

Html 学习指南

作者：向仔州

URL 地址协议.....	3
Link 标签 偷别人网站的 icon.....	3
Img 标签 图片显示.....	4
H1 ~ H6 标签 显示字符大小方便网站 SEO.....	6
P 标签 段落文本.....	7
Hr 标签 分隔符.....	8
 空格符.....	10
span 标签	10
em 标签	10
i 标签 斜体字.....	11
sub 标签 上标.....	11
sup 标签 下标.....	11
del 标签 字符删除.....	11
超级链接标签 a 标签.....	11
锚点链接.....	13
ul 标签 HTML 有序列表和无序列表.....	15
ol 标签 有序列表.....	16
dl 标签 自定义.....	17
dl 列表标签案例.....	18
table 标签 表格标签.....	20
div 标签	25
span 标签的使用.....	26
form 标签 表单使用.....	27
input 标签	27
select 标签 下拉列表.....	28
textarea 标签 文本域.....	30
如果我想网页有多选功能怎么办？	31
如果我要给网页输入密码怎么办？ 那就要用密码框.....	32
<label></label>标签：该标签是指明哪一个字符，在鼠标移动到这个字符的时候显示字符的特殊颜色或者功能。	32
重置按钮.....	32
QQ 用户注册案例:.....	33
from 标签上报服务器功能.....	34
From 标签里用图片当做按钮来提交数据.....	38
勾选框和选择标签如何设置页面打开默认选择.....	39
fieldset 标签 表单分组.....	40
iframe 标签 内联标签.....	41
base 标签	42
字符实体	43

我们整个 html 包含两个标签

- 1: 头部用<head></head>标签
- 2: 还有个就是<body></body>标签

```
<html>
  <head>
    <!--头部标签--> html 语法注释就要用尖括号加感叹号,快速注释快捷键 ctrl+/
    <title></title>      <!--title 是头部标签, 这个标签是给浏览器用的-->
  </head>
  <body>
    <!--body 标签里放的是网站上面用户看到的东西 也就算内容标签-->
  </body>
</html>
```

以上就是 html 的结构

给标签里面加属性

```
<html lang = "zn_CH">      如果要给标签里面加属性就要在标签中间空格 html  lang 这样
</html>
```

文本文件用 txt.htm 和 txt.html 作为后缀都是一样的, 我喜欢用.html

DOCTYPE 标签

<!DOCTYPE html> 就是表示这个文本是支持 html5 的

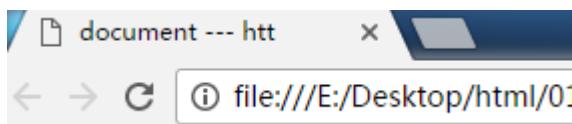
<html lang="en"> <!-- 搜索引擎优化, 当搜索引擎打开这个页面的时候可以推
断改页面是中文还是英文

<title>document --- htt</title>

title 标签就是这个

<title>document --- htt</title>

浏览器 tab 栏的名字



注意: 每次打开 sublime 的时候中文字符都会变成乱码, 所以在用
sublime 的时候要设置保存格式在 file->Reopen with Encoding->UTF-8

如果浏览器打开后显示中文乱码,

```
<head>
    <title></title>
    <meta charset=UTF-8>
</head>
```

需要加 UFT8

head 标签 : `<head></head>` 该标签用于嵌套给浏览器设置用的标签

link 标签: `<link rel="dns-prefetch" href="http://baidu.com">`

link 标签就是域名的预解析，让你在我的网页打开其他网站比在其他网页上打开这个网站快一点

link 标签用来偷取别人网站上 icon 图标给自己用

```
<head>
    <meta charset="utf-8">
    <title>02demo</title>
    <link rel="shortcut icon" href="http://www.itcast.com/">
</head>
```

`href="http://www.itcast.com/"` 这一段就是要写入你要偷取哪个网

站的 icon 图标，把那个网站地址写上来



URL 地址协议

协议规定格式: `scheme: //host.domain:port/path/filename`

`scheme` – 定义因特网服务类型，最常用的是 `http`

`host`-定义主机，`http` 协议默认主机是 `www`.

`Port`-定义主机上端口号，如果没定义 `http` 默认是 `80`

`Path`-定义服务器上的路径，如果没定义，则文件必须在服务器根目录下

`Filename` 定义文件名称

比如:

`scheme: //host.domain:port/path/filename`

`http://www.itcast.cn`

协议 HTTP

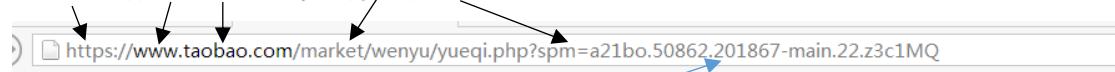
域名

没有端口号，
默认 80 端口

路径也没有文件也没有，那么就
是访问网站根目录下的文件

在来一个新的网站

scheme: //host.domain:port/path/filename



就是浏览器通过这个协议去访问服务器下面的这个文件。这就是 URL 协议
最常用的协议就是 http, https, ftp。

http 协议就是浏览器网页常用的协议

https 协议就是比 http 多了一个加密功能

ftp 就是一个文件传输协议，比如从服务器下载文件就走 ftp。

网页图片显示

src = 里面存放图片的地址

alt = 如果地址里面没有 imgg.png 图片那么就显示 alt 里面的文字

title = 如果鼠标移动到图片上，就会在鼠标上显示 title 里面的文字

width = 设置图片显示宽度

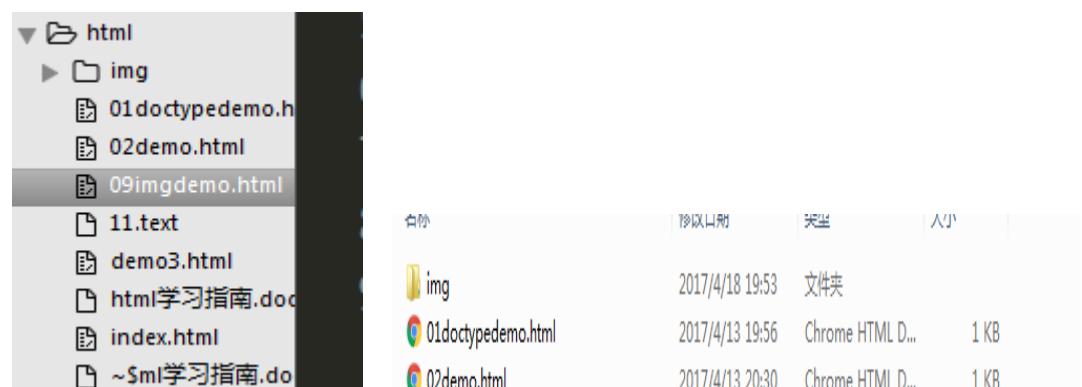
height = 设置图片显示高度

例如：

我要在网页下显示图片

先在工程里面建立一个存放图片的文件夹，记住一定要文件夹在 html 工程目录下

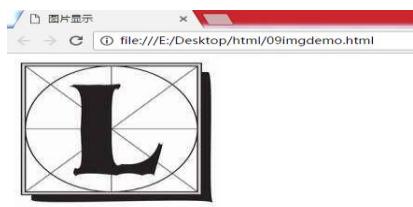
右击工程 new folder 写上文件夹名称



然后再将图片放入 img，图片格式支持 jpg, png, gif

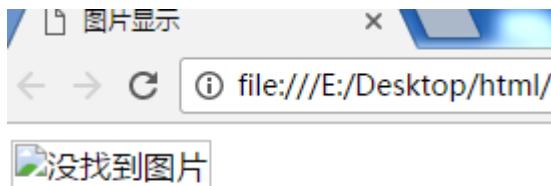


```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>图片显示</title>
</head>
<body>
    
</body>
</html>
```



执行代码后就是上面这个 L 图片

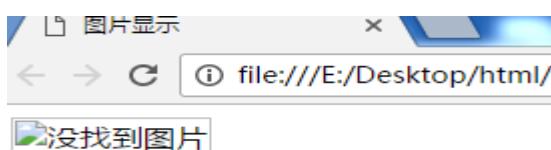
如果将 L 图片在目录下删除



这就是 alt="" 里面的内容

或者在代码里面把图片的名字或者路径写错了

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta charset="utf-8">
5     <title>图片显示</title>
6
7 </head>
8 <body>
9
10    
11 </body>
12 </html>
```



这就是 alt="" 里面的内容

图片显示就是这样操作的。

```
<body>
    
</body>
```

这是设置图片在浏览器的大小

登录QQ浏览器 | 上网导航 | 上网导航 | 百度



Gif 格式图片：支持动画，但是颜色支持很少，只能处理 256 种颜色。好处在于支持透明颜



色，图片容量小。

这种简单的颜色。不能色彩太多的图片。

Jpg 格式图片：就比 GIF 图片支持的色彩多，但是不能动画，比如广告就要用这个，



这就是 jpg 图片，支持颜色渐变，绚丽的色彩。

Html 中的 h1 到 h6 标签

```
View Goto Tools Project Preferences Help
1  <!DOCTYPE html> <!-- DOC怕怕怕怕呀 -->
2  <html lang="en">
3  <head>
4      <meta charset="utf-8">
5      <title>document --- http://</title>
6  </head>
7  <body>
8      <h1>h1标签显示</h1>
9      <h2>h2标签显示</h2>
0      <h3>h3标签显示</h3>
1      <h4>h4标签显示</h4>
2      <h5>h5标签显示</h5>
3      <h6>h6标签显示</h6>
4  </body>
5 </html>
```

网页显示效果

h1标签显示

h2标签显示

h3标签显示

h4标签显示

h5标签显示

h6标签显示

我觉得就是字体大小的区别

但是一般网页的开头大标题都是用 h1,



```
/*44m 14s 28ms*/ type: 'css'; msi: 'https://i.sinaimg.cn/common/nav/username/1/1/top_account_v2.css';  
<!--页面顶部 begin--&gt;<br/>(script charset='utf-8' type="text/javascript" src="/i.sinaimg.cn/common/channelNav/1js/1a.js"/>  
(link rel="stylesheet" href="/i0.sinaimg.cn/common/channelNav/css/common_nav.css" type="text/css")  
<!--主导航--&gt;<br/>>(div class="sinajs-top-bar-wrap" id="sinajs-top-bar-wrap-astro" data-sudaclick="public_topnav")</div>  
(!--主导航-->  
(!--次级领导 end-->  
>(div class="wrap-outer clearfix" id="wrapOuter")  
>(div class="wrap-inner")  
>(div class="site-header clearfix")</div>  
>(div class="page-header")  
>(h1 id="articleTitle" cl="187316" docid="fyelmy2500043" style="font-size: 1em;">习近平出席  
并主持一带一路国际合作高峰论坛<h1>
```

也不知道是行业
标准还是什么的
必须用 h1

其实 h1 到 h6 字体样式可以用 CSS 来改，所以 h1 和 h6 字体样式意义不大。
不过是浏览器默认的字体样式规格。

一般情况下一个页面就只有有个 H1 标签，不要在一个页面里面乱加 H1，
否则百度搜索很难找到这个网页。

H2 以后的标签可以在一个页面有多个，但是不要滥用，否则搜索引擎会认为网页作弊。

Html 中段落标签 p 标签

P 标签: <p>段落文本</p>

比如网页里面里面的一些段落

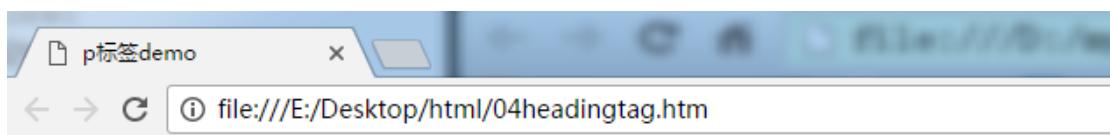
新华社北京4月18日电（记者 李宣良）中共中央总书记、国家主席、中央军委主席习近平18日接见新调整组建军级单位主官，并对各单位发布训令，强调要坚决贯彻党中央和中央军委决策指示，以党在新形势下的强军目标为引领，坚持政治建军、改革强军、依法治军，聚焦能打仗、打胜仗推进各项工作，聚精会神锻造召之即来、来之能战、战之必胜的精兵劲旅，为实现中国梦强军梦作出新的更大的贡献。

