元素的内容，本例中就是所输入的文本本身。
4. **元素**（Element）：开始标签、结束标签与内容相结合，便是一个完整的元素。

一些常元素的标签：

- `<h1>` - `<h6>`：HTML标题
- `<hr>`：水平线
- `<br>`：回车

## 嵌套元素

你可以把元素放到其它元素之中——这被称作嵌套。

欢迎大家加入科协

```
<p><em>欢迎</em>大家加入<strong>科协</strong></p>
```

注意元素需要被正确地嵌套。

## 块级元素和内联元素

在 HTML 中有两种重要元素类别，块级元素和内联元素。

- 块级元素在页面中以块的形式展现 —— 相对于其前面的内容它会出现在新的一行，其后的内容也会被挤到下一行展现。块级元素通常用于展示页面上结构化的内容，例如段落、列表、导航菜单、页脚等等。一个以 block 形式展现的块级元素不会被嵌套进内联元素中，但可以嵌套在其它块级元素中。
- 内联元素通常出现在块级元素中并环绕文档内容的一小部分，而不是一整个段落或者一组内容。内联元素不会导致文本换行：它通常出现在一堆文字之间例如超链接元素 `<a>` 或者强调元素 `<em>` 和 `<strong>`。

## 空元素

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 `<img>` 是用来在元素 `<img>` 所在位置插入一张指定的图片。例子如下：

```

```

## 属性

```
<p class="title">EESAST</p>
```

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，这个 class 属性给元素赋了一个识别的名字 (id)，这个名字此后可以被用来识别此元素的样式信息和其他信息。

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。

[百度一下](#)

```
<a href="https://www.baidu.com/" title="百度">百度一下</a>
```

常用属性：

- class：为html元素定义一个或多个类名 (classname)
- id：定义元素的唯一id

## 布尔属性

有时你会看到没有值的属性，它是合法的。这些属性被称为布尔属性，他们只能有跟它的属性名一样的属性值。例如 `disabled` 属性，他们可以标记表单输入使之变为不可用 (变灰色)，此时用户不能向他们输入任何数据。

```
<input type="text" disabled="disabled">
<input type="text" disabled> <!--两个写法等效-->
```

## HTML文件的结构

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>我的测试站点</title>
  </head>
  <body>
    <p>这是我的页面</p>
  </body>
</html>
```

1. `<!DOCTYPE html>`: 声明文档类型。
2. `<html></html>`: `<html>` 元素。这个元素包裹了整个完整的页面，是一个根元素。
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">`: 这个元素设置文档使用 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>`: `<body>` 元素。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。

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。

## 在 HTML 中应用 CSS 和 JavaScript

对于 CSS，需要加入一个 `<link>` 元素，`rel="stylesheet"` 表明这是文档的样式表，而 `href` 包含了样式表文件的路径，这个元素一般放在 `<head></head>` 里：

```
<link rel="stylesheet" href="my-css-file.css" />
```

或者把脚本直接写在元素里，在标签里写 CSS 的内容，像这样：

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>菜鸟教程(runoob.com)</title>
    <style>
      #para1
      {
```

```

        text-align:center;
        color:red;
    }
</style>
</head>
<body>
    <p id="para1">Hello world!</p>
    <p>这个段落不受该样式的影响。</p>
</body>
</html>

```

对于 js 脚本，加入 `<script>` 元素，`src` 为脚本的路径。`<script>` 部分没必要非要放在文档头部，实际上，把它放在文档的尾部 `</body>` 标签之前是一个更好的选择，这样可以确保在加载脚本之前浏览器已经解析了 HTML 内容。

```
<script src="my-js-file.js"></script>
```

这里 script 元素没有内容，但他不是一个空元素，你也可以直接把脚本写在元素里。

```

<!DOCTYPE html>
<html>

<head>
    <meta charset="utf-8">
    <title>菜鸟教程(runoob.com)</title>
    <script>
        function displayDate() {
            document.getElementById("demo").innerHTML = Date();
        }
    </script>
</head>

<body>
    <h1>我的第一个 JavaScript 程序</h1>
    <p id="demo">这是一个段落</p>
    <button type="button" onclick="displayDate()">显示日期</button>
</body>

</html>

```

在标签内加上 `defer` 可以告诉浏览器在解析完成 HTML 后再加载 JavaScript。这样可以确保在加载脚本之前浏览器已经解析了所有的 HTML 内容（如果脚本尝试访问某个不存在的元素，浏览器会报错）。

```
<script src="my-js-file.js" defer></script>
```

## HTML的基本布局

- `<header>`：页眉。
- `<nav>`：导航栏。
- `<main>`：主内容。主内容中还可以有各种子内容区段，可用 `<article>`、`<section>` 和 `<div>` 等元素表示。
- `<aside>`：侧边栏，经常嵌套在 `<main>` 中。
- `<footer>`：页脚。

# CSS

## 基础

**CSS**（Cascading Style Sheets，层叠样式表）是用来控制网页在浏览器中的显示外观的声明式语言。浏览器会根据 CSS 的样式定义将其选定的元素显示为恰当的形式。一条 CSS 的样式定义包括属性和属性值，它们共同决定网页的外观。

HTML 用于定义内容的结构和语义，CSS 用于设计风格和布局。比如，你可以使用 CSS 来更改内容的字体、颜色、大小、间距，将内容分为多列，或者添加动画及其他的装饰效果。

```
h1 {  
  color: red;  
  font-size: 5em;  
}
```

语法由一个选择器 (selector) 起头。它选择 (selects) 了我们将要用来添加样式的 HTML 元素。在这个例子中我们为一级标题（主标题 `<h1>`）添加样式。

接着输入一对大括号 `{ }`。在大括号内部定义一个或多个形式为 **属性 (property): 值 (value);** 的 **声明 (declarations)**。每个声明都指定了我们所选择元素的一个属性，之后跟一个我们想赋给这个属性的值。

冒号之前是属性，冒号之后是值。不同的 CSS 属性 (properties) 对应不同的合法值。在这个例子中，我们指定了 `color` 属性，它可以接受许多颜色值；还有 `font-size` 属性，它可以接收许多 size units 值。

## 元素选择器

使用元素选择器来直接匹配一种 html 元素

```
p {  
  color: green;  
}
```

不同选择器可以用 `,` 隔开使用多个选择器

```
p, li {  
  color: green;  
}
```

这种直接指定元素类型的写法只能同时应用于所有同类元素，在实际使用的时候是十分不方便的，因此我们还可以使用类名来进一步指定作用的对象。前面说到 HTML 中的元素可以添加属性，这里的类名选择器就对应标签中的 `class` 属性，比如：

```
<ul>  
  <li>项目一</li>  
  <li class="special">项目二</li>  
  <li>项目<em>三</em></li>  
</ul>
```

```
.special {  
  color: orange;  
  font-weight: bold;  
}
```

这里指定了属于 special 类的元素进行设置，在选择器的开头加 . 即为指定类，当然可以同时指定元素和类：

```
li.special,  
span.special {  
  color: orange;  
  font-weight: bold;  
}
```

也可以为HTML 中的元素添加id，id选择器对应标签中的id属性：

```
#unique { }
```

我们还可以根据元素的位置来进行选择，在选择器中使用 空格 来表示包含（嵌套）关系，用 + 来表示相邻关系，例如：

```
li em {  
  color: red;  
}
```

表示选择嵌套在 <li> 内部的 <em>

```
h1 + p {  
  color: blue;  
}
```

表示选择跟在 <h1> 后面的 <p>