

HTML & CSS Study Notes

1 HTML5 介绍

- 1-2 html 和 css 的关系

```
<p>我是一个 p 标签</p>
```

为 p 标签添加 css 修饰:

```
p {
    color: red;
    border: 1px solid blue;
    width: 140px;
    height: 40px;
}
```

定义一个 h1 标签:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Html 和 CSS 的关系</title>
    <style type="text/css">
      h1{
        font-size:12px;
        color:#930;
        text-align:center;
      }
    </style>
  </head>
  <body>
    <h1>Hello World!</h1>
  </body>
</html>
```

解释:

- css 是用来修饰 html 样式的
- html 本身是有一些默认样式, 如果我们想改变 html 标签的样式, 就需要借助 css
- html+css 构成了我们网页的基本页面结构和样式

- 1-3 使用说明书 - 标签的语法

1. 标签由英文尖括号 `<` 和 `>` 括起来, 如 `<html>` 就是一个标签。
2. html 中的标签一般都是成对出现的, 分开始标签和结束标签。结束标签比开始标签多了一个 `/`。
3. 标签与标签之间是可以嵌套的, 但先后顺序必须保持一致, 如: `<div>` 里嵌套 `<p>`, 那么 `</p>` 必须放在 `</div>` 的前面。如下图所示。
4. HTML 标签不区分大小写, `<h1>` 和 `<H1>` 是一样的, 但建议小写, 因为大部分程序员都以小写为准。

- 1-4 html5 文档结构

```

<!DOCTYPE html> <!--声明文档类型为 html5-->
<html> <!--html 标签为 html 文件的根标签-->
  <head> <!--head 标签为头部标签，一般用来放置 meta 和 title 标签等-->
    <meta charset="UTF-8"> <!--声明当前文件字符集编码为 UTF-8-->
    <title>html 文档结构</title> <!--title 标签设置浏览器标题-->
  </head>
  <body>
    <!--body 标签为网页的内容-->
  </body>
</html>

```

1. `<!DOCTYPE html>`：文档类型声明，表示该文件为 HTML5 文件。`<!DOCTYPE>` 声明必须是 HTML 文档的第一行，位于 `<html>` 标签之前
 2. `<html></html>` 标签对：`<html>` 标签位于 HTML 文档的最前面，用来标识 HTML 文档的开始；`</html>` 标签位于 HTML 文档的最后面，用来标识 HTML 文档的结束；这两个标签对成对存在，中间的部分是文档的头部和主题。
 3. `<head></head>` 标签对：标签包含有关 HTML 文档的信息，可以包含一些辅助性标签。如 `<title></title>`，`<link /><meta />`，`<style></style>`，`<script></script>` 等，但是浏览器除了会在标题栏显示 `<title>` 元素的内容外，不会向用户显示 head 元素内的其他任何内容。
 4. `<body></body>` 标签对：它是 HTML 文档的主体部分，在此标签中可以包含 `<p><h1>
` 等众多标签，`<body>` 标签出现在 `</head>` 标签之后，且必须在闭标签 `</html>` 之前闭合。
- 1-5 认识 head 标签

```

<head>
  <!--head 标签为头部标签，通常嵌套 meta、title、style 等标签-->
  <meta charset="UTF-8">
  <title>head 标签</title> <!--设置浏览器标题-->
  <!--书写 css 样式-->
  <style>
    #box1 {
      color: red;
    }
  </style>
</head>

```

文档的头部描述了文档的各种属性和信息，包括文档的标题等，绝大多数文档头部包含的数据都不会真正作为内容显示给读者。

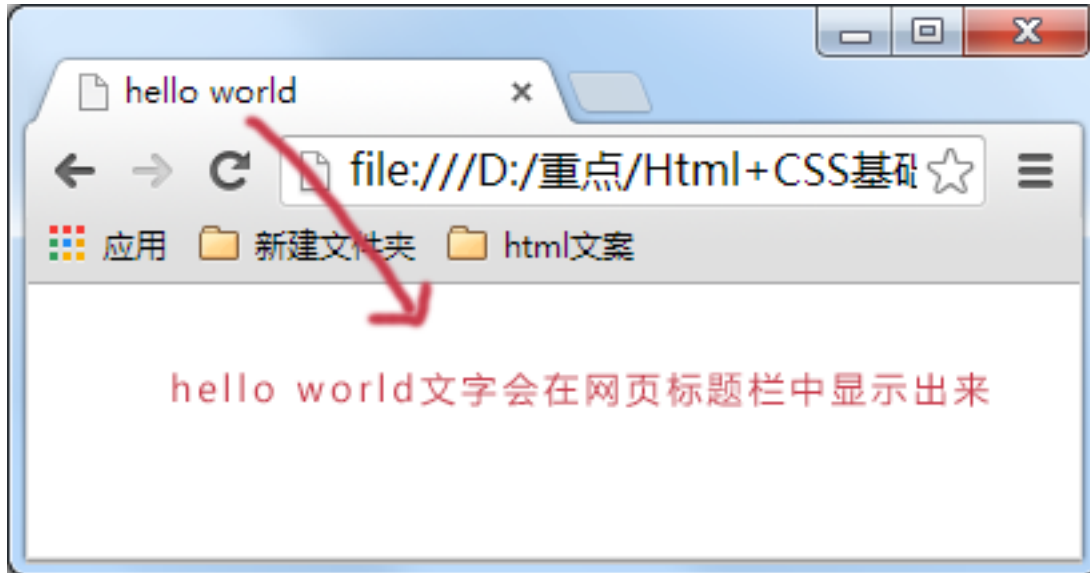
下面这些标签可用在 head 部分：

- 1、`head` 标签为双标签，有尾标签，`<head></head>`。
- 2、`head` 标签表示头部标签，通常用来嵌套 `meta`、`title`、`style` 等标签。
- 3、`<title>` 标签：在 `<title>` 和 `</title>` 标签之间的文字内容是网页的标题信息，它会出现在浏览器的标题栏中。网页的 `<title>` 标签用于告诉用户和搜索引擎这个网页的主要内容是什么，搜索引擎可以通过网页标题，迅速的判断出网页的主题。每个网页的内容都是不同的，每个网页都应该有一个独一无二的 title。
- 4、`<meta charset="UTF-8">` 设置当前文件字符编码
- 5、`style` 标签：双标签中设置当前文件样式

例如 title 标签：

```
<head>
  <title>hello world</title>
</head>
```

`<title>` 标签的内容“hello world”会在浏览器中的标题栏上显示出来，如下图所示：



- 1-6 认识 body 标签

在网页上要展示出来的页面内容一定要放在 `body` 标签中。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>了不起的盖茨比</title>
  </head>
  <body>
    <!-- 标题标签 -->
    <h1>了不起的盖茨比</h1>
    <!-- 段落标签 -->
    <p>1922 年的春天，一个想要成名名叫尼克 • 卡拉威（托比 • 马奎尔 Tobey Maguire 饰）的作家，离开了美国中西部，来到了纽约。那是一个道德感渐失，爵士乐流行，走私为王，股票飞涨的时代。为了追寻他的 <span>美国梦</span>，他搬入纽约附近一海湾居住。</p>
    <!-- 段落标签 -->
    <p>菲茨杰拉德，二十世纪美国文学巨擘之一，兼具作家和编剧双重身份。他以诗人的敏感和戏剧家的想象为"爵士乐时代"吟唱华丽挽歌，其诗人和梦想家的气质亦为那个奢靡年代的不二注解。</p>
  </body>
</html>
```

- 1-7 html 文件注释

什么是代码注释？代码注释的作用是帮助程序员标注代码的用途，过一段时间后再看你所编写的代码，就能很快想起这段代码的用途。代码注释不仅方便程序员自己回忆起以前代码的用途，还可以帮助其他程序员很快的读懂你的程序的功能，方便多人合作开发网页代码。

语法：

```
<!--**** 注释文字 -->
```

2 HTML5 语义化标签

- 2-1 语义化

标签的用途：我们学习网页制作时，常常会听到一个词，语义化。那么什么叫做语义化呢，说的通俗点就是：明白每个标签的用途（在什么情况下使用此标签合理）比如，网页上的文章的标题就可以用标题标签，网页上的各个栏目的栏目名称也可以使用标题标签。文章中内容的段落就得放在段落标签中等等。

好处：

1. 更容易被搜索引擎收录。
2. 更容易让屏幕阅读器读出网页内容。

- 2-2 段落标签

如果想在网页上显示文章，这时就需要 `<p>` 标签了，把文章的段落放到 `<p>` 标签中。

语法：

```
<p>段落文本</p>
```

注意一段文字一个 `<p>` 标签，如在一篇新闻文章中有 3 段文字，就要把这 3 个段落分别放到 3 个 `<p>` 标签中。如下图所示

```
<body>
```

```
<p>"玉米炒葡萄"的出现，引得一众大学生网友纷纷晒出菜单，盘点高校食堂的"奇葩菜品"。比如2011年华南理工大学广州学院的豉油鸡头；尤其擅用瓜果类的武汉大学食堂师傅"自创"了"中式沙拉"苹果炒西瓜、糖醋哈密瓜炒里脊和红枣银耳煮西瓜；曾风靡一时的月饼炒辣椒"出品"自福建师范大学；湖北经济学院食堂的师傅更是煞费苦心，为防H7N9"发明"了番茄炒菠萝。</p>
```

```
<p>而近期面世的"玉米炒葡萄"，被称为继上述"奇葩"菜式之后大学食堂的"又一神作"。有网友称赞"玉米炒葡萄"是"食堂饭菜中的小清新"，但@猫扑则忍不住喊道："大厨，你是跟汽修学校学的炒菜吗？"还有人高呼，"这是高校食堂要创立中国第九大菜系的节奏么？"</p>
```

```
<p>对于层出不穷的"雷人"菜品，网友们的看法可谓五花八门，饶有趣味。"真是把人雷得外焦内嫩啊"，"月饼炒辣椒，是中秋节吃剩的月饼的最佳处理方式，没有之一"</p>
```

```
</body>
```

在浏览器中显示的效果：



`<p>` 标签的默认样式，可以在上图中看出来，段前段后都会有空白，如果不喜欢这个空白，可以用 CSS 样式来删除或改变它，在后面课程中会学习到。

- 2-3 使用 `` 标签自定义文字样式

`` 标签是没有语义的，它的作用就是为了设置单独的样式用的。

如果现在我们把下面的第一段话“美国梦”三个字设置成 blue（蓝色），所以这样情况下就可以用到 `` 标签了。

如下面例子：

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>了不起的盖茨比</title>
    <style>
      span {
        color:blue
      }
    </style>
  </head>
  <body>
    <p>为了追寻他的<span>美国梦</span>，他搬入纽约附近一海湾居住。</p>
  </body>
</html>
```

语法：

```
<span>文本</span>
```

- 2-4 使用 `<h1>` 标签为网页增加标题

使用 `<hx>` 标签来制作文章的标题。标题标签一共有 6 个，`h1`、`h2`、`h3`、`h4`、`h5`、`h6` 分别为一级标题、二级标题、三级标题、四级标题、五级标题、六级标题。并且依据重要性递减。`<h1>` 是最高等级。语法：`<hx>` 标题文本 `</hx>` (x 为 1-6) 文章的标题前面已经说过了，可以使用标题标签，另外网页上的各个栏目的标题也可使用它们。如下图为腾讯网站

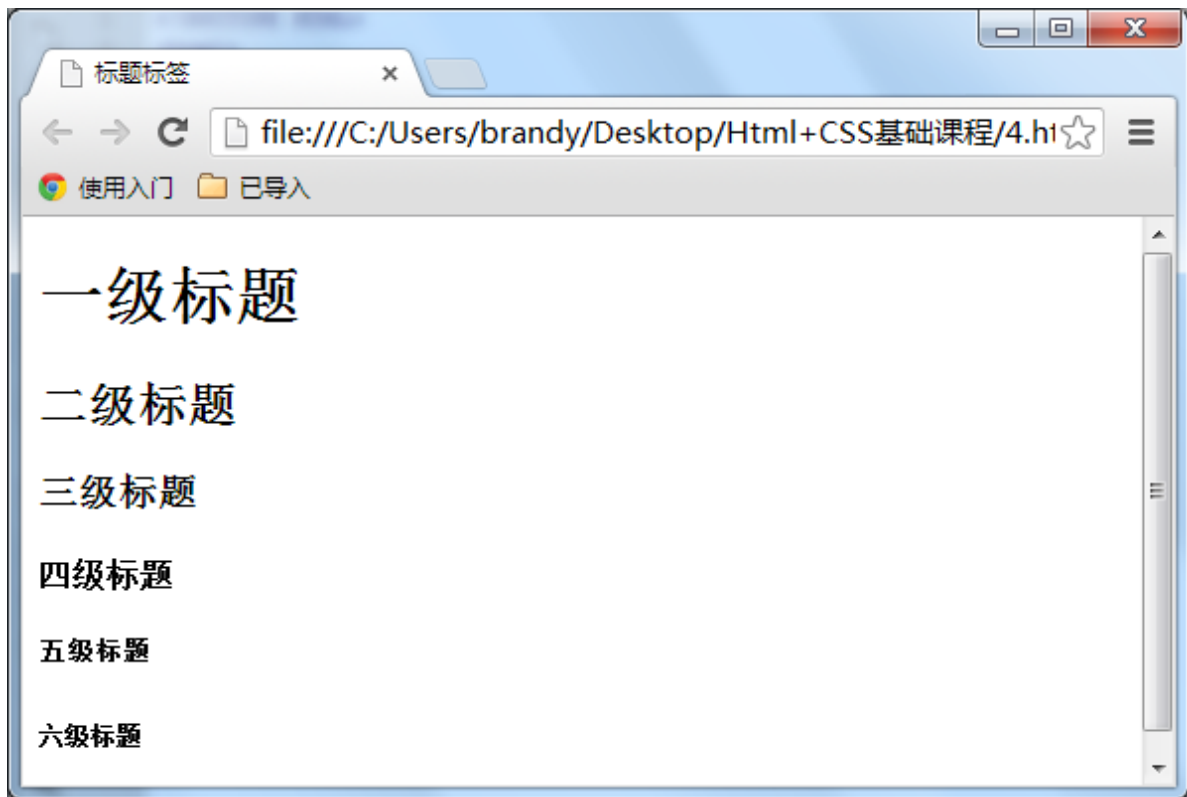


注意：因为 `h1` 标签在网页中比较重要，所以一般 `h1` 标签被用在网站名称上。腾讯网站就是这样做的。如：
`<h1>` 腾讯网 `</h1>`

`h1-h6` 标签的默认样式：

```
<body>
  <h1>一级标题</h1>
  <h2>二级标题</h2>
  <h3>三级标题</h3>
  <h4>四级标题</h4>
  <h5>五级标题</h5>
  <h6>六级标题</h6>
</body>
```

在浏览器中显示的样式：



从上面的图片可以看出标题标签的样式都会加粗，`h1` 标签字号最大，`h2` 标签字号相对 `h1` 要小，以此类推 `h6` 标签的字号最小。

- 2-5 使用 `<div>` 标签自定义块

在网页制作过程中，可以把一些独立的逻辑部分划分出来，放在一个 `<div>` 标签中，这个 `<div>` 标签的作用就相当于一个容器。

语法：

```
<div></div>
```

确定逻辑部分：

什么是逻辑部分？它是页面上相互关联的一组元素。如网页中的独立的栏目版块，就是一个典型的逻辑部分。如下图所示：图中用红色边框标出的部分就是一个逻辑部分，就可以使用 `<div>` 标签作为容器。



注意：仅仅是添加 `<div>` 标签对网页显示是没有变化的，还需要额外为 `<div>` 标签设置标签样式

• 2-6 `<header>` 标签定义头部区域

HTML5 新增的语义化标签，用来定义头部区域。如：



注意：和 `<head>` 标签的区别：`<head>` 是和 `<body>` 标签并列的，而 `<header>` 标签在 `<body>` 标签内使用。

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>header</title>
</head>
<body>
  <!--header 标签代表头部，但作用等同于 div，只是具备语义化-->
  <header></header>
</body>
</html>
```

• 2-7 `<footer>` 标签定义底部区域



```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>footer</title>
</head>
<body>
  <!--header 标签代表底部，但作用等同于 div，只是具备语义化-->
  <footer></footer>
</body>
</html>
```


- 2-8 `<section>` 定义区段

定义一个区域：和 `<div>` 标签没有本质区别

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>footer</title>
</head>
<body>
  <!--section 标签代表一个区域，但作用等同于 div，只是具备语义化-->
  <section></section>
</body>
</html>
```

- 2-9 `<aside>` 定义区段

定义一个侧边栏区域。

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>footer</title>
</head>
<body>
  <!--aside 标签代表侧边栏区域，但作用等同于 div，只是具备语义化-->
  <aside></aside>
</body>
</html>
```

3 HTML5 效果标签

- 3-1 使用 `
` 标签实现换行效果

在需要加回车换行的地方加入 `
`，作用相当于 word 文档中的回车。

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>br 标签的使用</title>
</head>
<body>
  <h2>《咏桂》</h2>
  <p>暗淡轻黄体性柔，<br />
    情疏迹远只香留。<br />
    何须浅碧深红色，<br />
    自是花中第一流。
  </p>
</body>
</html>
```

语法：

xhtml1.0 写法：


```

<head>
  <meta charset="UTF-8">
  <title>hr 标签使用</title>
</head>
<body>
  <p>火车飞驰过暗夜里的村庄，月光，总是太容易让思念寂寞，太容易让人觉得孤独。</p>
  <hr />
  <p>每一枚被风吹起的蒲公英，都载满了一双眼睛的深情告别与一个目光的依依不舍。那天，我拿着行李，带上一个背影的祝福与惆怅，
</body>
</html>

```

语法：

xhtml1.0 写法：