```
, wUrl: 'http://i.sinaimg.cn/100/1/180/14277.htm', mainUrl: 'http://i.sinaimg.cn/100/180/14277.htm', "wapURL": "http://video.sina.cn/news/j/2017-04-18/detail-ifyelmy2575872.d.html"], "downLoadDescr": "", "videoCard": false})&mc-e-plugin-videoList2-->  
(p) 原标题：习近平接见新调整组建军级单位主官 对各单位发布训令</p>  
>(p)  
新华社北京4月18日电（记者 李宣良）中共中央总书记、国家主席、中央军委主席习近平18日接见新调整组建军级单位主官，并对各单位发布训令，强调要坚决贯彻党中央和中央军委决策指示，以党在新形势下的强军目标为引领，坚持政治建军、改革强军、依法治军，聚焦能打仗、打胜仗推进各项工作，聚精会神锻造召之即来、来之能战、战之必胜的精兵劲旅，为实现中国梦强军梦作出新的更大的贡献。
```

这种就是 P 标签做的。

```
Goto Tools Project Preferences Help
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>p标签demo</title>
</head>
<body>
    <h1>段落标签p标签使用</h1>
    <p>p标签是段落使用方法</p>
    <p>两个p标签段落是相互隔开的</p>
</body>
</html>
```

P 标签的效果



段落标签p标签使用

p标签是段落使用方法

两个p标签段落是相互隔开的

一个 p 标签独占网页一行，你看第二个 p 标签就单独来一行，没有接着第一个 p 标签屁股后面跟着显示

两个 p 标签中间有空白自动隔开。

h1~h6 标签和 p 标签都是独占段落，写了几行段落的字就占几行。

hr 标签：水平线标签<hr>

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>p标签demo</title>
</head>
<body>
    <h1>段落标签p标签使用</h1>
    <p>p标签是段落使用方法</p>
    <hr>
    <p>两个p标签段落是相互隔开的</p>
</body>
</html>
```

分隔行标签

下面图片就是 hr 标签的显示

段落标签p标签使用

p标签是段落使用方法

两个p标签段落是相互隔开的

在两个段落之中加入 hr 标签就可以在两个段落之中加分隔行。

空格使用：我在 p 标签文字段落里面加了很多空格。

```
<p>p标签是段落使用方法</p>
```

浏览器显示结果

p标签是段落使用方法 这个空格貌似空的没有 sublime 要求的这么多。

怎么只有一个空格

这个就涉及到 html 文档空格合并功能，也就是多个空格或者换行会合并成一个空格。

如果我就想让是 段落这里换行怎么办呢？

请用
标签

```
<p>p标签是<br>段落使用方法</p>
```

浏览器显示为

段落标签p标签使用

p标签是
段落使用方法

就将
标签后的段落换行了

两个p标签段落是相互隔开的

如果我想在两个文字之间加空格怎么办呢？

请用

<p>p标签是 段落使用方法</p>

一个 添加一个空格

段落标签p标签使用

p标签是 段落使用方法

<p>p标签是 段落使用方法</p>
多个 加多个空格

p标签是 段落使用方法

这就是空格区别

span 标签:

span 标签是段落中的部分文字包裹起来,然后进行处理,比如 CSS 处理该段落包裹的文字

<p> 这p标签要用span标签来标注重点 重点处理标注了</p>

浏览器显示为:



这p标签要用span标签来标注重点 重点处理标注了

其实看不出

什么, 只有用 css 才能看出什么

两行 span 标签是没有换行功能的, 不像 p 标签可以换行

-----重点-----
+++++++=+++++++=+++++++=+++++++=+++++++=

浏览器显示为

-----重点----- ++++++=+++++=+++++=+++++=+++++=+++++=

不像 p 标签可以换行

<p> 这p标签要用span标签来标注重点 重点处理标注了</p>
<p> p标签有换行功能</p>

这p标签要用span标签来标注重点 重点处理标注了

p标签有换行功能

em 标签: em 标签就是对文档中一小段文本进行重点强调。默认斜体, 可以修改。

<p>em标签是对这一段落加重的标签, 用em标签包裹的文本默认是斜体。
当然可以修改, 而且有利于网站的SEO</p>

*em*标签是对这一段落加重的标签，用*em*标签包裹的文本默认是斜体。当然可以修改，而且有利于网站的SEO

i 标签： i 标签是斜体字标签，和 em 一样，但是 em 可以优化 seo，i 标签不行

```
<P><i>--- i标签是斜体字标签</i></P>
```

浏览器显示为：

---*i标签是斜体字标签*

其实不推荐 i 标签使用，因为字体后面都是用 css 修改的。

Sub 标签： 字符上标

```
<p>2<sub>23</sub></p>    223
```

Sup 标签： 字符下标

```
<p> H<sub>2</sub>O </p>    H2O
```

这就是底数和指数显示标签

del 标签： 是删除字

```
<p> p<del>标签有换行功能</del></p>
```

浏览器显示为：

p标签有换行功能

超级链接标签 a 标签：



```

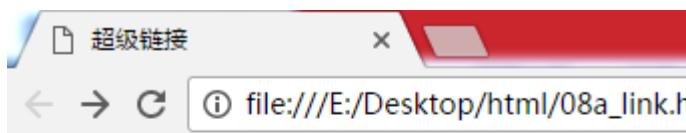
</head>
<body>
    <p>如何使用超级链接</p>
    <a href="https://www.baidu.com/">百度跳转</a>
</body>
</html>

```

超级链接标准代码是 [网页上跳转字符](要跳转的网址)
浏览器显示为：

href = ""这个里面写地址

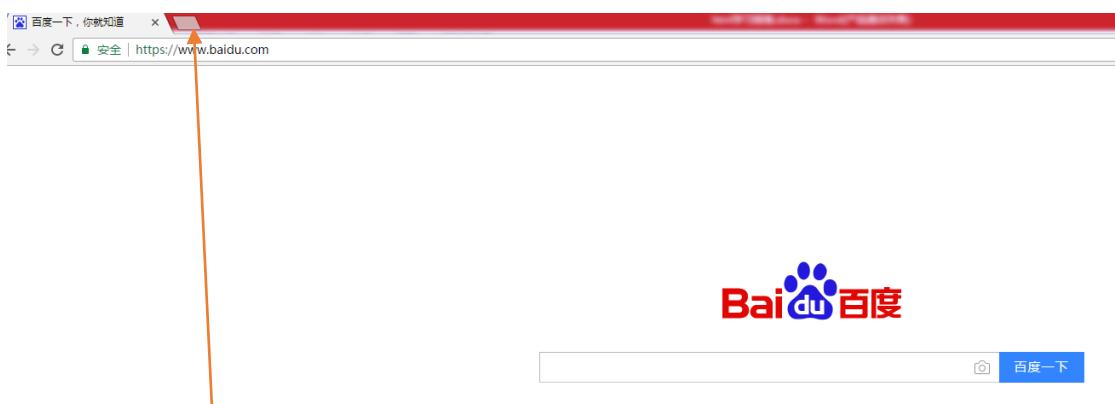
这是在网页上显示要跳转的文字



如何使用超级链接

百度跳转

点击百度跳转，进入百度网页。



但是这里有个问题就是，点击新的网页后浏览器 tab 栏没有增加新的 tab 地址。
这样就会造成，我想看以前的网页要不停的后退，很麻烦。

所以我们要用到 a 标签里面的 target 属性

target="_blank"属性就是在新的 tab 中打开网页

```

<body>
    <p>如何使用超级链接</p>
    <a href="https://www.baidu.com/">百度跳转</a>
    <hr>
    <a target="_blank" href="https://www.taobao.com/">
        这样跳转后在新tab栏中打开网站</a>
</body>
</html>