```
<hr />
```

html4.01 写法：

```
<hr>
```

注意：

1. `<hr />` 标签和 `
` 标签一样也是一个空标签，所以只有一个开始标签，没有结束标签。
2. `<hr />` 标签的在浏览器中的默认样式线条比较粗，颜色为灰色，可能有些人觉得这种样式不美观，没有关系，这些外在样式在我们以后学习了 CSS 样式表之后，都可以对其修改。

4 HTML5 列表标签

- 4-1 使用 `` 标签实现无序列表

在浏览网页时，你会发现网页上有很多信息的列表，如新闻列表、图片列表，如下图所示：

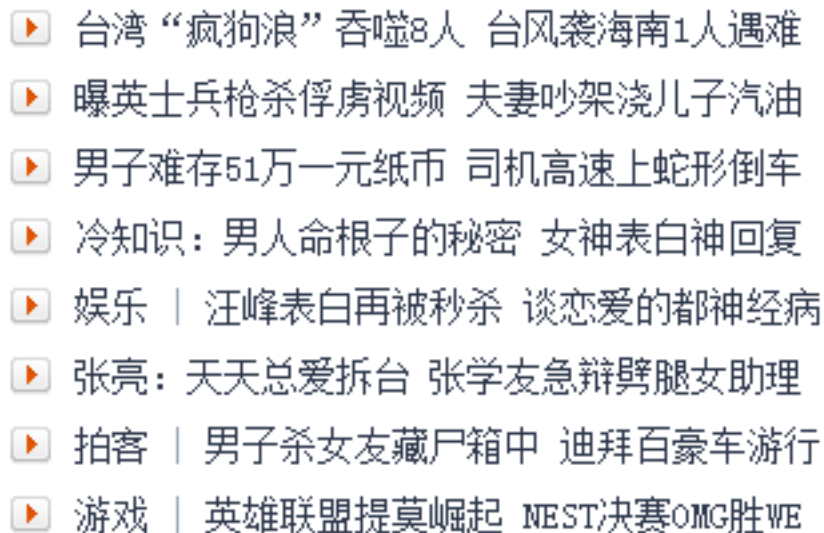


Figure 1: 新闻列表

这些列表就可以使用 `ul-li` 标签来完成。`ul-li` 是没有前后顺序的信息列表。

语法：



Party攻略



水果大作战



米菲兔01



娃娃和小熊

Figure 2: 图片列表

```
<ul>
  <li>信息</li>
  <li>信息</li>
  .....
</ul>
```

举例:

```
<ul>
  <li>精彩少年</li>
  <li>美丽突然出现</li>
  <li>触动心灵的旋律</li>
</ul>
```

ul-li 在网页中显示的默认样式一般为: 每项 li 前都自带一个圆点, 如下图所示:



- 4-2 使用 `` 标签实现有序列表

如果想在网页中展示有前后顺序的信息列表，怎么办呢？如，当当网上的书籍热卖排行榜，如下图所示。这类信息展示就可以使用 `` 标签来制作有序列表来展示。



语法：

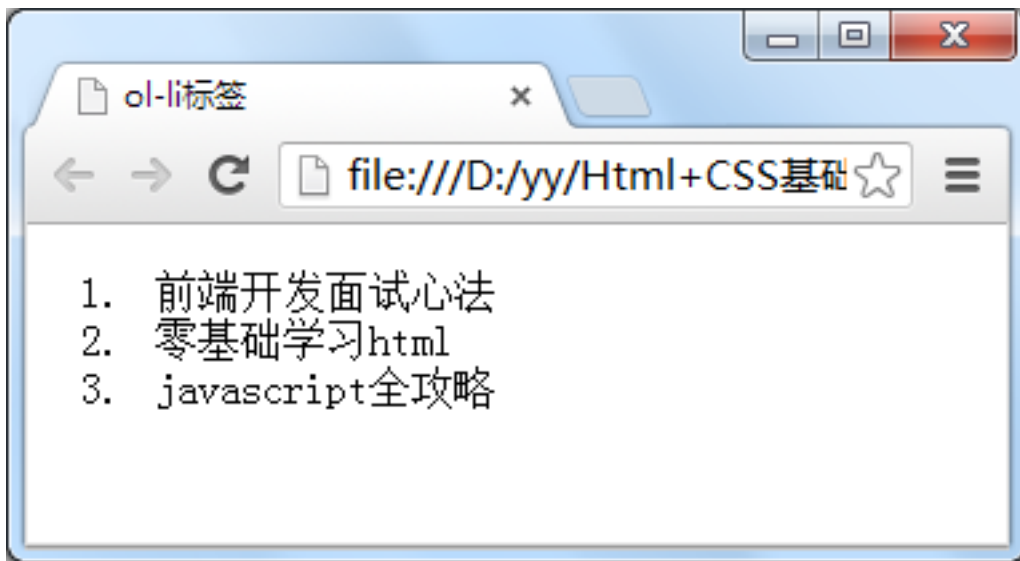
```
<ol>
  <li>信息</li>
  <li>信息</li>
  .....
</ol>
```

举例：

下面是一个热点课程下载排行榜：

```
<ol>
  <li>前端开发面试心法 </li>
  <li>零基础学习 html</li>
  <li>JavaScript 全攻略</li>
</ol>
```

`` 在网页中显示的默认样式一般为：每项 `` 前都自带一个序号，序号默认从 1 开始，如下图所示：



5 HTML5 图片、链接及表格标签

- 5-1 使用 `` 标签为网页添加图片

在网页的制作中为使网页炫丽美观，肯定是缺少不了图片，可以使用 `` 标签来插入图片。

语法：

```

```

举例：

```
<img src = "myimage.gif" alt = "My Image" title = "My Image" />
```

最后这个 `/` 无影响

讲解：

1. src: 标识图像的位置；
 2. alt: 指定图像的描述性文本，当图像不可见时（下载不成功时），可看到该属性指定的文本；
 3. title: 提供在图像可见时对图像的描述（鼠标滑过图片时显示的文本）；
 4. 图像可以是 GIF, PNG, JPEG 格式的图像文件。
- 5-2 使用 `<a>` 标签为网页添加超链接

使用 `<a>` 标签可实现超链接，它在网页制作中可以说是无处不在，只要有链接的地方，就会有这个标签。

语法：

```
<a href=" 目标网址" title=" 鼠标滑过显示的文本">链接显示的文本</a>
```

例如：

```
<a href="http://www.imooc.com" title=" 点击进入慕课网">click here!</a>
```

上面例子作用是单击 `click here!` 文字，网页链接到 `http://www.imooc.com` 这个网页。

title 属性的作用，鼠标滑过链接文字时会显示这个属性的文本内容。这个属性在实际网页开发中作用很大，主要方便搜索引擎了解链接地址的内容（语义化更友好），如右侧案例代码（11-13 行）。

注意：只要为文本加入 a 标签后，文字的颜色就会自动变为蓝色（被点击过的文本颜色为紫色），颜色很难看，可以通过 CSS 样式设置（a{color:#000}）。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>a 标签</title>
  </head>
  <body>
    <ul>
      <li><a href="#" title=" 零基础学习 html">零基础学习 html</a></li>
      <li><a href="#" title="JavaScript 全攻略">JavaScript 全攻略</a></li>
    </ul>
    <p>1922 年的春天，一个想要成名名叫尼克 • 卡拉威（<a href="http://www.m1905.com/mdb/star/3316/" title=
```

- 5-3 在新建浏览器窗口中打开链接

`<a>` 标签中的 target 属性：

新浪
腾讯

我们要实现这样的效果，可以输入以下代码：

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>a 标签 target 属性</title>
  </head>
  <body>
    <!--当前窗口打开链接对应的网页-->
    <a href="https://www.sina.com.cn" target="_self">新浪</a>
    <!--新窗口打开链接对应的网页-->
    <a href="https://www.qq.com" target="_blank">腾讯</a>
  </body>
</html>
```

技术点的解释：

`<a>` 标签有的 target 属性，代表打开网页的方式。可选值为 `_self` 和 `_blank`，默认值为 `_self`，代表在当前页面打开链接，`_blank` 代表在新窗口打开链接。

- 5-4 使用 table 家族为网页添加表格

有时候我们需要在网页当中添加一些数据，如某公司想展示该公司员工的通信录，如下表：

企业员工通讯录 → 标题标签

企业员工通讯录 → 表格标题 (和文章标题区分)

姓名	电话	电子邮件	职务
张三	18278900988	zhangsan@163.com	研发工程师
王二	16589012689	wanger@163.com	研发经理
李四	17230019065	lisi@163.com	研发工程师

→ 表格的行

表格的单元格 (行和列都是由单元格构成的)

表格的列

想要在网页中实现以上效果，我们可以使用以下代码：

```
<body>
  <!--标题标签-->
  <h3>企业员工通讯录</h3>
  <!--水平线标签-->
  <hr />
  <!--表格标签 border 属性代表给表格加上边框-->
  <table border="1">
    <!--表格标题标签-->
    <caption>企业员工通讯录</caption>
    <!--tr 代表一行-->
    <tr>
      <!--th 代表一列，有字体加粗效果-->
      <th>姓名</th>
      <th>电话</th>
      <th>电子邮件</th>
      <th>职务</th>
    </tr>
    <tr>
      <!--td 代表一列，字体正常效果-->
      <td>张三</td>
      <td>18278900988</td>
      <td>zhangsan@163.com</td>
      <td>研发工程师</td>
    </tr>
    <tr>
      <td>王二</td>
      <td>16589012689</td>
      <td>wanger@163.com</td>
      <td>研发经理</td>
    </tr>
    <tr>
      <td>李四</td>
      <td>17230019065</td>
```



```

        <td>lisi@163.com</td>
        <td>研发工程师</td>
    </tr>
</table>
</body>

```

解释：

创建表格的四个元素： `table` 、 `tr` 、 `th` 、 `td`

1. `<table>...</table>` ： 整个表格以 `<table>` 标记开始、`</table>` 标记结束。
2. `<tr>...</tr>` ： 表格的一行，所以有几对 `<tr></tr>` 表格就有几行。
3. `<td>...</td>` ： 表格的一个单元格，一行中包含几对 `<td></td>` ，说明一行中就有几列。
4. `<th>...</th>` ： 表格的头部一个单元格，表格表头。
5. 表格中列的个数，取决于一行中数据单元格的个数。
6. `border` 属性可以为表格添加边框，属性值为数字。

注意：

1. `table` 标签用来定义整个表格，为双标签，必须有结束标签。
2. `table` 标签里面可以放 `caption` 标签和 `tr` 标签。
3. `caption` 标签用来定义表格的标题。
4. `tr` 标签用来设置表格的行，`tr` 里面只能放 `th` 或者 `td` 标签，一组 `tr` 标签代表一行。
5. `th` 用来设置表格的标题，会加粗居中显示。也就是 `th` 标签中的文本默认为粗体并且居中显示。
6. `td` 用来设置表格的列，一组 `td` 标签代表一列。
7. `table` 表格在没有添加 `border` 属性之前，在浏览器中显示是没有表格线的。

例：

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>学习表格标签</title>
  </head>
  <body>
    <table border="1">
      <!--表格标题标签-->
      <caption>前端三剑客</caption>
      <!--tr 代表一行-->
      <tr>
        <!--th 代表一列，有字体加粗效果-->
        <th>知识点</th>
        <th>重要程度</th>
        <th>难度</th>
        <th>学习周期</th>
      </tr>
      <tr>
        <!--td 代表一列，字体正常效果-->

```

```

        <td>html</td>
        <td>5 星</td>
        <td>3 星</td>
        <td>7 天</td>
    </tr>
    <tr>
        <td>css</td>
        <td>5 星</td>
        <td>4 星</td>
        <td>10 天</td>
    </tr>
    <tr>
        <td>js</td>
        <td>5 星</td>
        <td>5 星</td>
        <td>20 天</td>
    </tr>
</table>
</body>
</html>

```

前端三剑客

知识点	重要程度	难度	学习周期
html	5星	3星	7天
css	5星	4星	10天
js	5星	5星	20天

- 5-5 使用 `thead` 、 `tbody` 、 `tfoot` 定义表格

- `<thead>` 标签定义表格头部
- `<tbody>` 标签来定义表格的内容
- `<tfoot>` 来定义表格的底部

实现一个成绩表，各科成绩和总分，

```

<body>
  <table border="1">
    <!--tr 代表一行-->
    <thead>
      <tr>
        <!--th 代表一列，有字体加粗效果-->
        <th>科目</th>
        <th>分数</th>
      </tr>
    </thead>

```

```

<tbody>
  <tr>
    <!--td 代表一列，字体正常效果-->
    <td>语文</td>
    <td>99</td>
  </tr>
  <tr>
    <td>数学</td>
    <td>60</td>
  </tr>
</tbody>

<tfoot>
  <tr>
    <td>总分</td>
    <td>159</td>
  </tr>
</tfoot>
</table>
</body>

```

效果如下图：

成绩表

科目	分数
语文	99
数学	60
总分	159

← thead标签包裹的表头
 ← tbody标签包裹的表身
 ← tfoot标签包裹的表尾

表格第一行为表头数据，我们用 `<thead>` 标签包裹，中间的科目和分数为表格的主体内容，我们用 `<tbody>` 标签包裹，最后的总分我们用 `<tfoot>` 标签包裹。

解释：

- 1、`<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。
- 2、`<tbody>...</tbody>`：如果不加 `<thead>`、`<tbody>`、`<tfoot>`，表格加载完后才显示。加上这些表格结构，tbody 包含行的内容下载完优先显示，不必等待表格结束后再显示，同时如果表格很长，用 tbody 分段，可以一部分一部分地显示。（通俗理解表格可以按结构一块块的显示，不在等整个表格加载完后显示。）
- 3、`<tfoot>` 元素用于对 HTML 表格中的表注（页脚）内容进行分组。
- 4、`thead`、`tfoot` 以及 `tbody` 元素使您有能力对表格中的行进行分组。当您创建某个表格时，您也许希望拥有一个标题行，一些带有数据的行，以及位于底部的一个总计行。这种划分使浏览器有能力支持独立于表格标题和页脚的表格正文滚动。当长的表格被打印时，表格的表头和页脚可被打印在包含表格数据的每张页面上。

6 HTML5 表单标签，与浏览器交互

• 6-1 使用 `<form>` 创建表单

网站怎样与用户进行交互？答案是使用 HTML 表单 (form)。表单是可以把浏览器输入的数据传送到服务器端，这样服务器端程序就可以处理表单传过来的数据。

语法：

```
<form method=" 传送方式" action=" 服务器文件">
```

讲解：

1. `<form>` : `<form>` 标签是成对出现的，以 `<form>` 开始，以 `</form>` 结束
2. `<action>` : 浏览者输入的数据被传送到地方，比如一个 PHP 页面 (save.php)
3. `<method>` : 数据传送的方式 (get/post)

```
<form method="post" action="save.php">
  <label for="username">用户名:</label>
  <input type="text" name="username" id="username" value="" />
  <label for="pass">密码:</label>
  <input type="password" name="pass" id="pass" value="" />
  <input type="submit" value=" 确定" name="submit" />
  <input type="reset" value=" 重置" name="reset" />
</form>
```

用户名: 密码:

注意：

1. 所有表单控件（文本框、文本域、按钮、单选框、复选框等）都必须放在 `<form></form>` 标签之间（否则用户输入的信息可提交不到服务器上）。
2. method : post/get 的区别这一部分内容属于后端程序员考虑的问题。

• 6-2 文本输入框、密码输入框

当用户要在表单中键入字母、数字等内容时，就会用到文本输入框。文本框也可以转化为密码输入框。

语法：

```
<form>
  <input type="text/password" name=" 名称" value=" 文本" />
</form>
```

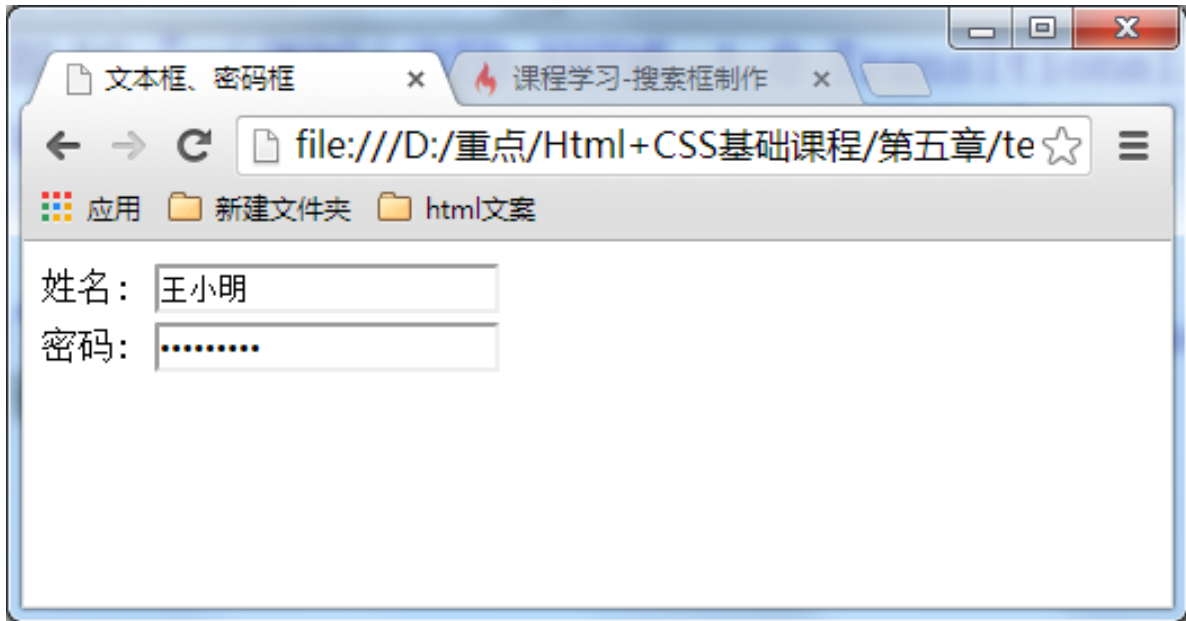
1. type:
当 type="text" 时，输入框为文本输入框；
当 type="password" 时，输入框为密码输入框。
2. name: 为文本框命名，以备后台程序 ASP、PHP 使用。
3. value: 为文本输入框设置默认值。（一般起到提示作用）

举例：

```
<form>
  姓名: <input type="text" name="myName">
<br/>
```

```
密码: <input type="password" name="pass">
</form>
```

在浏览器中显示的结果:



- 6-3 placeholder 属性的使用

input 标签中占位符 placeholder 属性, 有时候需要提示用户输入框需要输入框的内容, 那么就会用到我们的占位符, 比如下面的效果:



想要实现这样的效果, 我们只需要输入以下代码:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>input-placeholder</title>
  </head>
  <body>
    <!--placeholder 属性为输入框设置占位符, 值为占位符的内容-->
    <input type="text" placeholder=" 请输入关键字">
  </body>
</html>
```

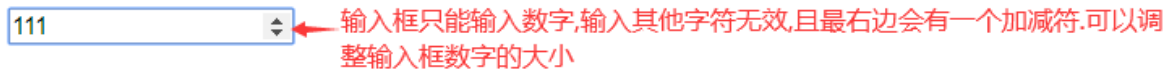
解释:

1. placeholder 属性为输入框占位符, 里面可以放提示的输入信息。

- placeholder 属性的值可以任意填写, 当输入框输入内容时, 占位符内容消失, 输入框无内容时, 占位符内容显示。
- 占位符内容不是输入框真正的内容。

- 6-4 数字输入框

input 标签中的数字框 number 类型, 先来看一下数字框:



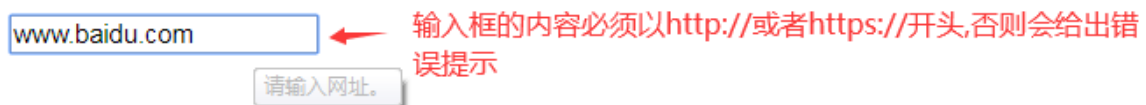
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>input-number</title>
  </head>
  <body>
    <!--把 input 的 type 属性设置为 number, 输入框的内容只能为数字, 且输入框右侧会有上下按钮进行加减-->
    <input type="number">
  </body>
</html>
```

解释:

- input 的 type 属性设置为 number, 则表示该输入框的类型为数字。
- 数字框只能输入数字, 输入其他字符无效。
- 数字框最右侧会有一个加减符号, 可以调整输入数字的大小, 不同浏览器表现不一致。

- 6-5 网址输入框

input 标签中的网址框 url 类型, 先来看一下网址框:



```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>input-url</title>
  </head>
  <body>
    <!--把 input 的 type 属性设置为 url, 输入框的内容必须以 http://或者 https://开头-->
    <input type="url">
  </body>
</html>
```

解释:

- input 的 type 属性设置为 url, 则表示该输入框的类型为网址。
- 数字框的值需以 http:// 或者 https:// 开头, 且后面必须有内容, 否则表单提交的时候会报错误提示。

- 6-6 邮箱输入框

input 标签中的邮箱框 email 类型，先来看一下邮箱框：



```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>input-url</title>
  </head>
  <body>
    <!--把 input 的 type 属性设置为 email，输入框的内容必须要包含 @，且 @ 后面必须有内容-->
    <input type="email">
  </body>
</html>
```

解释：

1. Input 的 type 属性设置为 email，则表示该输入框的类型为邮箱。
 2. 数字框的值必须包含 @。
 3. 数字框的值 @ 之后必须有内容，否则会报错误提示。
- 6-7 使用 <textarea> 标签创建文本域

当用户需要在表单中输入大段文字时，需要用到文本输入域。

语法：

```
<textarea rows=" 行数" cols=" 列数">文本</textarea>
```

1. <textarea> 标签是成对出现的，以 <textarea> 开始，以 </textarea> 结束。
2. cols: 多行输入域的列数。
3. rows: 多行输入域的行数。
4. 在 <textarea></textarea> 标签之间可以输入默认值。

举例：

```
<form method="post" action="save.php">
  <label>联系我们</label>
  <textarea cols="50" rows="10">在这里输入内容...</textarea>
</form>
```

在浏览器中显示结果：



注意：这两个属性可用 css 样式的 width 和 height 来代替；col 用 width、row 用 height 来代替。

- 6-8 使用 `<label>` 标签

`label` 标签不会向用户呈现任何特殊效果，它的作用是为鼠标用户改进了可用性。如果你在 `label` 标签内点击文本，就会触发此控件。就是说，当用户单击选中该 `label` 标签时，浏览器就会自动将焦点转到和标签相关的表单控件上（就自动选中与该 `label` 标签相关连的表单控件上）。

语法：

```
<label for=" 控件 id 名称">
```

注意：标签的 `for` 属性中的值应当与相关控件的 `id` 属性值一定要相同。

例子：

```
<form>
  <label for="uname">输入你的用户名</label>
  <input type="text" id="uname" placeholder="Enter username">
  <br>
  <label for="pass">输入你的密码</label>
  <input type="password" id="pass" placeholder="Enter password">
</form>
```

直观的效果是：当点击被 `label` 所框起来的文字（e.g. “输入你的用户名”）时，鼠标光标会自动移至其对应的输入框。

- 6-9 单选框、复选框，让用户选择

在使用表单设计调查表时，为了减少用户的操作，使用选择框是一个好主意，html 中有两种选择框，即单选框和复选框，两者的区别是单选框中的选项用户只能选择一项，而复选框中用户可以任意选择多项，甚至全选。

语法：

```
<input type="radio/checkbox" value=" 值" name=" 名称" checked="checked"/>
```


1. type:

当 type="radio" 时，控件为单选框

当 type="checkbox" 时，控件为复选框

2. value: 提交数据到服务器的值（后台程序 PHP 使用）

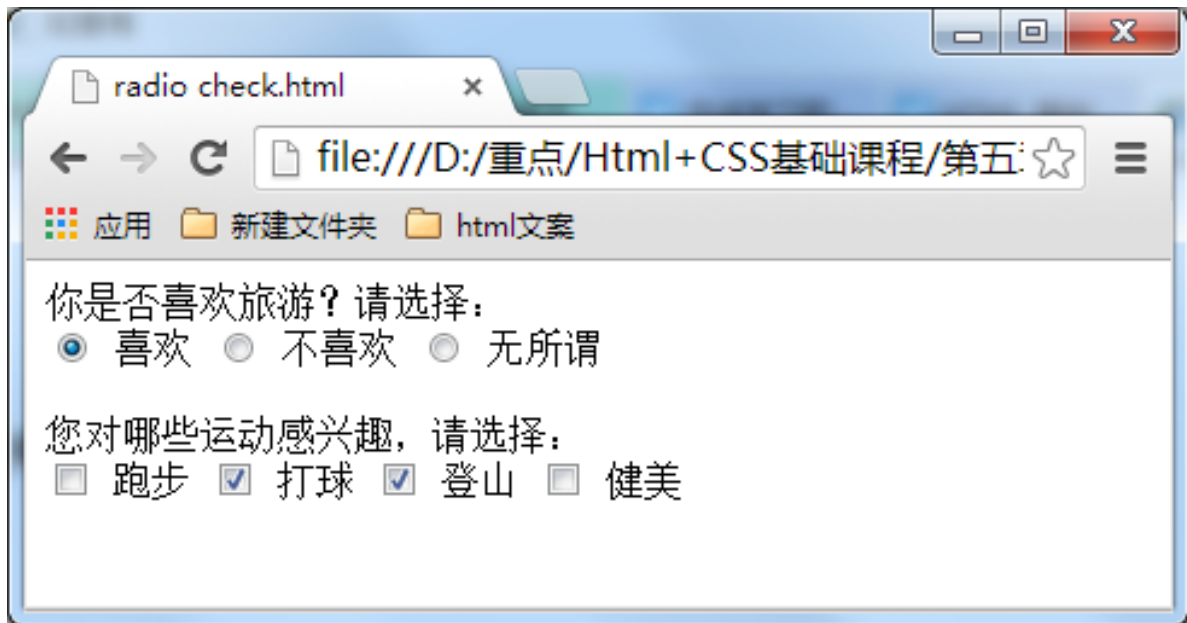
3. name: 为控件命名，以备后台程序 ASP、PHP 使用

4. checked: 当设置 checked="checked" 时，该选项被默认选中

如下面代码：

```
<form name="iForm" method="post" action="save.php">
    你是否喜欢旅游? <br />
    <input type="radio" name="radioLove" value=" 喜欢" checked="checked"> 喜欢
    <input type="radio" name="radioLove" value=" 不喜欢"> 不喜欢
    <input type="radio" name="radioLove" value=" 无所谓"> 无所谓
    <br /><br />
    你对哪些运动感兴趣? <br />
    <input type="checkbox" name="checkbox1" value=" 跑步"> 跑步
    <input type="checkbox" name="checkbox2" value=" 打球" checked="checked"> 打球
    <input type="checkbox" name="checkbox3" value=" 登山" checked="checked"> 登山
    <input type="checkbox" name="checkbox4" value=" 健身"> 健身
</form>
```

在浏览器中显示的结果：



注意：同一组的单选按钮，name 取值一定要一致，比如上面例子为同一个名称“radioLove”，这样同一组的单选按钮才可以起到单选的作用。

- 6-10 使用 select 、 option 标签创建下拉菜单

下拉列表在网页中也常会用到，它可以有效的节省网页空间。既可以单选、又可以多选。如下代码：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
```

```

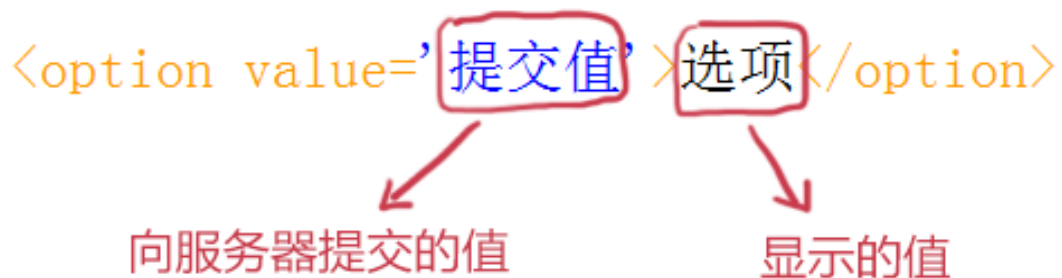
<title>select 下拉框</title>
</head>
<body>
  <form>
    <select>
      <option value=" 看书">看书</option>
      <option value=" 旅游">旅游</option>
      <option value=" 运动">运动</option>
      <option value=" 购物" selected="selected">购物</option>
    </select>
  </form>
</body>
</html>

```

讲解：

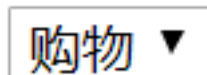
1. select 和 option 标签都是双标签，它总是成对出现的，需要首标签和尾标签。
2. select 标签里面只能放 option 标签，表示下拉列表的选项。
3. option 标签放选项内容，不放置其他标签。
4. value:

`<option value='提交值'>选项</option>`



向服务器提交的值 显示的值

5. selected="selected": 设置 selected="selected" 属性，则该选项就被默认选中。在浏览器中显示的结果：



• 6-11 提交按钮

在表单中有两种按钮可以使用，分别为：提交按钮、重置。这一小节讲解提交按钮：当用户需要提交表单信息到服务器时，需要用到提交按钮。

语法：

```
<input type="submit" value=" 提交">
```

type: 只有当 type 值设置为 submit 时，按钮才有提交作用。value: 按钮上显示的文字

举例：

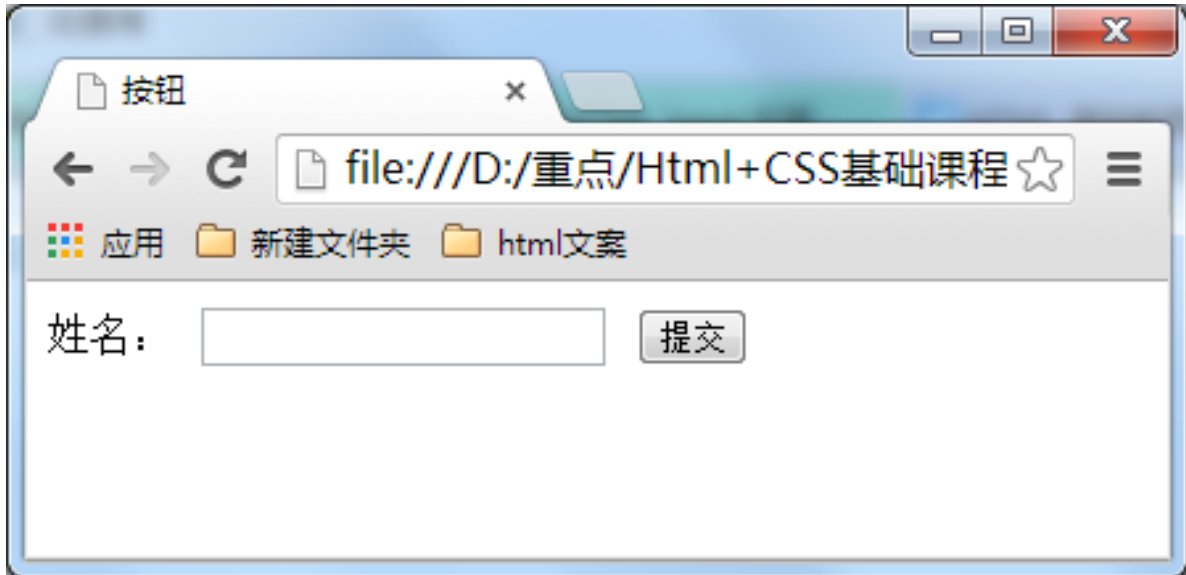
```

<form method="post" action="save.php">
  <label for="myName">姓名: </label>

```

```
<input type="text" value=" " name="myName" />
<input type="submit" value=" 提交 " name="submitBtn" />
</form>
```

在浏览器中显示的结果：



- 6-12 使用重置按钮，重置表单信息

当用户需要重置表单信息到初始时的状态时，比如用户输入“用户名”后，发现书写有误，可以使用 **重置按钮** 使输入框恢复到初始状态。只需要把 type 设置为“reset”就可以。

语法：

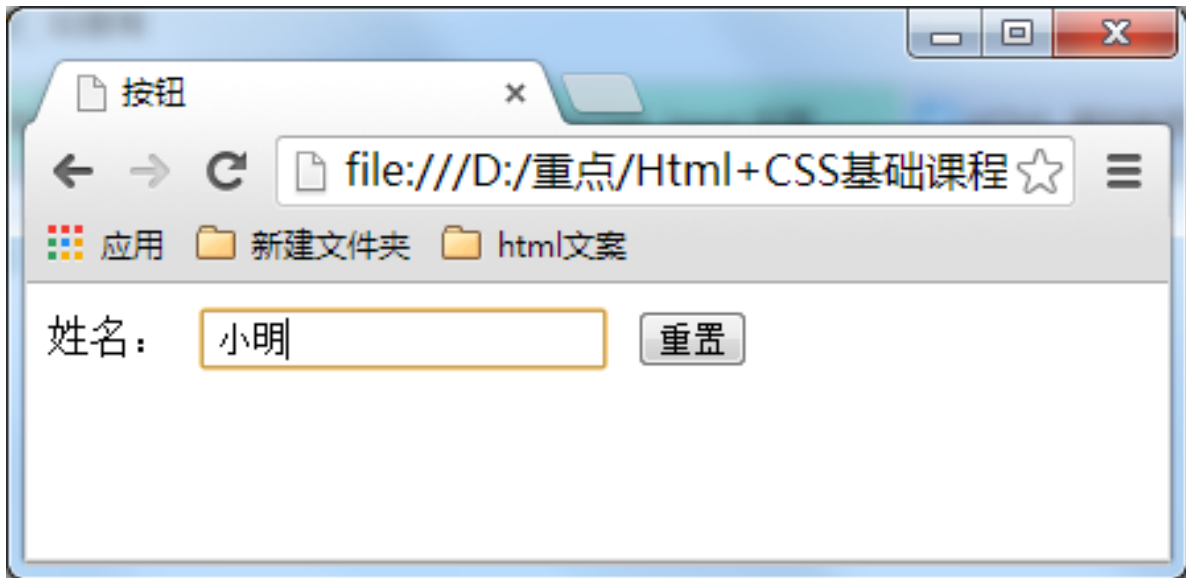
```
<input type="reset" value=" 重置">
```

type: 只有当 type 值设置为 reset 时，按钮才有提交作用。value: 按钮上显示的文字

举例：

```
<form method="post" action="save.php">
  <label for="myName">姓名: </label>
  <input type="text" value=" " name="myName" />
  <input type="reset" value=" 重置 " name="resetBtn" />
</form>
```

在浏览器中显示的结果：



```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>重置按钮</title>
  </head>
  <body>
    <form action="save.php" method="post">
      <label>爱好:</label>
      <select>
        <option value=" 看书">看书</option>
        <option value=" 旅游" selected="selected">旅游</option>
        <option value=" 运动">运动</option>
        <option value=" 购物">购物</option>
      </select>
      <input type="submit" value=" 确定" />
      <input type="reset" value=" 重置" />
    </form>
  </body>
</html>
```

爱好:

7 CSS 介绍，为网页添加样式

• 7-1 认识 CSS 样式

CSS 全称为“层叠样式表 (Cascading Style Sheets)”，它主要是用于定义 HTML 内容在浏览器内的显示样式，如文字大小、颜色、字体加粗等。

如下列代码：

```
p{
  font-size:12px;
```

```
color:red;
font-weight:bold;
}
```

使用 CSS 样式的一个好处是通过定义某个样式，可以让不同网页位置的文字有着统一的字体、字号或者颜色等。

• 7-2 CSS 样式的优势

为什么使用 CSS 样式来设置网页的外观样式呢？一段文字，我们想把“超酷的互联网”、“服务及时贴心”、“有趣易学”这三个短语的文本颜色设置为红色，这时就可以通过设置样式来设置，而且只需要编写一条 CSS 样式语句。

第一步：把这三个短语用 `` 括起来；

第二步：写入下列代码：

```
span{
    color:red;
}

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>CSS 样式的优势</title>
    <style type="text/css">
      span {
        color: red;
      }
    </style>
  </head>
  <body>
    <p>慕课网，<span>超酷的互联网</span>、IT 技术免费学习平台，创新的网络一站式学习、实践体验；<span>服务及时贴心</span></p>
  </body>
</html>
```

• 7-3 CSS 代码语法

CSS 样式由选择符和声明组成，而声明又由属性和值组成，如下图所示：



选择符：又称选择器，指明网页中要应用样式规则的元素，如本例中是网页中所有的段（p）的文字将变成蓝色，而其他的元素（如 ol）不会受到影响。

声明：在英文大括号“{}”中的就是声明，属性和值之间用英文冒号“:”分隔。当有多条声明时，中间可以英文分号“;”分隔，如下所示：

```
p{font-size:12px;color:red;}
```

注意：

1. 最后一条声明可以没有分号，但是为了以后修改方便，一般也加上分号。
2. 为了使用样式更加容易阅读，可以将每条代码写在一个新行内，如下所示：

```
p{
    font-size:12px;
    color:red;
}
```

- 7-4 CSS 注释代码

就像在 Html 的注释一样，在 CSS 中也有注释语句：用 `/* 注释语句 */` 来标明（Html 中使用 `<!--注释语句-->`）。就像下面代码：

```
p {
    /* 设置文字字号为 12px*/
    font-size: 12px;
    /* 设置文字颜色为红色 */
    color: red;
}
```

- 7-5 内联式 CSS 样式

从 CSS 样式代码插入的形式来看基本可以分为以下 3 种：内联式、嵌入式和外部式三种。

内联式 CSS 样式表就是把 CSS 代码直接写在现有的 HTML 标签中，如下面代码：

```
<p style="color:red">这里文字是红色。</p>
```

注意要写在元素的开始标签里，下面这种写法是错误的：

```
<p>这里文字是红色。</p style="color:red">
```

并且 CSS 样式代码要写在 `style=""` 双引号中，如果有多条 CSS 样式代码设置可以写在一起，中间用分号隔开。如下代码：

```
<p style="color:red;font-size:12px"> 这里文字是红色。</p>
```

- 7-6 嵌入式 CSS 样式

嵌入式 CSS 样式，就是可以把 CSS 样式代码写在 `<style type="text/css"></style>` 标签之间。如下面代码实现把三个 `` 标签中的文字设置为红色：

```
<style type="text/css">
    span{
        color:red;
    }
</style>
```

嵌入式 CSS 样式必须写在 `<style></style>` 之间，并且一般情况下嵌入式 CSS 样式写在 `<head></head>` 之间。如右边编辑器中的代码。

- 7-7 外部式 CSS 样式

外部式 CSS 样式（也可称为外联式）就是把 CSS 代码写一个单独的外部文件中，这个 CSS 样式文件以“`.css`”为扩展名，在 `<head>` 内（不是在 `<style>` 标签内）使用 `<link>` 标签将 CSS 样式文件链接到 HTML 文件内，如下面代码：

```
<link href="**base.css**" rel="stylesheet" type="text/css" />
```

注意：

1. CSS 样式文件名称以有意义的英文字母命名，如 `main.css`。
2. `rel="stylesheet" type="text/css"` 是固定写法不可修改。

3. `<link>` 标签位置一般写在 `<head>` 标签之内。
- 7-8 三种链接方式的优先级

对于同一个元素我们同时用了三种方法设置 CSS 样式，那么哪种方法真正有效呢？

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>css3 中三种引入样式优先级</title>
    <link href="style.css" rel="stylesheet" type="text/css">
    <style type="text/css">
      span {
        color: red;
      }
    </style>
  </head>
  <body>
    <p>慕课网, <span style="color: pink">超酷的互联网</span>、IT 技术免费学习平台, 创新的网络一站式学习、实践体验
  </body>
</html>

/*style.css*/
span{
  color: blue;
}
```

在上面这种情况：

1. 使用 内联式 CSS 设置“超酷的互联网”文字为 粉色 。
2. 然后使用 嵌入式 CSS 来设置文字为 红色 。
3. 最后又使用 外部式 设置文字为 蓝色 （style.css 文件中设置）。

但最终你可以观察到“超酷的互联网”这个短词的文本被设置为了 粉色 。因为这三种样式是有优先级的，记住他们的优先级： 内联式 > 嵌入式 > 外部式

但是嵌入式 > 外部式有一个前提：嵌入式 CSS 样式的位置一定在外部式的后面。如上面代码就是这样，`<link href="style.css" ...>` 代码在 `<style type="text/css">...</style>` 代码的前面（实际开发中也是这么写的）。感兴趣的小伙伴可以试一下，把它们调换顺序，再看他们的优先级是否变化。

其实总结来说，就是 就近原则 （离被设置元素越近优先级别越高）。

但注意上面所总结的优先级是有一个前提：内联式、嵌入式、外部式样式表中 CSS 样式是在的相同权值的情况下。

8 CSS3 选择器

- 8-1 什么是选择器？

每一条 CSS 样式声明（定义）由两部分组成，形式如下：

```
选择器 {
  样式;
}
```

在 `{}` 之前的部分就是“选择器”，“选择器”指明了 `{}` 中的“样式”的作用对象，也就是“样式”作用于网页中的哪些元素。比如下方代码中“body”就是选择器。

```
body {
    font-size: 12px;
    color: red;
}
```

- 8-2 标签选择器

标签选择器其实就是 html 代码中的标签。如下代码中的 `<html>` 、 `<body>` 、 `<h1>` 、 `<p>` 、 `` 。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>标签选择器</title>
    <style type="text/css">
      h1{
        font-weight:normal;
        color:red;
      }
    </style>
  </head>

  <body>
    <h1>勇气</h1>
    <p>三年级时，我...</p>
    <p>到了三年级下学期时，我...</p>
  </body>
</html>
```

- 8-3 类选择器

类选择器在 CSS 样式编码中最常用到的，如：可以实现为“胆小如鼠”、“勇气”字体设置为红色。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>认识 html 标签</title>
    <style type="text/css">
      .stress {
        color: red;
      }
      .class {
        color: green;
      }
    </style>
  </head>

  <body>
    <h1>勇气</h1>
    <p>三年级时，我还是一个<span class="stress">胆小如鼠</span>的小女孩，就一直没有这个<span class="stress">勇气</span>。</p>
    <p>到了三年级下学期时，我们班上了一节<span class="class">公开课</span>，。。。</p>
  </body>
</html>
```

语法：


```
.类选择器名称 {css 样式代码;}
```

注意：

1. 英文圆点开头
2. 其中类选择器名称可以任意起名（但不要起中文噢）

使用方法：

第一步：使用合适的标签把要修饰的内容标记起来，如下：

```
<span>胆小如鼠</span>
```

第二步：使用 class=“类选择器名称” 为标签设置一个类，如下：

```
<span class="stress">胆小如鼠</span>
```

第三步：设置类选择器 css 样式，如下：

```
.stress{color:red;}/ * 类前面要加入一个英文圆点 */
```

- 8-4 ID 选择器

```
<body>
  <!--给元素 id 属性赋值-->
  <div id="box">一个 div</span>
</body>
```

```
<style>
  #box {
    color: red;
  }
</style>
```

解释：

1. 使用 ID 选择器，必须给标签添加上 id 属性，为标签设置 id=“ID 名称”，而不是 class=“类名称”。
2. ID 选择符的前面是井号（#）号，而不是英文圆点（.）。
3. id 属性的值既为当前标签的 id，尽量见名思意，语义化。

如下代码：

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>认识 html 标签</title>
    <style type="text/css">
      #stress {
        color: red;
      }
      #class {
        color: green;
      }
    </style>
  </head>

  <body>
    <h1>勇气</h1>
```

```

    <p>三年级时，我还是一个<span id="stress">胆小如鼠</span>的小女孩，就一直没有这个勇气来回答老师提出的问题。学校举
    <p>到了三年级下学期时，我们班上了一节<div id="class">公开课</div>，...</p>
  </body>
</html>

```

上面代码中 `<div id="class"> 公开课 </div>` 会报错,见下一节。而 ` 胆小如鼠 ` , 是因为 `p` 标签中一般不要用 `div`, `div` 等级比较特殊, 容易发生错误, 但依旧使用的 `id` 。

`<div>` 和 `` 这两个有什么区别?

`div` 是块级元素, 默认占一整行, 可以设置宽高,`span` 是行内元素, 默认在同一行排列, 不能设置宽高, 宽高是由 `span` 标签内容撑开的。

• 8-5 类和 ID 选择器的区别

学习了类选择器和 ID 选择器, 我们会发现他们之间有很多的相似处, 是不是两者可以通用呢? 我们不要着急先来总结一下他们的相同点和不同点:

相同点: 可以应用于任何元素 不同点:

1. **ID** 选择器只能在文档中使用一次。与类选择器不同, 在一个 HTML 文档中, ID 选择器只能使用一次, 而且仅一次。而类选择器可以使用多次。

下面代码是正确的:

```

<p>三年级时，我还是一个<span class="stress">胆小如鼠</span>的小女孩，就一直没有这个<span class="stress">勇气</span>

```

而下面代码是错误的:

```

<p>三年级时，我还是一个<span id="stress">胆小如鼠</span>的小女孩，就一直没有这个<span id="stress">勇气</span>

```

2. 可以使用类选择器词列表方法为一个元素同时设置多个样式。我们可以为一个元素同时设多个样式, 但只可以用类选择器的方法实现, ID 选择器是不可以的 (不能使用 **ID** 词列表)。

下面的代码是正确的

```

.stress{
  color:red;
}
.bigszie{
  font-size:25px;
}
<p>到了<span class="stress bigszie">三年级</span>下学期时，我们班上了一节公开课...</p>

```

上面代码的作用是为“三年级”三个文字设置文本颜色为红色并且字号为 25px。

下面的代码是不正确的

```

#stressid{
  color:red;
}
#bigszieid{
  font-size:25px;
}
<p>到了<span id="stressid bigszieid">三年级</span>下学期时，我们班上了一节公开课...</p>

```

上面代码不可以实现为“三年级”三个文字设置文本颜色为红色并且字号为 25px 的作用。

• 8-6 子选择器

还有一个比较有用的选择器子选择器, 即大于符号 (`>`), 用于选择指定标签元素的第一代子元素。如:

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>子选择器</title>
    <style type="text/css">
      /* 添加边框样式（粗细为 1px，颜色为红色的实线） */
      .first>span {
        border: 1px solid red;
      }
      /* 使 class 名为 food 下的子元素 li（水果、蔬菜）加入红色实线边框 */
      .food>li {
        border: 1px solid red;
      }
    </style>
  </head>

  <body>
    <p class="first">三年级时，<span>我还是一个<span>胆小如鼠</span>的小女孩</span>，就一直没有这个勇气来回答老
    <h1>食物</h1>
    <ul class="food">
      <li>水果
        <ul>
          <li>香蕉</li>
          <li>苹果</li>
          <li>梨</li>
        </ul>
      </li>
      <li>蔬菜
        <ul>
          <li>白菜</li>
          <li>油菜</li>
          <li>卷心菜</li>
        </ul>
      </li>
    </ul>
  </body>
</html>

```

效果如图：

三年级时，我还是一个胆小如鼠的小女孩，上课从来不敢回答老师提出的问题，生怕回答错了老师会批评我。就一直没有这个勇气来回答老师提出的问题。学校举办的活动我也没勇气参加。

食物

- 水果
 - 香蕉
 - 苹果
 - 梨
- 蔬菜
 - 白菜
 - 油菜
 - 卷心菜

- 8-7 后代选择器

包含选择器，即加入空格，用于选择指定标签元素下的后辈元素。如：

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>子选择器</title>
    <style type="text/css">
      /* 添加边框样式（粗细为 1px，颜色为红色的实线） */
      .first span {
        border: 1px solid red;
      }
      /* 使 class 名为 food 下的子元素 li（水果、蔬菜）加入红色实线边框 */
      .food li {
        border: 1px solid red;
      }
    </style>
  </head>

  <body>
    <p class="first">三年级时，<span>我还是一个<span>胆小如鼠</span>的小女孩</span>，就一直没有这个勇气来回答老
    <h1>食物</h1>
    <ul class="food">
      <li>水果
        <ul>
          <li>香蕉</li>
          <li>苹果</li>
          <li>梨</li>
        </ul>
      </li>
```

```

        <li>蔬菜
            <ul>
                <li>白菜</li>
                <li>油菜</li>
                <li>卷心菜</li>
            </ul>
        </li>
    </ul>
</body>
</html>

```

效果如图:

三年级时，我还是一个胆小如鼠的小女孩，上课从来不敢回答老师提出的问题，生怕回答错了老师会批评我。就一直没有这个勇气来回答老师提出的问题。学校举办的活动我也没勇气参加。

- 水果
 - 香蕉
 - 苹果
 - 梨
- 蔬菜
 - 白菜
 - 油菜
 - 卷心菜

请注意这个选择器与子选择器的区别，子选择器（child selector）仅是指它的直接后代，或者你可以理解为作用于子元素的第一代后代。而后代选择器是作用于所有子后代元素。后代选择器通过空格来进行选择，而子选择器是通过“>”进行选择。

总结：> 作用于元素的第一代后代，空格作用于元素的所有后代。

• 8-8 通用选择器

通用选择器是功能最强大的选择器，它使用一个（*）号指定，它的作用是匹配 html 中所有标签元素，如：

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>* 选择符</title>
        <style type="text/css">
            * {
                color: red;
                font-size:20px;
            }
        </style>
    </head>

    <body>
        <h1>勇气</h1>
    </body>
</html>

```

```
<p>三年级时，我还是一个胆小如鼠的小女孩，一直没有这个勇气来回答老师提出的问题。学校举办的活动我也没勇气参加。</p>
</body>
</html>
```

勇气

三年级时，我还是一个胆小如鼠的小女孩，上课从来不敢回答老师提出的问题，生怕回答错了老师会批评我。就一直没有这个勇气来回答老师提出的问题。学校举办的活动我也没勇气参加。

到了三年级下学期时，我们班上了一节公开课，老师提出了一个很简单的问题，班里很多同学都举手了，甚至成绩比我差很多的，也举手了，还说着："我来，我来。"我环顾了四周，就我没有举手。

- 8-9 伪类选择器

更有趣的是伪类选择符，为什么叫做伪类选择符，它允许给 html 不存在的标签（标签的某种状态）设置样式，比如说我们给 html 中一个标签元素的鼠标滑过的状态来设置字体颜色：

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>伪类选择符</title>
    <style type="text/css">
      a:hover {
        color: red;
        font-size: 20px;
      }
    </style>
  </head>

  <body>
    <h1>勇气</h1>
    <p class="first">三年级时，我还是一个<a>胆小如鼠</a>的小女孩，一直没有这个勇气来回答老师提出的问题。学校举办的活
  </body>
</html>
```

上面一行代码就是为 a 标签鼠标滑过的状态设置字体颜色变红。这样就会使第一段文字内容中的“胆小如鼠”文字加入鼠标滑过字体颜色变为红色特效。

关于伪选择符：

关于伪类选择符，到目前为止，可以兼容所有浏览器的“伪类选择符”就是 a 标签上使用: hover 了（其实伪类选择符还有很多，尤其是 css3 中，但是因为不能兼容所有浏览器，本教程只是讲了这一种最常用的）。其实: hover 可以放在任意的标签上，比如说 p: hover，但是它们的兼容性也是很不好的，所以现在比较常用的还是 a: hover 的组合。

- 8-10 分组选择器

当你想为 html 中多个标签元素设置同一个样式时，可以使用分组选择符 ()，如下代码为右侧代码编辑器中的 h1、span 标签同时设置字体颜色为红色：

```
h1,span{color:red;}
```

它相当于下面两行代码：

```
h1{color:red;}
span{color:red;}
```

要求：把第一段全部文字颜色设置为绿色，同时把第二段文字中的“简单”文字颜色设置为绿色：

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>分组选择符</title>
    <style type="text/css">
      .first,#second span{
        color: green;
      }
    </style>
  </head>

  <body>
    <h1>勇气</h1>
    <p class="first">三年级时，我还是一个<span>胆小如鼠</span>的小女孩，一直没有这个勇气来回答老师提出的问题。</p>
    <p id="second">到了三年级下学期时，我们班上了一节公开课，老师提出了一个很<span>简单</span>的问题，...</p>
  </body>
</html>
```

勇气

三年级时，我还是一个胆小如鼠的小女孩，上课从来不敢回答老师提出的问题，生怕回答错了老师会批评我。就一直没有这个勇气来回答老师提出的问题。学校举办的活动我也没勇气参加。

到了三年级下学期时，我们班上了一节公开课，老师提出了一个很简单的问题，班里很多同学都举手了，甚至成绩比我差很多的，也举手了，还说着："我来，我来。"我环顾了四周，就我没有举手。

9 CSS3 的继承，优先级和重要性

- 9-1 样式的继承

CSS 的某些样式是具有继承性的，那么什么是继承呢？继承是一种规则，它允许样式不仅应用于某个特定 html 标签元素，而且应用于其后代。比如下面代码：如某种颜色应用于 p 标签，这个颜色设置不仅应用 p 标签，还应用于 p 标签中的所有子元素文本，这里子元素为 span 标签。

```
p{color:red;}
```

```
<p>三年级时，我还是一个<span>胆小如鼠</span>的小女孩。</p>
```

可见 p 中的文本与 span 中的文本都设置为了红色。但注意有一些 css 样式是不具有继承性的。如 border:1px solid red; p{border:1px solid red;}

```
<p>三年级时，我还是一个<span>胆小如鼠</span>的小女孩。</p>
```

在上面例子中它代码的作用只是给 p 标签设置了边框为 1 像素、红色、实心边框线，而对于子元素 span 是没起到作用的。

- 9-2 选择器的优先级

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>css3 选择器优先级</title>
    <style type="text/css">
      /* 分别使用 id 选择器、类选择器、标签选择器、通配符选择器给 div 设置不同字体颜色 */
      #box {
        color:red;
      }
      .dv {
        color:blue;
      }
      div {
        color:black;
      }
      * {
        color:green;
      }
    </style>
  </head>

  <body>
    <!--给 div 分别设置了 id、class 属性，并写了行内样式-->
    <div id="box" class="dv" style="color:orange">
      我是一个 div
    </div>
  </body>
</html>
```

我是一个div ← 最终div呈现为橘黄色,对应行内样式的字体颜色

解释：

1. 如果一个元素使用了多个选择器，则会按照选择器的优先级来给定样式。
2. 选择器的优先级依次是：内联样式 > id 选择器 > 类选择器 > 标签选择器 > 通配符选择器

- 9-3 权值计算-特殊性

有的时候我们为同一个元素设置了不同的 CSS 样式代码，那么元素会启用哪一个 CSS 样式呢？下面我们一起来看一下代码：


```
p{color:red;}
.first{color:green;}
<p class="first">三年级时，我还是一个<span>胆小如鼠</span>的小女孩。</p>
```

p 和 .first 都匹配到了 p 这个标签上，那么会显示哪种颜色呢？green 是正确的颜色，那么为什么呢？是因为浏览器是根据权值来判断使用哪种 css 样式的，权值高的就使用哪种 css 样式。

下面是权值的规则：

标签的权值为 1，类选择符的权值为 10，ID 选择符的权值最高为 100。例如下面的代码：

```
p{color:red;} /* 权值为 1*/
p span{color:green;} /* 权值为 1+1=2*/
.warning{color:white;} /* 权值为 10*/
p span.warning{color:purple;} /* 权值为 1+1+10=12*/
#footer .note p{color:yellow;} /* 权值为 100+10+1=111*/
```

注意：还有一个权值比较特殊—继承也有权值但很低，有的文献提出它只有 0.1，所以可以理解为继承的权值最低。

要求：为“胆小如鼠”这几个文本设置权值更高的 CSS 样式代码来覆盖以前的 CSS 样式代码

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>特殊性</title>
    <style type="text/css">
      p{color:red;}
      .first{color:green;}/* 因为权值高显示为绿色 */
      span{color:pink;}/* 设置为粉色 */
      p span{color:purple;}
      /* p>span{color:purple;} 也可 */
    </style>
  </head>
  <body>
    <h1>勇气</h1>
    <p class="first">三年级时，我还是一个<span>胆小如鼠</span>的小女孩，一直没有这个勇气来回答老师提出的问题。</p>
  </body>
</html>
```