```

添加了这 target

浏览器显示为：

在新标签中打开的网站



锚点链接：

锚点链接是什么意思呢？

在一个网页里面有很多段图片，我点击平均薪资的时候

网页就跳到平均薪资的图片，但是记住网页没有变，还是在一页里面。这些图片都在一页里面。

比如我再点击移动开发



自动跳到移动开发区域。

这就是锚点，先给页面下面的每个图片区域标记个记号点，然后按照这个记号点跳转过去
这个还是用超级链接 [a 标签来做，只是 href 属性里面不是写入其他网页地址，而是本页面的图片记号。](#)

用 a 标签将字符包裹起来，该字符就有了跳转段落功能

```
<body>
  <p>锚点啊发发安抚安抚安抚啊收费额副歌部分个地方
    是的vb是吧是是是V是V东<a href="#maodian">北师大</a>许是V是V是V是
    睡吧睡吧不是不是不地方非得把对方电饭煲辅导班锚点啊发发安抚安抚
    安抚安抚安抚安抚安抚安抚安抚安抚安抚安抚安抚安抚
```

这个文本有点大，[href](#) 指定要跳转到 id=maodian 的地方，记得标记 maodian 前面要加#号

```
<br>
<br>
  <span id="maodian">是的vb是吧是是是V是V东北师大许是V是V是V是V是
    睡吧不是不是不地方非得把对方电饭煲辅导班暗示法撒发发给V
    </span>
```

这就是记号所在的段落

浏览器显示为：



锚点啊发发安抚安抚安抚啊收费额副歌部分个地方 是的vb是吧是是是V是V东北师大许是V是V是V是睡吧
安抚安抚啊收费额副歌部分个地方 是的vb是吧是是是V是V东北师大许是V是V是V是V是睡吧睡吧不是不是不地方非得
个地方 是的vb是吧是是是V是V东北师大许是V是V是V是睡吧睡吧不是不是不地方非得把对方电饭煲辅导班锚点啊
是V东北师大许是V是V是V是睡吧睡吧不是不是不地方非得把对方电饭煲辅导班锚点啊发发安抚安抚安抚安抚
是睡吧睡吧不是不是不地方非得把对方电饭煲辅导班锚点啊发发安抚安抚安抚安抚啊收费额副歌部分个地方锚点啊发发安抚
非得把对方电饭煲辅导班锚点啊发发安抚安抚安抚安抚啊收费额副歌部分个地方锚点啊发发安抚安抚安抚
点啊发发安抚安抚安抚安抚啊收费额副歌部分个地方锚点啊发发安抚安抚安抚安抚啊收费额副歌部
安抚啊收费额副歌部分个地方

也就是 span 标签指定的段落，所以 span 标签的用途就是这些，让 span 包裹的字符有特殊功能

跳转到这里了

是的vb是吧是是是V是V东北师大许是V是V是V是睡吧睡吧不是不是不地方非得把对方电饭煲辅导班暗示法撒发发给V付过款婆陪哦评价吗 偶看
是的vb是吧是是是V是V东北师大许是V是V是V是V是睡吧睡吧不是不是不地方非得把对方电饭煲辅导班锚点啊发发安抚安抚安抚安抚啊收
师大许是V是V是V是V是睡吧睡吧不是不是不地方非得把对方电饭煲辅导班锚点啊发发安抚安抚安抚安抚啊收费额副歌部分个地方锚点
吧不是不是不地方非得把对方电饭煲辅导班锚点啊发发安抚安抚安抚安抚啊收费额副歌部分个地方锚点啊发发安抚安抚安抚

但是为什么跳转到的行不是在浏览器第一行显示，浏览器第一行还是很多空白占据
那是因为这个字符的结尾是

标签最后一行.如果在结束的

```
收费额副歌部分个地方锚点啊发发安抚安抚安  
副歌部分个地方 是的vb是吧是是是V是V东北师大  
是不是不是不地方非得把对方电饭煲辅导班锚  
安抚安抚啊收费额副歌部分个地方锚点啊发发  
啊收费额副歌部分个地方 是的vb是吧是是是V  
是睡吧睡吧是不是不是不地方非得把对方电饭  
安抚安抚安抚啊收费额副歌部分个地方  
19 <br>  
50 <br>  
51 <br>  
52 <br>
```

这就让文字跳转到浏览器上面了，自己多实验。

HTML 有序列表和无序列表

首页 剧集 电影 综艺 音乐 少儿 来疯 直播 片库 资讯 拍客 纪实 公益 体育 汽车 科技 财经 娱乐 文化 动漫 搞笑 旅游 时尚 亲子 教育 游戏 更多▼

这种就是无序列表，每个标题没有序号。



常用的搜索引擎

1. 百度
2. Google
3. 搜狗
4. 雅虎
5. 360

这种就是有序列表，列表前面有序号。

无序列表标签

```
</head>
<body>
  <ul>
    <li>北京</li>
    <li>上海</li>
    <li>成都</li>
    <li>武汉</li>
  </ul>
</body>
</html>
```

每个是一个列表项

浏览器显示效果：



- 北京
- 上海
- 成都
- 武汉

有序列表标签

```
7 <body>
8     <!--ul标签用于生成有序列表-->
9     <ul>
0         <li>北京</li>
1         <li>上海</li>
2         <li>成都</li>
3         <li>武汉</li>
4     </ul>
5     <!--ol标签用于生成有序列表-->
6     <ol>
7         <li>北京</li>
8         <li>上海</li>
9         <li>成都</li>
0         <li>武汉</li>
1     </ol>
2 </body>
```