解释：

虽然 .first{color:green;} 权值很高，但对于“胆小如鼠”这几个文本，只是和其存在继承关系。但继承的权值很低，除非元素本身没有样式，否则可以忽略继承来的样式，当多个样式作用于同一个元素的时候。这里 p span{color:purple;} 才是直接作用于“胆小如鼠”这个元素的。

• 9-4 选择器最高层级!important

我们在做网页代码的时，有些特殊的情况需要为某些样式设置具有最高权值，怎么办？这时候我们可以使用!important 来解决。

如下代码：

```
p{color:red!important;}
p{color:green;}
<p class="first">三年级时，我还是一个<span>胆小如鼠</span>的小女孩。</p>
```

这时 p 段落中的文本会显示的 red 红色。

注意：!important 要写在分号的前面

这里注意当网页制作者不设置 css 样式时，浏览器会按照自己的一套样式来显示网页。并且用户也可以在浏览器中设置自己习惯的样式，比如有的用户习惯把字号设置为大一些，使其查看网页的文本更加清楚。这时注意样式优先级为：浏览器默认的样式 < 网页制作者样式 < 用户自己设置的样式，但记住!important 优先级样式是个例外，权值高于用户自己设置的样式。

10 CSS3 字体样式

- 10-1 使用 font-family 设置字体系列

我们可以使用 css 样式为网页中的文字设置字体、字号、颜色等样式属性。下面我们来看一个例子，下面代码实现：为网页中的文字设置字体为宋体。

```
body{font-family:" 宋体";}
```

这里注意不要设置不常用的字体，因为如果用户本地电脑上如果没有安装你设置的字体，就会显示浏览器默认的字体。（因为用户是否可以看到你设置的字体样式取决于用户本地电脑上是否安装你设置的字体。）现在一般网页喜欢设置“微软雅黑”，如下代码：

```
body{font-family:"Microsoft Yahei";}
```

或

```
body{font-family:" 微软雅黑";}
```

注意：第一种方法比第二种方法兼容性更好一些。

因为这种字体即美观又可以在客户端安全的显示出来（用户本地一般都是默认安装的）。

- 10-2 使用 font-size 设置字体大小

可以使用下面代码设置网页中文字的字号为 12 像素：

```
body{font-size:12px;}
```

- 10-3 使用 font-weight 设置字体粗细

我们还可以使用 css 样式来改变文字的样式：粗体、斜体、下划线、删除线，可以使用下面代码实现设置文字以粗体样式显示出来。

```
p span{font-weight:bold;}
```

在这里大家可以看到，如果想为文字设置粗体是有单独的 css 样式来实现的，再不用为了实现粗体样式而使用 h1-h6 或 strong 标签了。

- 10-4 使用 font-style 设置字体样式

我们来学习设置字体样式 font-style 属性，比如我们想实现以下效果：

我是正常字体 ← 正常字体
我是斜体 ← 斜体
我是倾斜的字体 ← 倾斜的字体,与斜体有区别

如果实现以上效果，我们可以输入以下代码：

```
<body>
  <div class="box1">
    我是正常字体
  </div>
  <div class="box2">
    我是斜体
```

```

</div>
<div class="box3">
    我是倾斜的字体
</div>
</body>

<style type="text/css">
    .box1 {
        font-style: normal;
    }
    .box2 {
        font-style: italic;
    }
    .box3 {
        font-style: oblique;
    }
</style>

```

解释：

1. font-style 可以设置字体样式，并且有 3 种设置方式。
 2. 正常字体为 normal，也是 font-style 的默认值。
 3. italic 为设置字体为斜体，用于字体本身就有倾斜的样式。
 4. oblique 为设置倾斜的字体，强制将字体倾斜。
 5. italic 是字体系列（比如宋体，楷体）里的斜体，而 oblique 是字体倾斜效果，对于没有斜体的字体系列使用 oblique 实现斜体效果
- 10-5 使用 color 设置字体颜色

有时候当我们需要给字体设置颜色，比如实现以下效果：

我是box1
我是box2
我是box3

那么我们可以输入以下代码：

```

<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>>Document</title>
        <style type="text/css">
            #box1 {
                color:red;
            }
            #box2 {
                color:#abc123;
            }
            #box3 {
                color:rgb(12,34,56);
            }
        </style>
    </head>
    <body>
        <div id="box1">我是box1</div>
        <div id="box2">我是box2</div>
        <div id="box3">我是box3</div>
    </body>
</html>

```

```

    }
  </style>
</head>

<body>
  <div id='box1'>
    我是 box1
  </div>
  <div id='box2'>
    我是 box2
  </div>
  <div id='box3'>
    我是 box3
  </div>
</body>
</html>

```

解释：

1. color 属性可以设置字体颜色。
2. color 的值有 3 种设置方式：

— 英文命令颜色

```
p{color:red;}
```

— RGB 颜色

这个与 photoshop 中的 RGB 颜色是一致的，由 R(red)、G(green)、B(blue) 三种颜色的比例来配色。

```
p{color:rgb(133,45,200);}
```

每一项的值可以是 0~255 之间的整数，也可以是 0%~100% 的百分数。如：

```
p{color:rgb(20%,33%,25%);}
```

— 十六进制颜色

这种颜色设置方法是现在比较普遍使用的方法，其原理其实也是 RGB 设置，但是其每一项的值由 0-255 变成了十六进制 00-ff。

```
p{color:#00ffff;}
```

- 10-6 font 样式的简写方式

网页中的字体 css 样式代码也有他自己的缩写方式，下面是给网页设置字体的代码：

```

body{
  font-style:italic;
  font-weight:bold;
  font-size:12px;
  line-height:1.5em;
  font-family:" 宋体",sans-serif;
}

```

这么多行的代码其实可以缩写为一句：

```

body{
  font:italic bold 12px/1.5em " 宋体",sans-serif;
}

```

注意：

1. 使用这一简写方式你至少要指定 font-size 和 font-family 属性，其他的属性（如 font-weight、font-style、font-variant、line-height）如未指定将自动使用默认值。
2. 在缩写时 font-size 与 line-height 中间要加入 “/” 斜杠。

一般情况下因为对于中文网站，英文还是比较少的，所以下面缩写代码比较常用：

```
body{  
    font:12px/1.5em "宋体",sans-serif;  
}
```

只是有字号、行间距、中文字体、英文字体设置。

11 CSS3 文本样式

- 11-1 使用 text-decoration 添加文本修饰

我们来学习文本装饰，比如我们想实现以下效果：

有一条线在我头上 ← 装饰线在文本顶部

有一条线在我脚底 ← 装饰线在文本底部

有一条线从我身体穿过 ← 装饰线从文本穿过

想要实现以上效果，我们可以输入以下代码：

```
<body>  
  <div class="box1">  
    有一条线在我头上  
  </div>  
  <div class="box2">  
    有一条线在我脚底  
  </div>  
  <div class="box3">  
    有一条线从我身体穿过  
  </div>  
</body>  
  
<style type="text/css">  
  .box1 {  
    text-decoration:overline;  
  }  
  .box2 {  
    text-decoration:underline;  
  }  
  .box3 {  
    text-decoration:line-through;  
  }  
</style>
```

解释：

1. text-decoration 可以设置添加到文本的修饰。

2. text-decoration 默认值为 none, 定义标准的文本。
 3. text-decoration 的值为 underline 为定义文本下的一条线。
 4. text-decoration 的值为 overline 为定义文本上的一条线。
 5. text-decoration 的值为 line-through 为定义穿过文本下的一条线, 一般用于商品折扣价。
- 11-2 使用 text-indent 为文本添加首行缩进
中文文字中的段前习惯空两个文字的空白, 这个特殊的样式可以用下面代码来实现:

```
p{text-indent:2em;}
```


注意: 2em 的意思就是文字的 2 倍大小。
 - 11-3 使用 line-height 为文字设置行间距
另一个在段落排版中起重要作用的行间距 (行高) 属性 (line-height), 如下代码实现设置段落行间距为 1.5 倍。

```
p{line-height:1.5em;}
```
 - 11-4 使用 letter/word-spacing 增加或减少字符间的空白
中文字间隔、字母间隔设置:
如果想在网页排版中设置文字间隔或者字母间隔就可以使用 letter-spacing 来实现, 如下面代码:

```
h1{letter-spacing:50px;}
```


注意: 这个样式使用在英文单词时, 是设置字母与字母之间的间距。
单词间距设置:
如果我想设置英文单词之间的间距呢? 可以使用 word-spacing 来实现。如下代码:

```
h1{word-spacing:50px;}
```
 - 11-5 使用 text-align 设置文本对齐方式
想为块状元素中的文本、图片设置居中样式吗? 可以使用 text-align 样式代码, 如下代码可实现文本居中显示。(那么什么是块状元素呢? 在后面的 12-1、12-2 小节中会讲到。)

```
h1{text-align:center;}
```


同样可以设置居左:

```
h1{text-align:left;}
```


还可以设置居右:

```
h1{text-align:right;}
```
 - 11-6 长度值
长度单位总结一下, 目前比较常用到 px (像素)、em、% 百分比, 要注意其实这三种单位都是相对单位。

1、像素

像素为什么是相对单位呢? 因为像素指的是显示器上的小点 (CSS 规范中假设 “90 像素 = 1 英寸”)。实际情况是浏览器会使用显示器的实际像素值有关, 在目前大多数的设计者都倾向于使用像素 (px) 作为单位。

2、em

就是本元素给定字体的 font-size 值, 如果元素的 font-size 为 14px, 那么 1em = 14px; 如果 font-size 为 18px, 那么 1em = 18px。如下代码:

```
p{font-size:12px;text-indent:2em;}
```

上面代码就是可以实现段落首行缩进 24px（也就是两个字体大小的距离）。

下面注意一个特殊情况：

但当给 font-size 设置单位为 em 时，此时计算的标准以父元素的 font-size 为基础。如下代码：

html:

```
<p>以这个<span>例子</span>为例。</p>
```

css:

```
p{font-size:14px}  
span{font-size:0.8em;}
```

结果 span 中的字体“例子”字体大小就为 11.2px (14 * 0.8 = 11.2px)。

3、百分比

```
p{font-size:12px;line-height:130%}
```

设置行高（行间距）为字体的 130% (12 * 1.3 = 15.6px)。

12 CSS3 盒模型

• 12-1 元素分类

在讲解 CSS 布局之前，我们需要提前知道一些知识，在 CSS 中，html 中的标签元素大体被分为三种不同的类型：块状元素、内联元素（又叫行内元素）和内联块状元素。

常用的块状元素有：

```
<div> 、 <p> 、 <h1> ... <h6> 、 <ol> 、 <ul> 、 <dl> 、 <table> 、 <address> 、 <blockquote>  
、 <form>
```

常用的内联元素有：

```
<a> 、 <span> 、 <br> 、 <i> 、 <em> 、 <strong> 、 <label> 、 <q> 、 <var> 、 <cite> 、 <code>
```

常用的内联块状元素有：

```
<img> 、 <input>
```

• 12-2 块级元素

什么是块级元素？在 html 中 <div> 、 <p> 、 <h1> ... <h6> 、 、 <form> 和 就是块级元素。设置 display:block 就是将元素显示为块级元素。如下代码就是将内联元素 a 转换为块状元素，从而使 a 元素具有块状元素特点。

```
a{display:block;}
```

块级元素特点：

1. 每个块级元素都从新的一行开始，并且其后的元素也另起一行。（真霸道，一个块级元素独占一行）
2. 元素的高度、宽度、行高以及顶和底边距都可设置。
3. 元素宽度在不设置的情况下，是它本身父容器的 100%（和父元素的宽度一致），除非设定一个宽度。

• 12-3 内联元素

在 html 中, ``、`<a>`、`<label>`、`` 和 `` 就是典型的内联元素 (行内元素) (inline) 元素。当然块状元素也可以通过代码 `display:inline` 将元素设置为内联元素。如下代码就是将块状元素 `div` 转换为内联元素, 从而使 `div` 元素具有内联元素特点。

```
div{display:inline;}
```

特点:

1. 和其他元素都在一行上;
2. 元素的高度、宽度及顶部和底部边距不可设置;
3. 元素的宽度就是它包含的文字或图片的宽度, 不可改变。

内联元素之间有一个间距问题: 使用 `letter/word-spacing` 增加或减少字符间的空白

- 12-4 内联块状元素

内联块状元素 (inline-block) 就是同时具备内联元素、块状元素的特点, 代码 `display:inline-block` 就是将元素设置为内联块状元素。(css2.1 新增), ``、`<input>` 标签就是这种内联块状标签。

inline-block 元素特点:

1. 和其他元素都在一行上;
2. 元素的高度、宽度、行高以及顶和底边距都可设置。

- 12-5 none 不占据位置

none 设置此元素不会被显示, 当想要元素隐藏的时候可以使用此值。

```
p{none;}
```

- 12-6 什么是盒模型

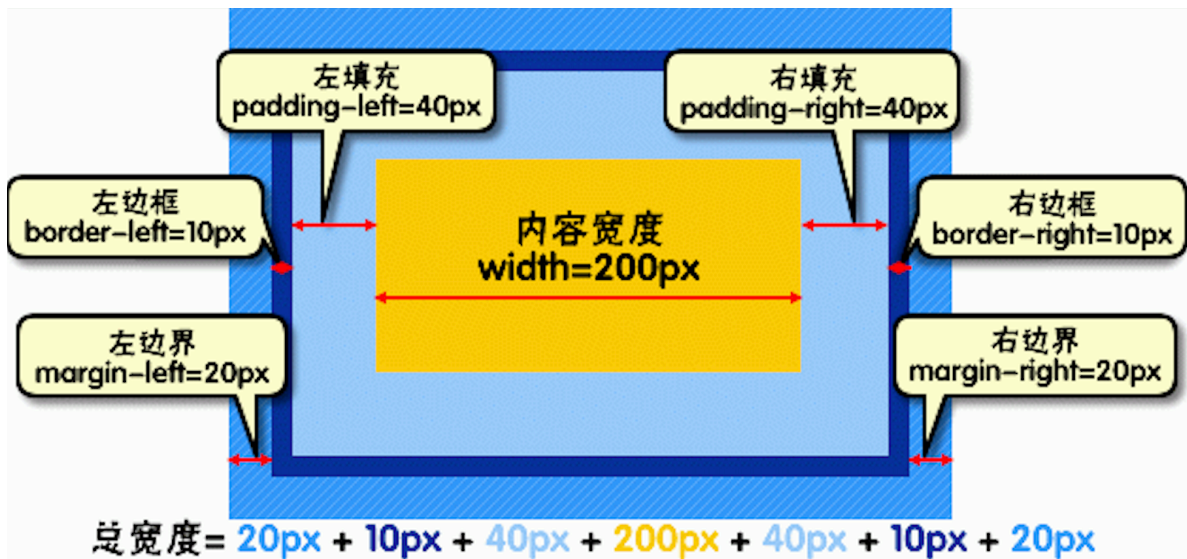
盒模型是一个页面元素, 比如 `<div>`, 其中包含内容, 内容可以为文字、图片或者另一个标签元素。内容和盒之间的距离称为内填充 (padding), 有四个方向, 分别为 `padding-top`, `padding-bottom`, `padding-left`, `padding-right`。两个盒模型之间的距离称为外边距 (margin)。盒子的边框称为 border。外边距和边框也有四个方向。内容的实际高度为其自身高度加上下内填充和上下边框, 宽度同理。

`<div>`、`<p>`、`<h>`、``、``、`<table>` 等块级标签都具备盒模型的特征。

- 12-7 宽度和高度

盒模型宽度和高度和我们平常所说的物体的宽度和高度理解是不一样的, css 内定义的宽 (width) 和高 (height), 指的是填充以里的内容范围。

因此一个元素实际宽度 (盒子的宽度) = 左边界 + 左边框 + 左填充 + 内容宽度 + 右填充 + 右边框 + 右边界。



元素的高度也是同理。

比如：

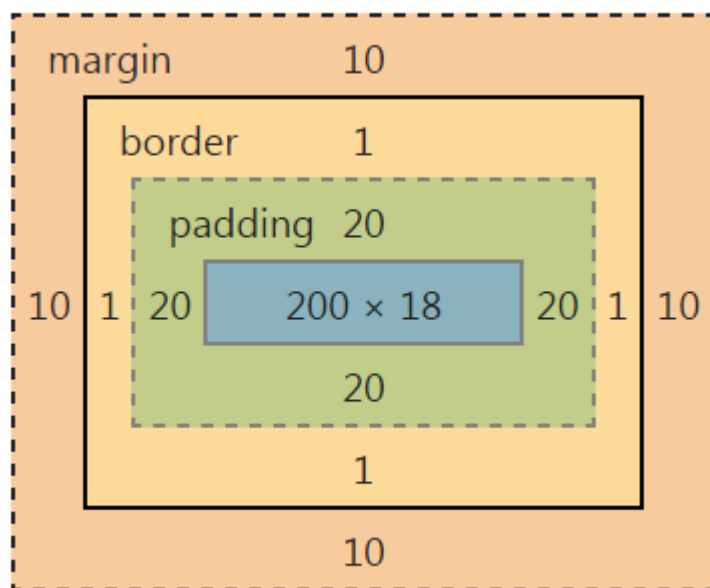
css 代码：

```
div{  
    width:200px;  
    padding:20px;  
    border:1px solid red;  
    margin:10px;  
}
```

html 代码：

```
<body>  
    <div>文本内容 </div>  
</body>
```

元素的实际长度为：10px+1px+20px+200px+20px+1px+10px=262px。在 chrome 浏览器下可查看元素盒模型，如下图：



- 12-8 背景色

网页中的标签不论是行内元素还是块状元素都可以给它设置一个背景色。

为标签设置背景颜色可以使 `background-color: 颜色值` 来实现。

例子如下：

```
div{background-color:red;}//为块状元素设置
a{background-color:green;}//为行内元素设置
```

- 12-9 使用 border 为盒子添加边框（一）

盒子模型的边框就是围绕着内容及补白的线，这条线你可以设置它的粗细、样式和颜色（边框三个属性）。

如下面代码为 div 来设置边框粗细为 2px、样式为实心的、颜色为红色的边框：

```
div{border:2px solid red;}
```

上面是 border 代码的缩写形式，可以分开写：

```
div{
  border-width:2px;
  border-style:solid;
  border-color:red;
}
```

注意：

1. border-style（边框样式）常见样式有：

dashed（虚线） | dotted（点线） | solid（实线）。

2. border-color（边框颜色）中的颜色可设置为十六进制颜色，如：

```
border-color:#888;//前面的井号不要忘掉。
```

3. border-width（边框宽度）中的宽度也可以设置为：

thin | medium | thick (但不是很常用), 最常还是用像素 (px)。

- 12-10 使用 border 为盒子添加边框 (二)

现在有一个问题, 如果有想为 p 标签单独设置下边框, 而其它三边都不设置边框样式怎么办呢? css 样式中允许只为一个方向的边框设置样式:

```
div{border-bottom:1px solid red;}
```

同样可以使用下面代码实现其它三边 (上、右、左) 边框的设置:

```
border-top:1px solid red;  
border-right:1px solid red;  
border-left:1px solid red;
```

- 12-11 使用 border-radius 设置圆角

元素边框的圆角效果可以使用 border-radius 属性来设置。圆角可分为左上、右上、右下、左下。如下代码:

```
div{border-radius: 20px 10px 15px 30px;}
```

效果:



也可以分开写:

```
div{  
  border-top-left-radius: 20px;  
  border-top-right-radius: 10px;  
  border-bottom-right-radius: 15px;  
  border-bottom-left-radius: 30px;  
}
```

如果四个圆角都为 10px; 可以这么写:

```
div{ border-radius:10px;}
```

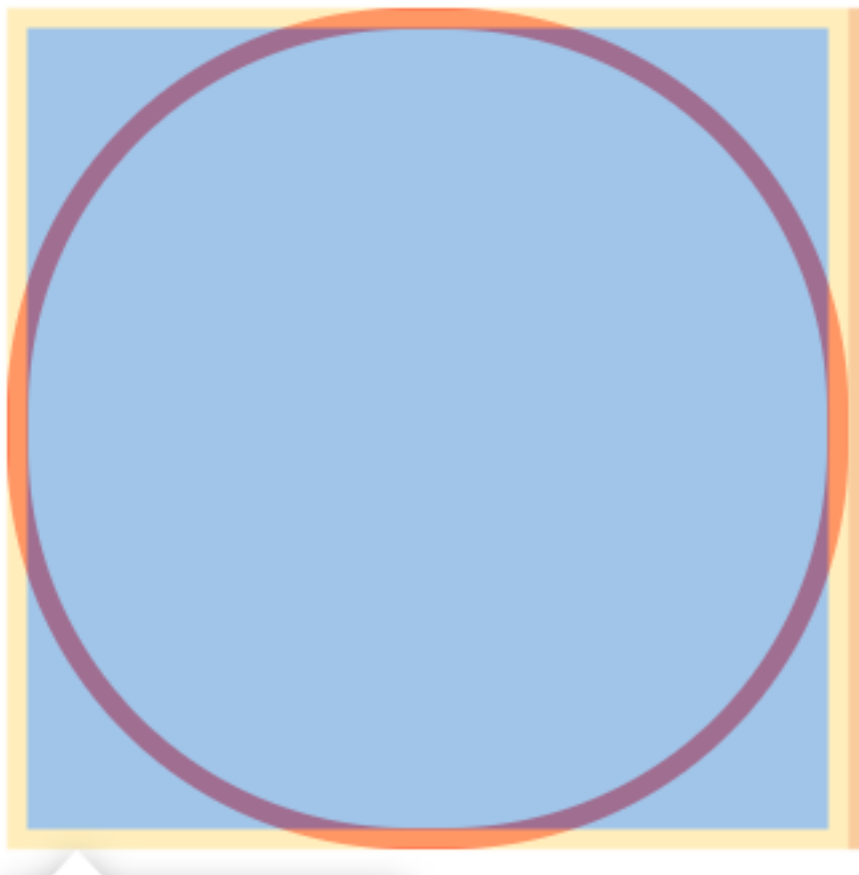
如果左上角和右下角圆角效果一样为 10px, 右上角和左下角圆角一样为 20px, 可以这么写:

```
div{ border-radius:10px 20px;}
```

需要特别注意的：一个正方形，当设置圆角效果值为元素宽度一半时，显示效果为圆形。例如：

```
div {  
    width: 200px;  
    height: 200px;  
    border: 5px solid red;  
    border-radius: 100px;  
}
```

效果：



也可以写为百分比 50%

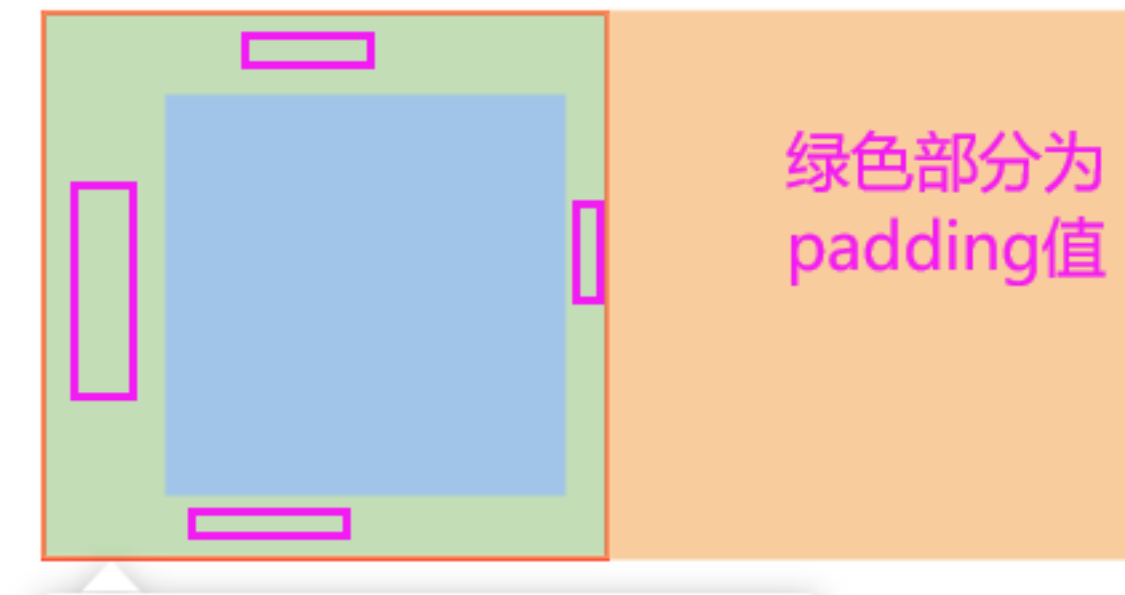
```
div {  
    width: 200px;  
    height: 200px;  
    border: 5px solid red;  
    border-radius: 50%;  
}
```

- 12-12 使用 padding 为盒子设置内边距（填充）

元素内容与边框之间是可以设置距离的，称之为“内边距（填充）”。填充也可分为上、右、下、左（顺时针）。如下代码：

```
div{padding:20px 10px 15px 30px;}
```

效果：



顺序一定不要搞混。可以分开写上面代码：

```
div{  
    padding-top:20px;  
    padding-right:10px;  
    padding-bottom:15px;  
    padding-left:30px;  
}
```

如果上、右、下、左的填充都为 10px; 可以这么写

```
div{padding:10px;}
```

如果上下填充一样为 10px, 左右一样为 20px, 可以这么写：

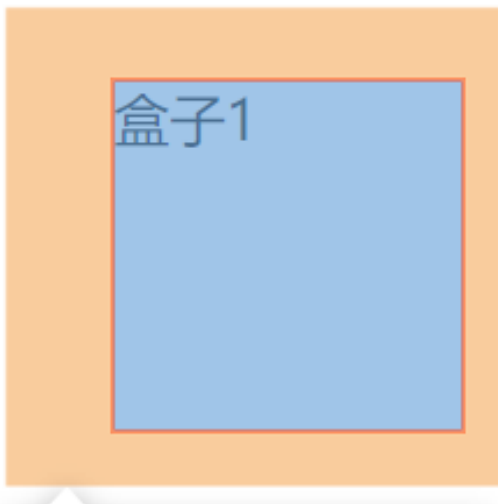
```
div{padding:10px 20px;}
```

- 12-13 使用 margin 为盒子设置外边距（边界）

元素与其它元素之间的距离可以使用边界（margin）来设置。边界也是可分为上、右、下、左。如下代码：

```
div{margin:20px 10px 15px 30px;}
```

效果：



黄色部分为margin
值

也可以分开写：

```
div{  
    margin-top:20px;  
    margin-right:10px;  
    margin-bottom:15px;  
    margin-left:30px;  
}
```

如果上右下左的边界都为 10px; 可以这么写：

```
div{ margin:10px;}
```

如果上下边界一样为 10px，左右一样为 20px，可以这么写：

```
div{ margin:10px 20px;}
```

总结一下：padding 和 margin 的区别，padding 在边框里，margin 在边框外。

13 CSS3 布局模型

- 13-1 css 布局模型

清楚了 CSS3 盒模型的基本概念、盒模型类型，我们就可以深入探讨网页布局的基本模型了。布局模型与盒模型一样都是 CSS3 最基本、最核心的概念。但布局模型是建立在盒模型基础之上，又不同于我们常说的 CSS3 布局样式或 CSS3 布局模板。如果说布局模型是本，那么 CSS3 布局模板就是末了，是外在的表现形式。CSS3 包含 3 种基本的布局模型，用英文概括为：Flow、Layer 和 Float。在网页中，元素有三种布局模型：

1. 流动模型 (Flow)
2. 浮动模型 (Float)
3. 层模型 (Layer)

- 13-2 流动模型 (一)

先来说一说流动模型，流动 (Flow) 是默认的网页布局模式。也就是说网页在默认状态下的 HTML 网页元素都是根据流动模型来分布网页内容的。

流动布局模型具有 2 个比较典型的特征：

第一点，块状元素都会在所处的包含元素内自上而下按顺序垂直延伸分布，因为在默认状态下，块状元素的宽度都为 100%。实际上，块状元素都会以行的形式占据位置。如右侧代码编辑器中三个块状元素标签 (div, h1, p) 宽度显示为 100%。

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>流动模式下的块状元素</title>
    <style type="text/css">
      #box1 {
        width: 300px;
        height: 100px;
      }

      div,
      h1,
      p {
        border: 1px solid red;
      }
    </style>
  </head>

  <body>
    <div id="box2">box2</div>
    <!--块状元素，由于没有设置宽度，宽度默认显示为 100%-->
    <h1>标题</h1>
    <!--块状元素，由于没有设置宽度，宽度默认显示为 100%-->
    <p>文本段文本段</p>
    <!--块状元素，由于没有设置宽度，宽度默认显示为 100%-->
    <div id="box1">box1</div>
    <!--块状元素，由于设置了 width:300px，宽度显示为 300px-->
  </body>
</html>

```

- 13-3 流动模型（二）

第二点，在流动模型下，内联元素都会在所处的包含元素内从左到右水平分布显示。（内联元素可不像块状元素这么霸道独占一行）

右侧代码编辑器中内联元素标签 a、span、em、strong 都是内联元素。

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>流动模式下的内联元素</title>
    <style type="text/css">
    </style>
  </head>

  <body>
    <a href="http://www.imooc.com">www.imooc.com</a>
    <span>强调</span>
    <em>重点</em>
    <strong>强调</strong>
  </body>
</html>

```

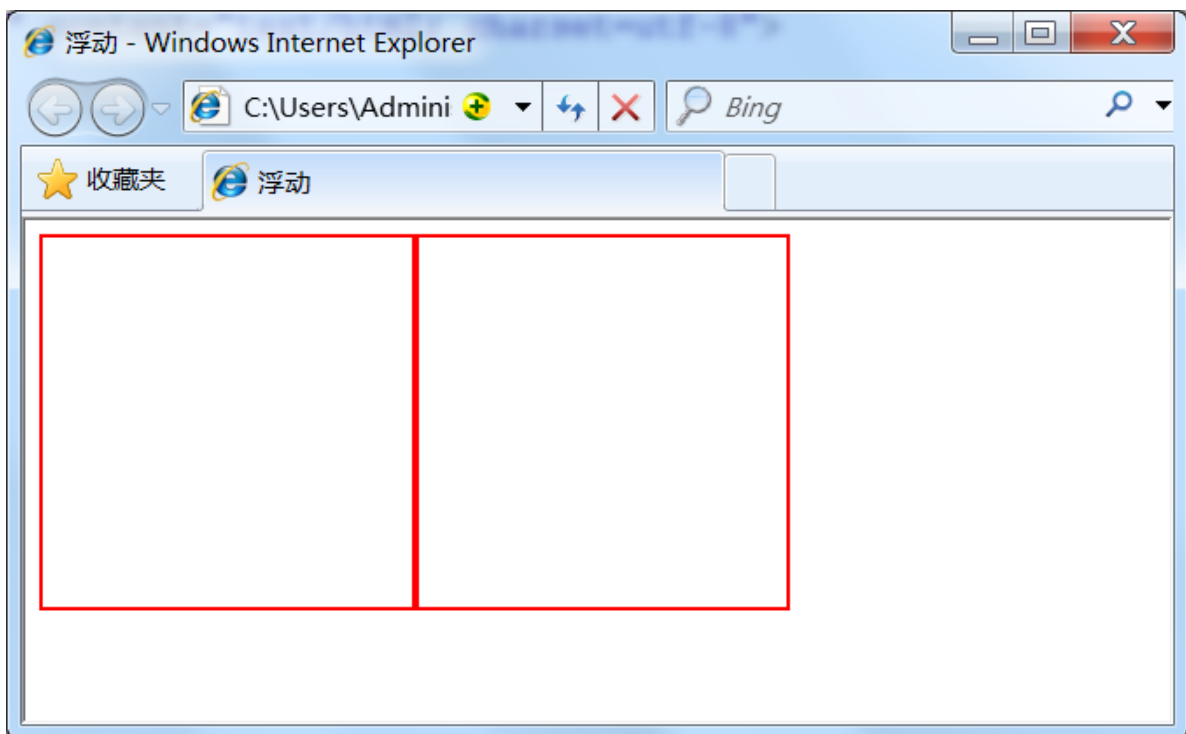
- 13-4 浮动模型

块状元素这么霸道都是独占一行，如果现在我们想让两个块状元素并排显示，怎么办呢？不要着急，设置元素浮动就可以实现这一愿望。

任何元素在默认情况下是不能浮动的，但可以用 CSS 定义为浮动，如 div、p、table、img 等元素都可以被定义为浮动。如下代码可以实现两个 div 元素一行显示。

```
div{
    width:200px;
    height:200px;
    border:2px red solid;
    float:left;
}
<div id="div1"></div>
<div id="div2"></div>
```

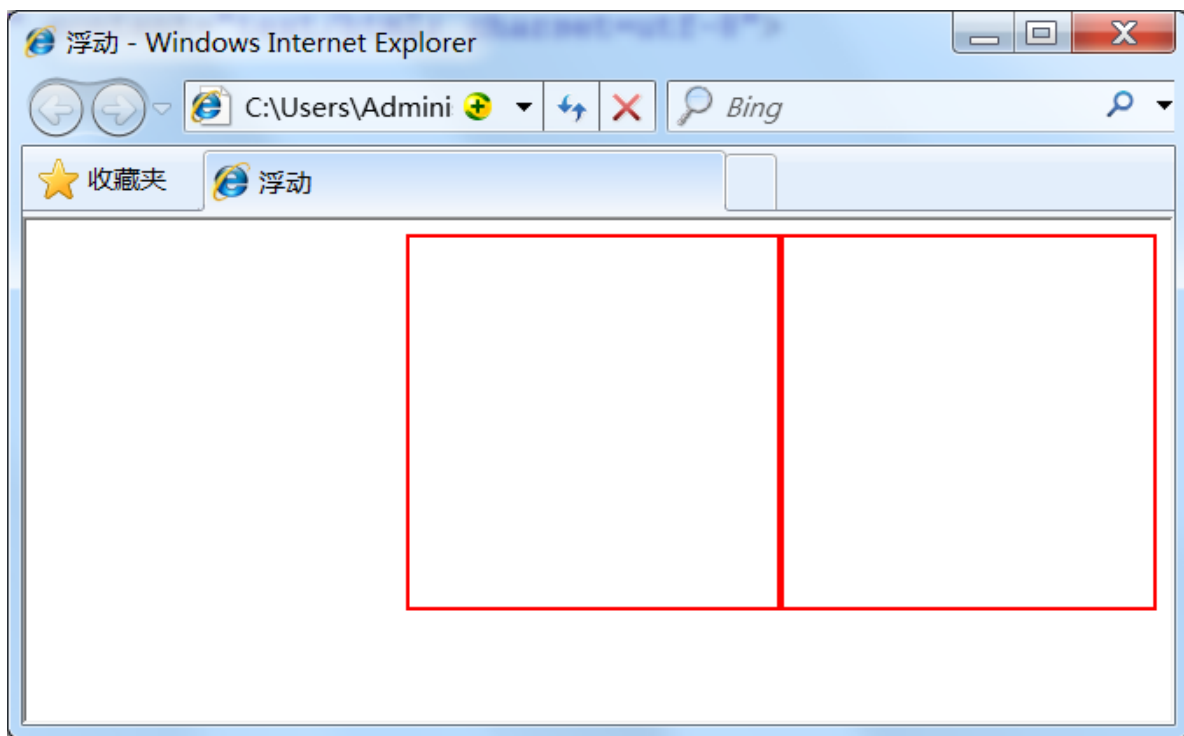
效果图



当然你也可以同时设置两个元素右浮动也可以实现一行显示。

```
div{
    width:200px;
    height:200px;
    border:2px red solid;
    float:right;
}
```