有序列表代码

浏览器显示为：



这里要注意标签里面除了标签，不要加任何东西，只能随便加加但是里面可以嵌套其他标签。标签里面也是。

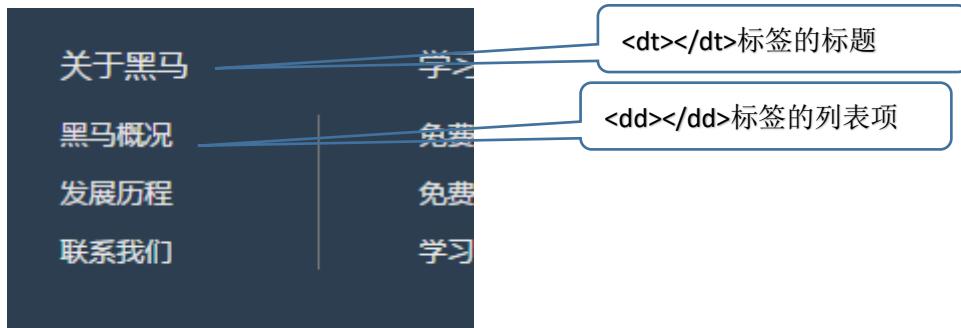
```
<body>
    <!--ul标签用于生成有序列表-->
    <ul>
        <li><span>北京</span><a href="#">你好</a></li>
        <li>上海</li>
        <li>成都</li>
        <li>武汉</li>
    </ul>
    <!--ol标签用于生成有序列表-->
```

- 北京 [你好](#)
- 上海
- 成都
- 武汉

1. 北京
2. 上海
3. 成都
4. 武汉

你看，嵌套了 span 标签和 a 标签，浏览器添加了你好的字段。

自定义标签`<dl></dl>`



```
</ul>

<dl>
    <dt>关于黑马</dt>
    <dd>黑马概况</dd>
    <dd>发展历程</dd>
    <dd>关于我们</dd>
</dl>
</body>
```

浏览器显示为:

关于黑马
黑马概况
发展历程
关于我们

记住`<dt></dt>`里面只能嵌套文本标签，比如 `span` 标签

如果你想定义两个列表

```
<!-- 自定义列表标签 -->
<dl>
    <dt>关于黑马</dt>
    <dd>黑马概况</dd>
    <dd>发展历程</dd>
    <dd>关于我们</dd>

    <dt>关于什么</dt>
    <dd>黑马什么</dd>
    <dd>发展什么</dd>
    <dd>我们什么</dd>
</dl>
```

关于黑马
黑马概况
发展历程
关于我们
关于什么
黑马什么
发展什么
我们什么

你只需要在 `dl` 标签里面添加第二级标题和列表项就是了，不需要再去加个 `dl` 标签

比如京东的网页，使用列表项

家用电器
手机 / 运营商 / 数码
电脑 / 办公
家居 / 家具 / 家装 / 厨具
男装 / 女装 / 童装 / 内衣
美妆个护 / 宠物
女鞋 / 箱包 / 钟表 / 珠宝
男鞋 / 运动 / 户外

家装节 > 家装城 > 居家日用 > 精品家具 > 家装建材 > 房屋达人 > 装修服务 >

厨具 > **【自营会场】** 烹饪锅具 | 刀剪菜板 | 厨房配件 | 水具酒具 | 餐具 | 茶具/咖啡具 | 保温杯

家纺 > 床品套件 | 被子 | 枕芯 | 蚊帐 | 凉席 | 毛巾浴巾 | 地毯地垫 | 床垫/床褥 | 毯子 | 抱枕靠垫 | 窗帘/窗纱
| 床单/床笠 | 被套 | 枕巾枕套 | 沙发垫套/椅垫 | 桌布/罩布 | 电热毯 | 布艺软饰

生活日用 > 收纳用品 | 雨伞雨具 | 净化除味 | 浴室用品 | 洗晒/熨烫 | 缝纫/针织用品 | 清洁工具

家装软饰 > **【家居软饰】** 装饰字画 | 装饰品摆件 | 手工/十字绣 | 相框/照片墙 | 墙贴/装饰贴 | 花瓶花艺 | 香薰蜡烛 | 节庆饰品
| 钟饰 | 帘艺隔断 | 创意家居 | 保暖防护

怎么京东的格式列表是横排的，我写的是竖排的
而且还不好看，这个就要靠 CSS 去修改它了。

```
► <dt class="cate_detail_tit" clstag="h|keycount|2016|0604c03">...</dt>
▼ <dd class="cate_detail_con" clstag="h|keycount|2016|0604d03">
    <a href="//list.jd.com/list.html?cat=1620,13780" class="cate_detail_con_lk" target="_blank">收纳用品</a>
    <a href="//list.jd.com/list.html?cat=1620,1624,1656" class="cate_detail_con_lk" target="_blank">雨伞雨具</a>
    <a href="//list.jd.com/list.html?cat=1620,1624,1661" class="cate_detail_con_lk" target="_blank">净化除味</a>
    <a href="//list.jd.com/list.html?cat=1620,1624,1657" class="cate_detail_con_lk" target="_blank">浴室用品</a>
    <a href="//list.jd.com/list.html?cat=1620,1624,1660" class="cate_detail_con_lk" target="_blank">洗晒/熨烫</a>
    <a href="//list.jd.com/list.html?cat=1620,1624,1658" class="cate_detail_con_lk" target="_blank">缝纫/针织用
```

这就是京东生活用品栏的 dt 标签

dl 列表标签案例

The screenshot shows the Ule.com website interface. At the top, there's a yellow header bar with links for '登录' and '注册'. Below it is a navigation bar with tabs for '中国' (selected), '美国', and '加拿大'. A search bar is also present. The main content area features a large blue promotional banner for '话费充值97.5元充100' with an '立即下载' button. Below the banner, there's a sidebar titled '美国馆推荐' containing a grid of product categories: 内衣, 汗蒸服, 卫衣, 打底裤, 大衣, 长裤; 职业装, 连体服, 短裤, 牛仔裤, 半身裙, 袜子; 连衣裙, 马甲, 运动装, T恤吊带衫, 上衣, 孕妇装. At the bottom of the page, there's a footer bar with links for '首页', '中国馆', '美国馆', and '加拿大馆'.

这种页面首先我们要分层

This screenshot is identical to the one above, showing the Ule.com website with its navigation bar, search function, and promotional banners. The main difference is that the content below the banner is now divided into distinct layers or sections, as indicated by the text above.

1.这是图片

首页 > 中国馆 > 女装/女士精品 > 在女装/女士精品下的所有商品

2.这是直接写字体

美国馆推荐

内衣	泳装	卫衣	打底裤	大衣	长裤
职业装	连体服	短裤	牛仔裤	半身裙	袜子
连衣裙	马甲	运动装	T恤吊带衫	上衣	孕妇装

3.这是列表使用

```
<!DOCTYPE html>
<html lang="zn_CH">
<head>
    <title></title>
    <meta charset=UTF-8>
</head>
<body>
    
    <br/><br/>
    首页>中国馆>女装/女士精品>所有商品
    <br/><br/>
    <dl>
        <dt>美国馆推荐</dt>
        <dd>内衣 &nbsp;&nbsp;泳装 &nbsp;&nbsp;游泳裤</dd>
        <dd>奶茶杯 &nbsp;&nbsp;苹果酱酱酱 &nbsp;游泳裤</dd>
    </dl>
</body>
</html>
```

br 就是换行



全球集市

中国

美国

加拿大

输入店铺名称

首页



中国馆



话费充值97.5元充100, 邮乐!

首页>中国馆>女装/女士精品>所有商品

dt 标签就是顶格

dd 标签就是缩进

美国馆推荐

内衣 泳装 游泳裤
奶茶杯 苹果酱酱酱 游泳裤

表格标签 table

<table></table>表示一个表格
<tr></tr>表示一整行
<td></td>表示一行里面的一格，多个 td 一行里面就有很多格

```
<head>
    <meta charset="utf-8">
    <title>table标签</title>
</head>
<body>
    <table border="1">
        <tr>
            <td>基本参数</td> <td>1.4T 帕沙特</td>
        </tr>

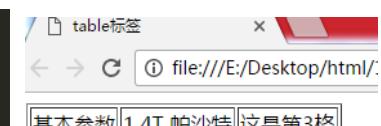
    </table>
</body>
</html>
```

一个 tr 标签，两个 td 标签，就是一行两格

浏览器显示为：



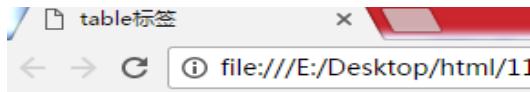
```
<body>
    <table border="1">
        <tr>
            <td>基本参数</td> <td>1.4T 帕沙特</td> <td>这是第3格</td>
        </tr>
    </table>
```



一个 tr 标签，三个 td 标签，就是一行三格，如果想多来几格，自己加 tr 标签

```
<body>
    <table>
        <tr>
            <td>基本参数</td> <td>1.4T 帕沙特</td> <td>这是第3格</td>
        </tr>
        <tr>
            <td>增加了个tr</td> <td>就增加了一行</td> <td>这是第3格</td>
        </tr>
    </table>
```

两个 tr 标签，表示两行，每行有三个 td 标签，表示每行三格，
浏览器显示为：



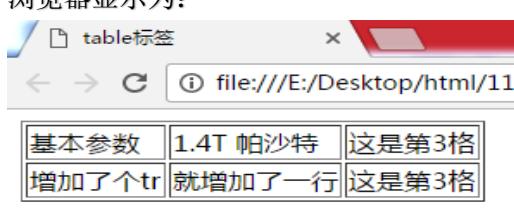
基本参数 1.4T 帕沙特 这是第3格
增加了个tr 就增加了一行 这是第3格
没有 border 确实很丑

```
<body>
  <table border="1">
    <tr>
      <td>基本参数</td> <td>1.4T 帕沙特</td> <td>这是第3格</td>
    </tr>

    <tr>
      <td>增加了个tr</td><td>就增加了一行</td><td>这是第3格</td>
    </tr>
  </table>
</body>
</html>
```

增加了 border

浏览器显示为：

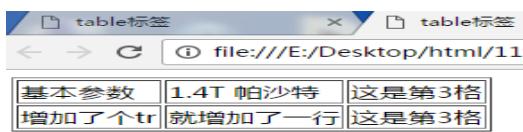


有边框好漂亮

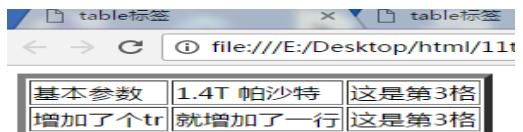
```
<table border="0">
```

基本参数 1.4T 帕沙特 这是第3格
增加了个tr 就增加了一行 这是第3格

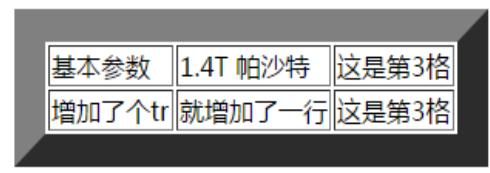
```
<table border="1">
```



```
<table border="5">
```



```
<table border="20">
```



Border="参数" 参数大小决定了表格外边框的粗细，但是以后用 CSS 来修改了，所以这里只是演示一下而已，没必要用 border 去设置边框大小。

这种直接在 body 标签里面加 table 标签有个不爽的地方，语意不是很明确。

<thead></thead>标签是做表头

<tbody></tbody>标签放正常的文本数据

```
<body>
  <table border="20">
    <thead>
      <tr>
        <th>姓名</th><th>年龄</th><th>身高</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>基本参数</td> <td>1.4T 帕沙特</td> <td>这是第3格</td>
      </tr>

      <tr>
        <td>增加了个tr</td><td>就增加了一行</td>这是第3格</td>
      </tr>
    </tbody>
  </table>
</body>
```

浏览器显示为：

姓名	年龄	身高
基本参数	1.4T 帕沙特	这是第3格
增加了个tr	就增加了一行	这是第3格

所以用<thead>和<tbody>是为了加强文本的语意，方便浏览器 SEO，当然你不用 thead 和 tbody 也可以，看你怎么想了，我是觉得意思清楚些。

合并行单元格标签

基本参数	1. 4T帕沙特	这是第三格
这是第二排第1格	这是第二排第2格	这是第二排第3格
这是第三排第1格	这是第三排第2格	这是第三排第3格

我们要合并第 1 行

```

</head>
<body>
    <table border="1">
        <tr>
            <td rowspan="1">基本参数</td><td>1.4T帕沙特</td><td>这是第三格</td>
        >
    </tr>
    <tr>
        <td>这是第二排第1格</td><td>这是第二排第2格</td><td>
            这是第二排第3格</td>
        </td>
    </tr>
    <tr>
        <td>这是第三排第1格</td><td>这是第三排第2格</td><td>
            这是第三排第3格</td>
        </td>
    </tr>
</table>