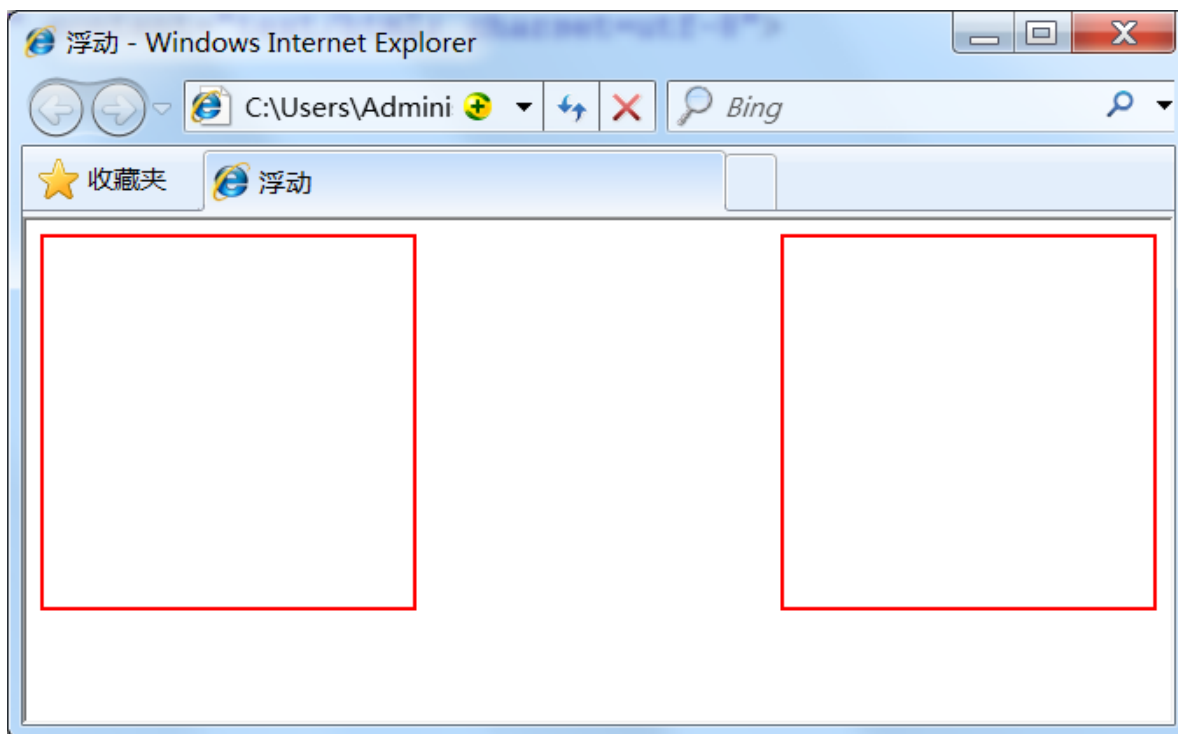
效果图



又有小伙伴问了，设置两个元素一左一右可以实现一行显示吗？当然可以：

```
div{  
    width:200px;  
    height:200px;  
    border:2px red solid;  
}  
#div1{float:left;}  
#div2{float:right;}
```

效果图



- 13-5 什么是层模型？

什么是层布局模型？层布局模型就像是图像软件 Photoshop 中非常流行的图层编辑功能一样，每个图层能够精确定位操作，但在网页设计领域，由于网页大小的活动性，层布局没能受到热捧。但是在网页上局部使用层布局还是有其方便之处的。下面我们来学习一下 html 中的层布局。

如何让 html 元素在网页中精确定位，就像图像软件 Photoshop 中的图层一样可以对每个图层能够精确定位操作。CSS 定义了一组定位 (positioning) 属性来支持层布局模型。

层模型有三种形式：

1. 绝对定位 (position: absolute)
2. 相对定位 (position: relative)
3. 固定定位 (position: fixed)

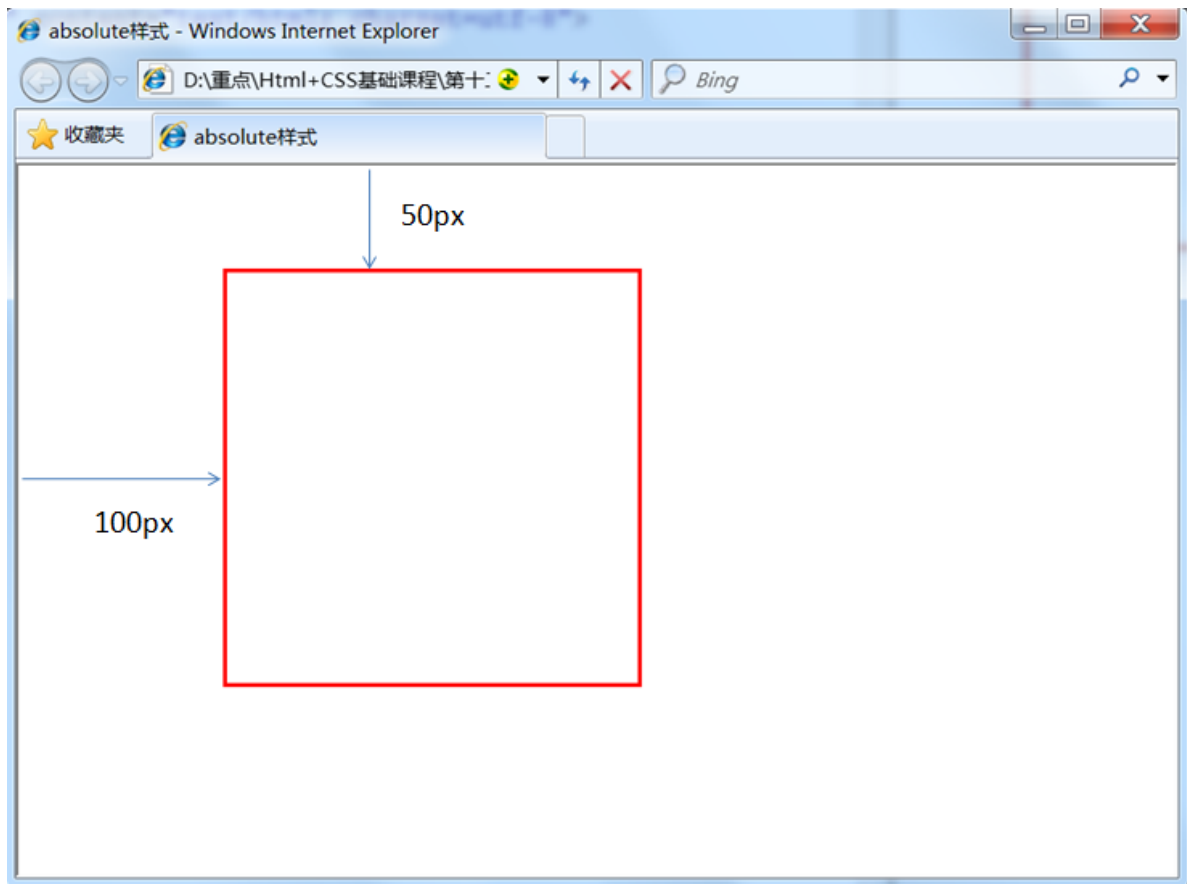
- 13-6 层模型之绝对定位

如果想为元素设置层模型中的绝对定位，需要设置 **position:absolute**(表示绝对定位)，这条语句的作用将元素从文档流中拖出来，然后使用 left、right、top、bottom 属性相对于其最接近的一个具有定位属性的父包含块进行绝对定位。如果不存在这样的包含块，则相对于 body 元素，即相对于浏览器窗口。

如下面代码可以实现 div 元素相对于浏览器窗口向右移动 100px，向下移动 50px。

```
div{
    width:200px;
    height:200px;
    border:2px red solid;
    position:absolute;
    left:100px;
    top:50px;
}
<div id="div1"></div>
```

效果如下：



- 13-7 层模型之相对定位

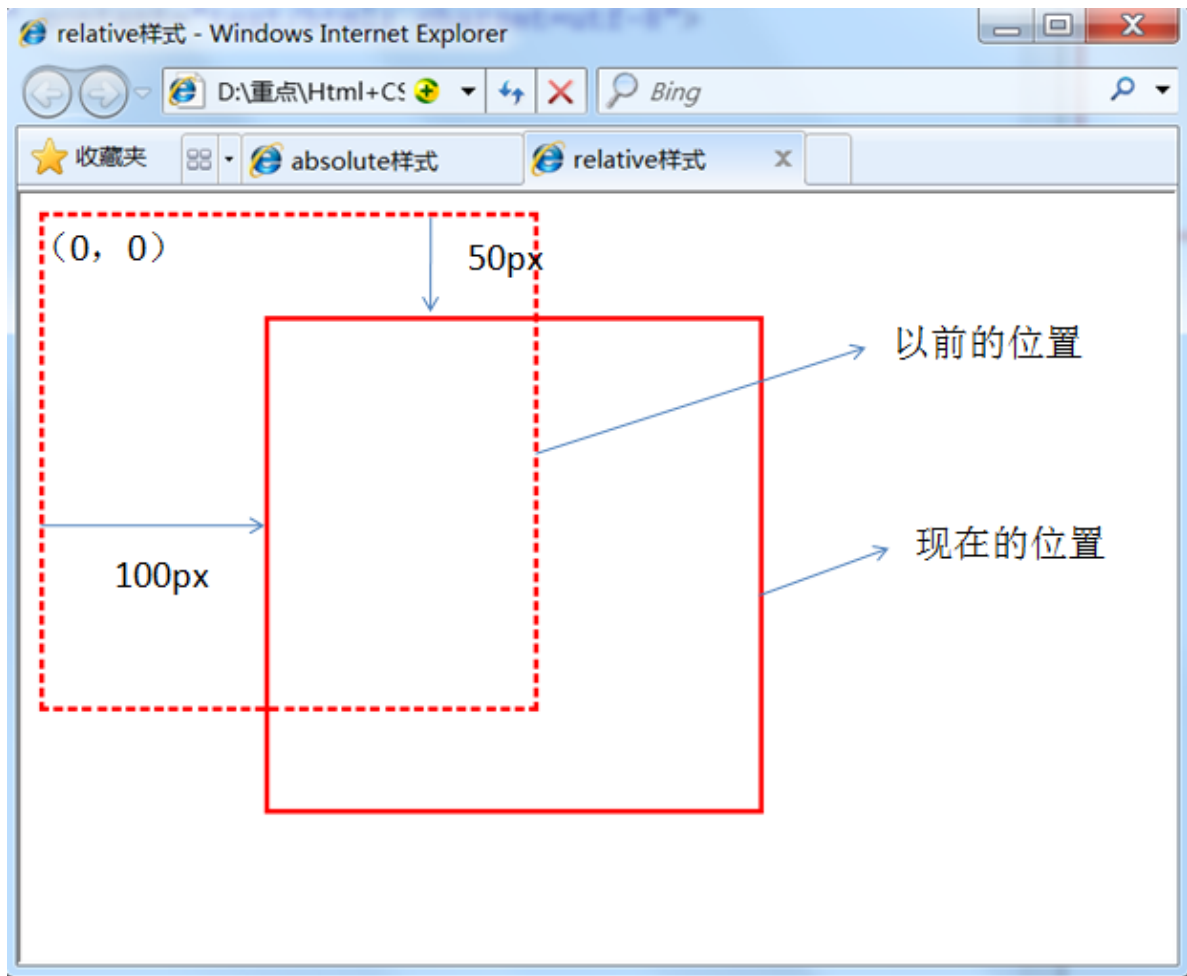
如果想为元素设置层模型中的相对定位，需要设置 `position:relative`（表示相对定位），它通过 `left`、`right`、`top`、`bottom` 属性确定元素在正常文档流中的偏移位置。相对定位完成的过程是首先按 `static(float)` 方式生成一个元素（并且元素像层一样浮动了起来），然后相对于以前的位置移动，移动的方向和幅度由 `left`、`right`、`top`、`bottom` 属性确定，偏移前的位置保留不动。

如下代码实现相对于以前位置向下移动 50px，向右移动 100px；

```
#div1{
    width:200px;
    height:200px;
    border:2px red solid;
    position:relative;
    left:100px;
    top:50px;
}
```

```
<div id="div1"></div>
```

效果图：

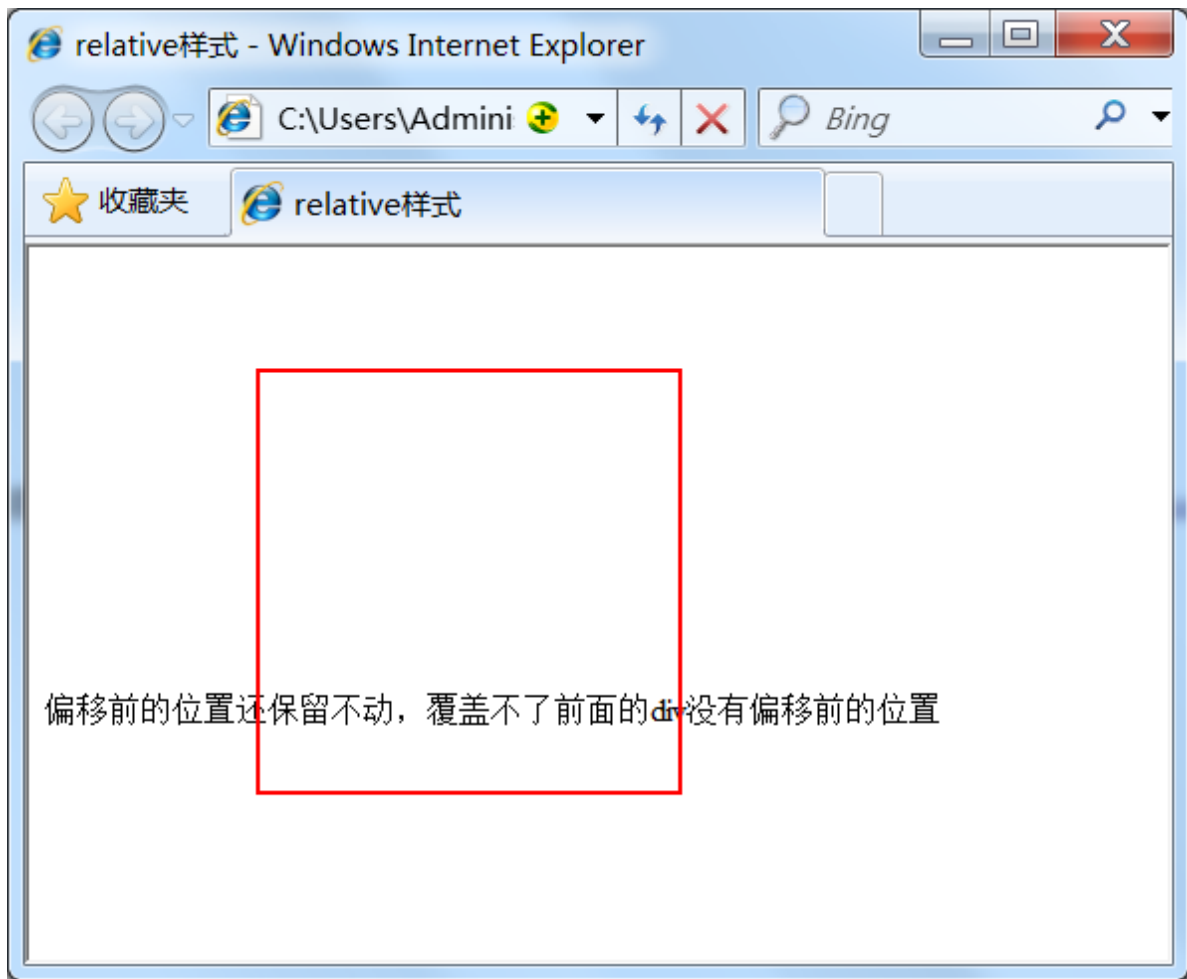


什么叫做“偏移前的位置保留不动”呢？

大家可以做一个实验，在右侧代码编辑器的 19 行 div 标签的后面加入一个 span 标签，在标并在 span 标签中写入一些文字。如下代码：

```
<body>
  <div id="div1"></div><span>偏移前的位置还保留不动，覆盖不了前面的 div 没有偏移前的位置 </span>
</body>
```

效果图：



从效果图中可以明显的看出，虽然 div 元素相对于以前的位置产生了偏移，但是 div 元素以前的位置还是保留着，所以后面的 span 元素是显示在了 div 元素以前位置的后面。

- 13-8 层模型之固定定位

fixed: 表示固定定位，与 absolute 定位类型类似，但它的相对移动的坐标是视图（屏幕内的网页窗口）本身。由于视图本身是固定的，它不会随浏览器窗口的滚动条滚动而变化，除非你在屏幕中移动浏览器窗口的屏幕位置，或改变浏览器窗口的显示大小，因此固定定位的元素会始终位于浏览器窗口内视图的某个位置，不会受文档流动影响，这与 background-attachment:fixed; 属性功能相同。以下代码可以实现相对于浏览器视图向右移动 100px，向下移动 50px。并且拖动滚动条时位置固定不变。

```
#div1{
    width:200px;
    height:200px;
    border:2px red solid;
    position:fixed;
    left:100px;
    top:50px;
}
<p>文本文本文本文本</p>
```

- 13-9 Relative 与 Absolute 组合使用

小伙伴们学习了 12-6 小节的绝对定位的方法：使用 `position:absolute` 可以实现被设置元素相对于浏览器（body）设置定位以后，大家有没有想过可不可以相对于其它元素进行定位呢？答案是肯定的，当然可以。使用 `position:relative`

来帮忙，但是必须遵守下面规范：

1. 参照定位的元素必须是相对定位元素的前辈元素：

```
<div id="box1"><!--参照定位的元素-->
  <div id="box2">相对参照元素进行定位</div><!--相对定位元素-->
</div>
```

从上面代码可以看出 box1 是 box2 的父元素（父元素当然也是前辈元素了）。

2. 参照定位的元素必须加入 position:relative;

```
#box1{
  width:200px;
  height:200px;
  position:relative;
}
```

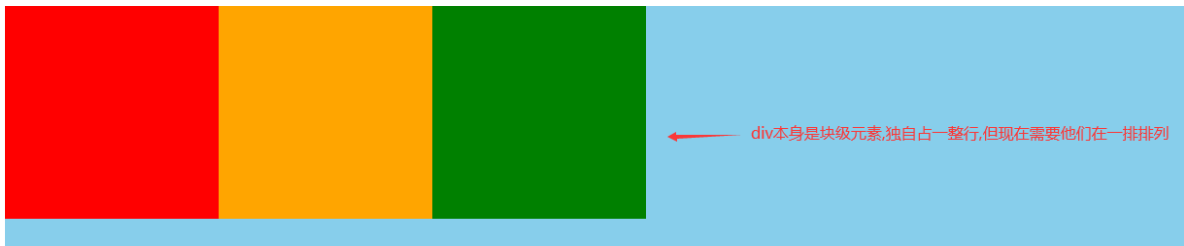
3. 定位元素加入 position:absolute，便可以使用 top、bottom、left、right 来进行偏移定位了。

```
#box2{
  position:absolute;
  top:20px;
  left:30px;
}
```

这样 box2 就可以相对于父元素 box1 定位了（这里注意参照物就可以不是浏览器了，而可以自由设置了）。

14 flex 弹性盒模型

- 14-1 弹性盒模型之 flex 属性



实现上图效果，我们需要输入以下代码：

```
<body>
  <div class="box"></div>
  <div class="box1"></div>
  <div class="box2"></div>
  <div class="box3"></div>
</body>
```

```
<style type="text/css">
  .box {
    height: 400px;
    background:skyblue;
    display: flex;
  }
  .box1 {
    width: 200px;
    height: 200px;
    background: red;
  }
```

```

}
.box2 {
  width: 200px;
  height: 200px;
  background: orange;
}
.box3 {
  width: 200px;
  height: 200px;
  background: green;
}
</style>