```

在 td 后面加 rowspan 属性就可以了，但是这里 rowspan 不能选择 1，应该选择 2

基本参数	1. 4T帕沙特	这是第三格
这是第二排第1格	这是第二排第2格	这是第二排第3格
这是第三排第1格	这是第三排第2格	这是第三排第3格

浏览器显示

```

<body>
    <table border="1">
        <tr>
            <td rowspan="2">基本参数</td><td>1.4T帕沙特</td><td>这是第三格</td>
        >
    </tr>
    <tr>
        <td>这是第二排第1格</td><td>这是第二排第2格</td><td>
            这是第二排第3格</td>
        </td>
    </tr>
    <tr>
        <td>这是第三排第1格</td><td>这是第三排第2格</td><td>
            这是第三排第3格</td>
        </td>
    </tr>
</table>

```

基本参数	1. 4T帕沙特	这是第三格
这是第二排第1格	这是第二排第2格	这是第二排第3格
这是第三排第1格	这是第三排第2格	这是第三排第3格

浏览器显示

在某一行 td 里面加 rowspan 属性，就以这一行为标准向下合并

比如我想合并三行单元格，就在 rowspan="3"

```

<body>
    <table border="1">
        <tr>
            <td rowspan="3">基本参数</td><td>1.4T帕沙特</td><td>这是第三格</td>
        >
    </tr>
    <tr>
        <td>这是第二排第1格</td><td>这是第二排第2格</td><td>
            这是第二排第3格</td>
        </td>
    </tr>
    <tr>
        <td>这是第三排第1格</td><td>这是第三排第2格</td><td>
            这是第三排第3格</td>
        </td>
    </tr>
</table>

```

1. 4T帕沙特	这是第三格	
这是第二排第1格	这是第二排第2格	这是第二排第3格
这是第三排第1格	这是第三排第2格	这是第三排第3格

如果你想合并第 2 列的三行单元格，那么久在第 2 列的 td 标签里面加 rowspan

```

<body>
  <table border="1">
    <tr>
      <td rowspan="3">基本参数</td><td rowspan="3">1.4T帕沙特</td><td>
        这是第三格</td>
    </tr>
    <tr>
      <td>这是第二排第1格</td><td>这是第二排第2格</td><td>
        这是第二排第3格</td>
    </tr>
    <tr>
      <td>这是第三排第1格</td><td>这是第三排第2格</td><td>
        这是第三排第3格</td>
    </tr>
  </table>
</body>

```

浏览器显示为

基本参数	1.4T帕沙特	这是第三格		
		这是第二排第1格	这是第二排第2格	这是第二排第3格
		这是第三排第1格	这是第三排第2格	这是第三排第3格

如果你想从第 2 排开始合并单元格。

```

<body>
  <table border="1">
    <tr>
      <td>基本参数</td><td rowspan="2">1.4T帕沙特</td><td>这是第三格</td>
    </tr>
    <tr>
      <td rowspan="2">这是第二排第1格</td><td>这是第二排第2格</td><td>
        这是第二排第3格</td>
    </tr>
    <tr>
      <td>这是第三排第1格</td><td>这是第三排第2格</td><td>
        这是第三排第3格</td>
    </tr>
  </table>

```

浏览器显示为：

基本参数	1.4T帕沙特	这是第三格	
这是第二排第1格		这是第二排第2格	这是第二排第3格
	这是第三排第1格	这是第三排第2格	这是第三排第3格

你想合并哪一列就进入哪一列的 td，想合并哪一行就在哪一行 tr 里面 td 加上 rowspan
colspan 属性是一个单元格合并一行两格，或者三格

```

<body>
  <table border="1">
    <tr>
      <!--一个单元格占用两行-->
      <td rowspan="2">基本参数</td><td>1.4T帕沙特</td><td>这是第三格</td>
    </tr>
    <tr>
      <td>这是第二排第1格</td><td>这是第二排第2格</td><td>
        这是第二排第3格</td>
    </tr>
    <tr>
      <!--一个单元格占用两列-->
      <td colspan="2">这是第三排第1格</td><td>这是第三排第2格</td><td>
        这是第三排第3格</td>
    </tr>
  </table>

```

基本参数	1.4T帕沙特	这是第三格	
这是第二排第1格		这是第二排第2格	这是第二排第3格
	这是第三排第1格	这是第三排第2格	这是第三排第3格
这是第三排第1格		这是第三排第2格	这是第三排第3格

div 标签的使用

div 是块级标签

```
View Goto Tools Project Preferences Help
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta charset="utf-8">
5     <title>span_div标签</title>
6 </head>
7 <body>
8     <div>第一个div</div>
9     <div>第二个div</div>
10    <div>第三个div</div>
11
12
13 </body>
14 </html>
```

浏览器显示为



第一个div
第二个div
第三个div

你看一个 div 标签独占一行.

如果在第一行 div 里面加一个换行标签

The diagram shows the HTML code with annotations:

```
<body>
    <div>
        第一个div
        <br>
        第一个div
    </div>
    <div>第二个div</div>
    <div>第三个div</div>
</body>
```

- An orange arrow points from the word "第一个div" in the first line of the code to the first "div" tag in the browser output.
- A blue arrow points from the word "第一个div" in the second line of the code to the second "div" tag in the browser output.
- A red arrow points from the word "第一个div" in the third line of the code to the third "div" tag in the browser output.
- The browser output shows three lines of text: "第一个div", "第一个div", and "第二个div". The first two lines are grouped together under the first div tag in the code, while the third line is under the second div tag.

你看第二行的第一个 div 标签还是单独占用了浏览器一行.

单色 span 标签就不一样

span 标签的使用

```
<body>
    <span>第一个span</span>
    <span>第二个span</span>
    <span>第三个span</span>
</body>
```

浏览器显示为



第一个span 第二个span 第三个span

你看不管多少个 **span** 标签,在浏览器里面都是在一行里面挨着顺序来,不会像 **div** 标签那样独占一行.