```

上面的代码：三个块元素设置大小以及背景色，在父容器中添加 flex。

解释：

1. 设置 display: flex 属性可以把块级元素在一排显示。
 2. flex 需要添加在父元素上，改变子元素的排列顺序。
 3. 默认为从左往右依次排列，且和父元素左边没有间隙。
- 14-2 使用 justify-content 属性设置横轴排列方式

这一章节我们来学习 justify-content 属性，本属性定义了项目在主轴上的对齐方式。结合上一节的布局例子进行理解，属性值分别为：

```
justify-content: flex-start | flex-end | center | space-between | space-around;
```

flex-start : 交叉轴的起点对齐

```

.box {
  background: blue;
  display: flex;
  justify-content: flex-start;
}

```

实现效果：



flex-end : 右对齐

```

.box {
  background: blue;
  display: flex;
  justify-content: flex-end;
}

```

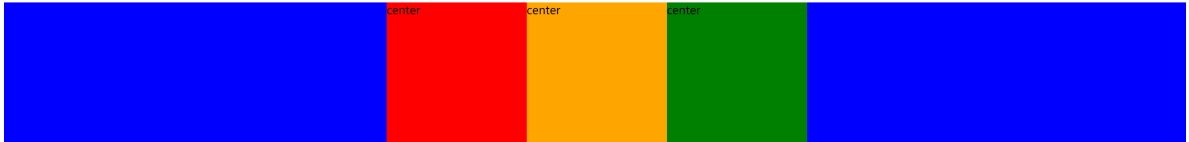
实现效果：



`center` : 居中

```
.box {  
  background: blue;  
  display: flex;  
  justify-content: center;  
}
```

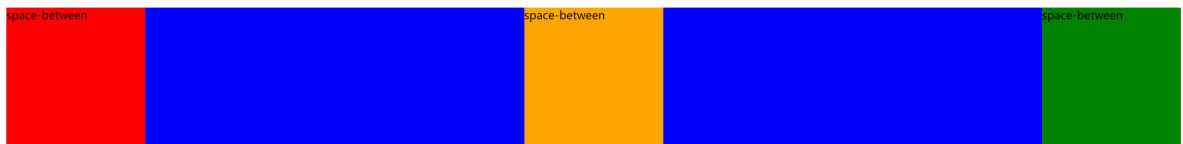
实现效果:



`space-between` : 两端对齐, 项目之间的间隔都相等。

```
.box {  
  background: blue;  
  display: flex;  
  justify-content: space-between;  
}
```

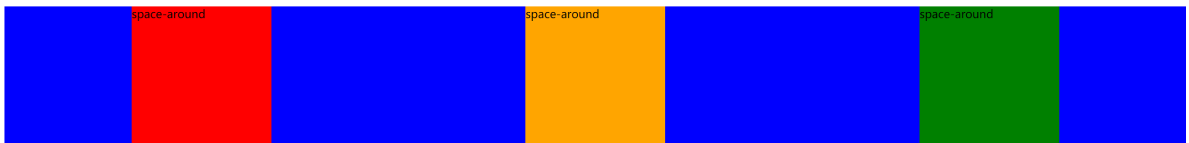
实现效果:



`space-around` : 每个项目两侧的间隔相等。所以, 项目之间的间隔比项目与边框的间隔大一倍。

```
.box {  
  background: blue;  
  display: flex;  
  justify-content: space-around;  
}
```

实现效果:



- 14-3 使用 `align-items` 属性设置纵轴排列方式

这一章节我们来学习 `align-items` 属性, 本属性定义了项目在交叉轴上的对齐方式。属性值分别为:

`align-items`: `flex-start` | `flex-end` | `center` | `baseline` | `stretch`;

结合右侧编辑器中的布局以及下面的样式设置进行理解:

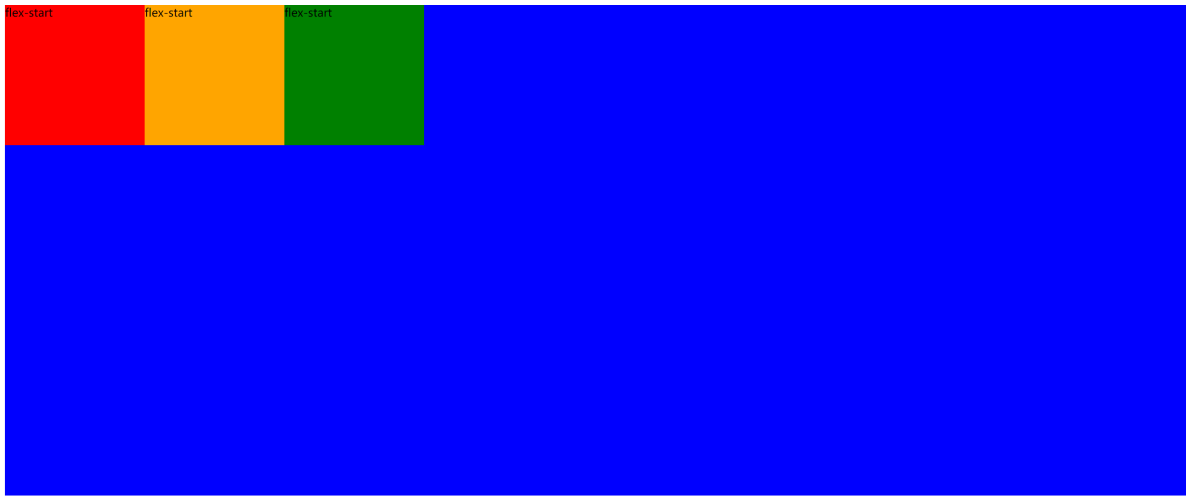
`flex-start` : 默认值, 左对齐

```
.box {  
  height: 700px;  
  background: blue;  
  display: flex;
```



```
    align-items: flex-start;
}
```

实现效果:



flex-end : 交叉轴的终点对齐

```
.box {
    height: 700px;
    background: blue;
    display: flex;
    align-items: flex-end;
}
```

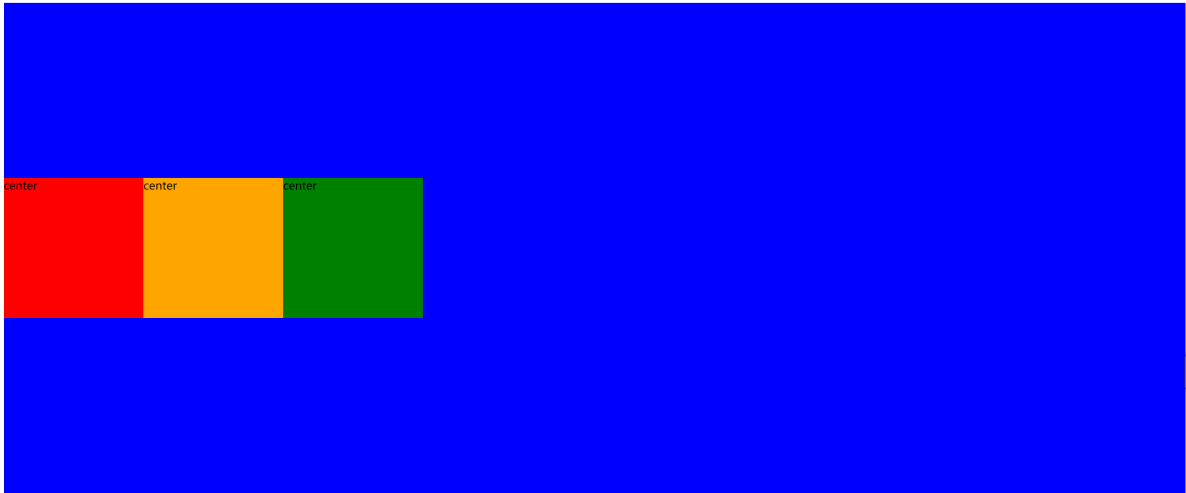
实现效果:



center : 交叉轴的中点对齐

```
.box {
    height: 700px;
    background: blue;
    display: flex;
    align-items: center;
}
```

实现效果：



baseline : 项目的第一行文字的基线对齐。

```
<head>
  <meta charset="UTF-8">
  <title>align-items</title>
  <style type="text/css">
    .box {
      height: 700px;
      background: blue;
      display: flex;
      align-items: baseline;
    }

    .box div {
      width: 200px;
      height: 200px;
    }

    .box1 {
      background: red;
    }

    .box2 {
      font-size: 30px;
      background: orange;
    }

    .box3 {
      font-size: 50px;
      background: green;
    }
  </style>
</head>

<body>
  <div class="box">
    <div class="box1">baseline</div>
```

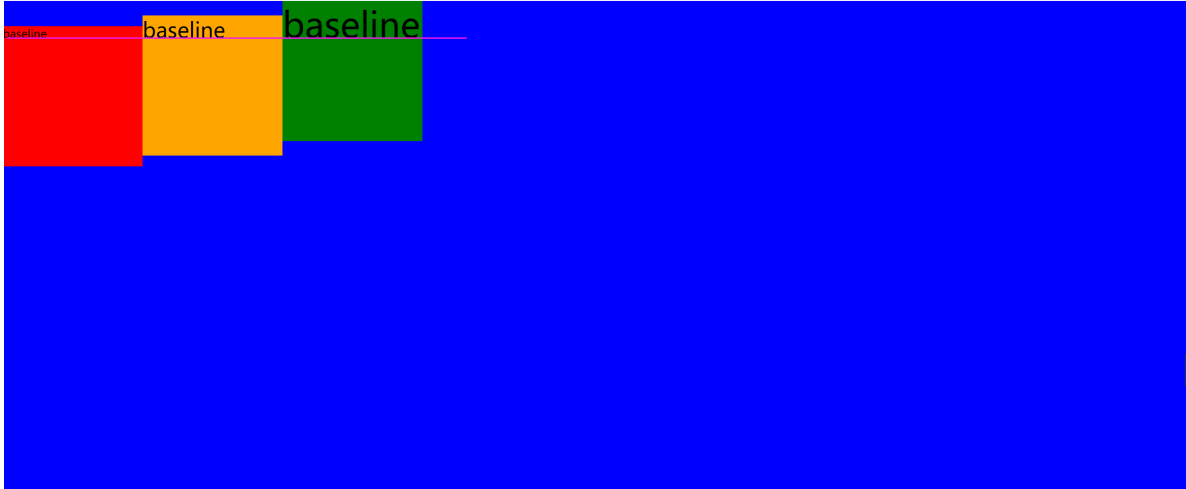
```

        <div class="box2">baseline</div>
        <div class="box3">baseline</div>
    </div>
</body>

```

三个盒子中设置不同的字体大小。

实现效果：



stretch (默认值)：如果项目未设置高度或设为 auto，将占满整个容器的高度。

```

.box {
    height: 300px;
    background: blue;
    display: flex;
    align-items: stretch;
}

.box div {
    /* 不设置高度，元素在垂直方向上铺满父容器 */
    width: 200px;
}

```

实现效果：



• 14-4 给子元素设置 flex 占比

这一章节我们来学习 flex 属性，设置子元素相对于父元素的占比。

可以参考代码：

```

<!DOCTYPE html>
<html>
  <head>

```

```

<meta charset="UTF-8">
<title>flex 占比</title>
<style type="text/css">
.box {
    height: 300px;
    background: blue;
    display: flex;
}

.box div {
    width: 200px;
    height: 200px;
}

.box1 {
    flex: 1;
    background: red;
}

.box2 {
    flex: 3;
    background: orange;
}

.box3 {
    flex: 2;
    background: green;
}
</style>
</head>

<body>
<div class="box">
    <div class="box1">flex:1</div>
    <div class="box2">flex:3</div>
    <div class="box3">flex:2</div>
</div>
</body>
</html>

```

测试效果如下：



解释：

1. 给子元素设置 flex 属性，可以设置子元素相对于父元素的占比。
2. flex 属性的值只能是正整数，表示占比多少。
3. 给子元素设置了 flex 之后，其宽度属性会失效。

15 CSS 样式设置小技巧

- 15-1 水平居中设置-行内元素

我们在实际工作中常会遇到需要设置水平居中的场景，比如为了美观，文章的标题一般都是水平居中显示的。

这里我们又得分两种情况：行内元素 还是 块状元素，块状元素里面又分为定宽块状元素，以及不定宽块状元素。今天我们先来了解一下行内元素怎么进行水平居中？

如果被设置元素为文本、图片等行内元素时，水平居中是通过给父元素设置 `text-align:center` 来实现的。（父元素和子元素：如下面的 html 代码中，div 是“我想要在父容器中水平居中显示”这个文本的父元素。反之这个文本是 div 的子元素）如下代码：

html 代码：

```
<body>
  <div class="txtCenter">我想要在父容器中水平居中显示。</div>
</body>
```

css 代码：

```
<style>
.txtCenter{
  text-align:center;
}
</style>
```

- 15-2 水平居中设置-定宽块状元素

当被设置元素为 块状元素 时用 `text-align:center` 就不起作用了，这时也分两种情况：定宽块状元素和不定宽块状元素。

这一小节我们先来讲一讲定宽块状元素。（定宽块状元素：块状元素的宽度 `width` 为固定值。）

满足定宽和块状两个条件的元素是可以通过设置“左右 `margin`”值为“`auto`”来实现居中的。我们来看个例子就是设置 `div` 这个块状元素水平居中：

html 代码：

```
<body>
  <div>我是定宽块状元素，哈哈，我要水平居中显示。</div>
</body>
```

css 代码：

```
<style>
div{
  border:1px solid red;/* 为了显示居中效果明显为 div 设置了边框 */
  width:200px;/* 定宽 */
  margin:20px auto;/* margin-left 与 margin-right 设置为 auto */
}
</style>
```

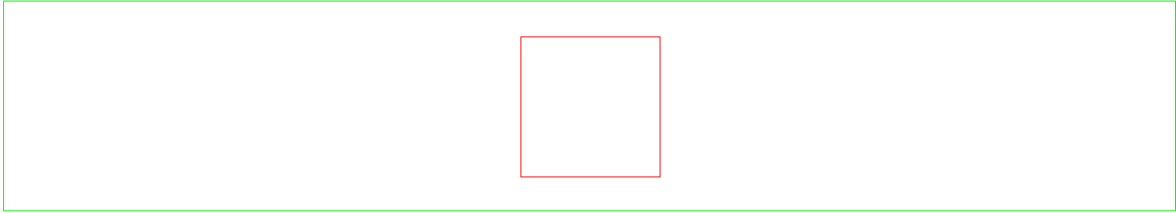
也可以写成：

```
margin-left:auto;
margin-right:auto;
```

注意：元素的“上下 `margin`”是可以随意设置的。

- 15-3 面试常考题之已知宽高实现盒子水平垂直居中

这一章节我们来学习已知宽高实现盒子水平垂直居中。通常使用定位完成，例如想要实现以下效果：



我们有如下两个 div 元素

```
<body>
  <div class="box">
    <div class="box1"></div>
  </div>
</body>
```

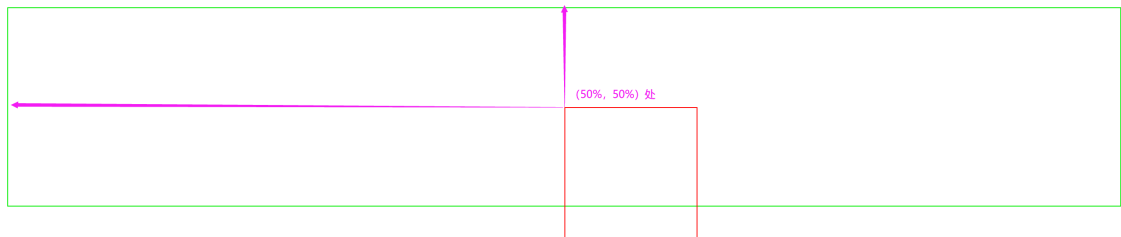
要实现子元素相对于父元素垂直水平居中, 我们只需要输入以下代码:

```
<style type="text/css">
  .box {
    border: 1px solid #00ee00;
    height: 300px;
    position: relative; /* 父元素相对定位 */
  }

  .box1 {
    position: absolute; /* 子元素绝对定位 */
    top: 50%; /*top 值为 50%*/
    left: 50%; /*left 值为 50%*/
    margin: -100px 0 0 -100px; /*margin-top 的值为负的高度的一半, margin-left 的值为负的宽度的一半 */
    width: 200px;
    height: 200px;
    border: 1px solid red;
  }
</style>
```

解释:

1. 利用父元素设置相对定位, 子元素设置绝对定位, 那么子元素就是相对于父元素定位的特性。
2. 子元素设置上和左偏移的值都为 50%, 是元素的左上角在父元素中心点的位置。效果:



3. 然后再用 margin 给上和左都给负的自身宽高的一半, 就能达到垂直水平居中的效果。

- 15-4 面试常考题之宽高不定实现盒子水平垂直居中

这一章我们来学习未知宽高实现盒子水平垂直居中, 通常使用定位以及 translate 完成。参考下面例子:

```
<div class="box">
  <div class="box1">
```

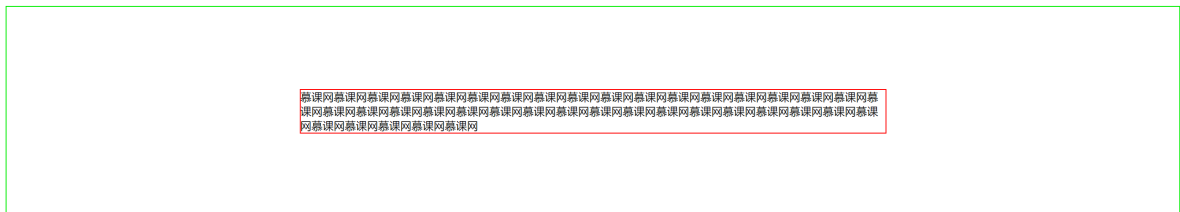
```
慕课网慕课网慕课网慕课网.....
</div>
</div>
```

添加样式:

```
.box {
  border: 1px solid #00ee00;
  height: 300px;
  position: relative;
}

.box1 {
  border: 1px solid red;
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
}
```

效果如下:



技术点的解释:

1. 利用父元素设置相对定位, 子元素设置绝对定位, 那么子元素就是相对于父元素定位的特性。
2. 子元素设置上和左偏移的值都为 50%。
3. 然后再用 css3 属性 translate 位移, 给上和左都位移-50% 距离, 就能达到垂直水平居中的效果。