像 **div** 这种块级标签有: **p , div , ul , ol ,dl,li ,dt,h1~h6**

像 **span** 行级标签有: **span , strong , em,**

表单的使用

什么是表单? 就是叫你注册账号, 填写用户信息就是表单。

就像暴雪战网注册这样, 填写一些信息, 确认后传送给后台服务器。

所以我们会用到 **form** 标签

form 标签使用

<form action="#"></form> 所以表单的标签必须包裹在 form 标签当中

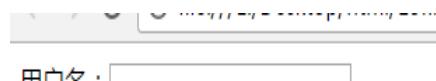
action 属性： 指向后台的一个地址

```
<body>
<!--所有表单标签必须包裹在form标签中-->
<!--action属性： 指向后台的一个地址-->
<form action="#">
    用户名: <input type="text" value="" name="">

    </form>
</body>
```

<input>标签是在网页中建立一个输入类型的标签

Input 中的 type 属性： 决定了该标签的格式。



浏览器显示为一个填写框，这就是 type 的 text 属性。

```
<body>
<!--所有表单标签必须包裹在form标签中-->
<!--action属性： 指向后台的一个地址-->
<form action="#">
    用户名: <input type="text" value="123" name="">

    </form>
</body>
```

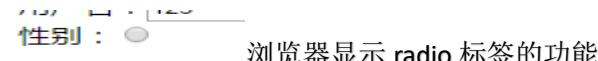
Input 中的 value 属性： 是在该填空表中显示数字，类似于密码提示那种，你点击该填空表就消失了。



浏览器显示，这个 123 就是 value 属性

性别： <input type="radio" value="" name="">

radio 标签： 就是在浏览器里面显示一个单选按钮。这就是 type 标签的另一个功能



浏览器显示 radio 标签的功能

```
性别： 男<input type="radio" value="" name="">
      女<input type="radio" value="" name="">
```

性别：男 女 这就是 `radio` 单选标签的功能，但是有个问题！！

你发现虽然浏览器显示了男女可选，但是不是单选是多选，这是为什么？

所以要在 **input 标签** 里面加 `name` 属性，两个标签 `name` 里面的内容相同就为一组。

```
性别：男<input type="radio" value="" name="nnn">
      女<input type="radio" value="" name="nnn">
```

性别：男 女 浏览器显示，这就证明了在一组 `name` 的标签可以单选了。

如果不在一组 `name` 呢？

```
性别：男<input type="radio" value="" name="nnn">
      女<input type="radio" value="" name="mmm">
```

性别：男 女 浏览器显示为，不在一组 `name` 又是多选了

我们来写个多 `radio` 标签应用

```
性别：男<input type="radio" value="" name="nnn">
      女<input type="radio" value="" name="nnn">
      <br>
```

```
性别：男<input type="radio" value="" name="mmm">
      女<input type="radio" value="" name="mmm">
```

性别：男 女

性别：男 女

浏览器显示为，相同的 `name` 之间可以单选

<select></select> 下拉列表框标签

但是要下拉多少个列表呢？我们要用 `<option> </option>` 标签

```
<form action="#">

  <select name="" id="">
    <option value="1">公历</option>
    <option value="2">农历</option>
  </select>
</form>
```



浏览器显示下拉列表

所以下拉列表标签都必须包裹在 `select` 标签里面，`select` 必须包裹在 `form` 表单标签里面。
如果给 `select` 标签加上 `multiple` 属性

```
<form action="#">

<select name="" id="">
    <option value="1">公历</option>
    <option value="2">农历</option>
</select>
<br>
<br>
<select name="" id="" multiple="multiple">
    <option value="1">公历</option>
    <option value="2">农历</option>
</select>
</form>
```

公历 ▼



浏览器显示为，`multiple` 属性将标签修改为直接下拉显示，然后选择，而不是上面那种点击下拉，然后选择。

有时候我想让下拉列表里面的其他选项在默认状态

```
<form action="#">
<select name="" id="">
    <option value="1" >公历</option>
    <option value="2" selected="true">农历</option>
    <option value="2" >新历</option>
</select>
```

默认改为农
历了

农历 ▼

浏览器默认显示的下拉列表。

我想默认下拉列表的哪一行，我只需要在该行添加 `selected` 属性位 `true` 就可以了。

```
<form action="#">
<select name="" id="">
    <option value="1" selected="true">公历</option>
    <option value="2" selected="true">农历</option>
    <option value="2" selected="true">新历</option>
</select>
```

新历 ▼

如果下拉列表每个标签都有 `selected` 的话就选择下拉列表最后一行为默认。

如果我想在网页里面放入一些文本怎么办？

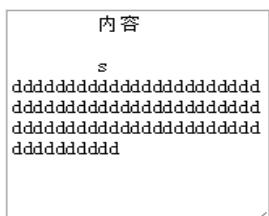
<textarea> </textarea> **文本域标签**

```
<form action="#">
<!--文本域--&gt;
&lt;textarea cols="20" rows="10"&gt;
内容

&lt;/textarea&gt;
&lt;/form&gt;</pre>
```

cols 为文本
边框长度

rows 为文本
边框高度



浏览器显示为，可以随便在文本里面加文字。

如果你想文本边框大点，就修改 cols 和 rows 属性。

```
<form action="#">
<!--文本域--&gt;
&lt;textarea cols="50" rows="50"&gt;
内容

&lt;/textarea&gt;
&lt;/form&gt;</pre>
```

内容
asardard

浏览器显示为 50 长 50 高的文本边框。可以写好多字哦

如果我想网页有多选功能怎么办？

input 标签里面把 type 改了。

```
<input type="checkbox">  
<input type="checkbox">  
<input type="checkbox">  
</form>
```



这就是多选功能

所以要注意了<input type = ""> type 里面的类型不一样，网页显示的也不一样。

type="checkbox" 就是多选

type="text" 文本框

type="radio" 就是单选

如果你想打开网页多选里面有个默认选项，那么你就在<input type>的哪一行加上 checked

```
<input type="checkbox">  
<input type="checkbox">  
<input type="checkbox" checked="">  
</form>
```



浏览器显示多选默认功能

如果我要给网页输入密码怎么办？那就要用密码框

在<input type> 类型换成 password

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

密码 :

这就是密码框

<label></label>标签：该标签是指明哪一个字符，在鼠标移动到这个字符的时候显示字符的特殊颜色或者功能。

```
<label for="lab"></label> label 标签: for 属性是指定要去那个标签里面去修改它的字体形态
<td>性别</td> <td>男<input type="radio" name="nnn" id="lab">
<label for="lab">男</label>
 &nbsp;&nbsp;女<input type="radio" name="nnn"></td>
```

性别 男 ♂ 男 女 ♀

你看前面那个男是一般的字符，鼠标移动过去没什么反应，但是后边这个男是 label 标签增加的属性，所以鼠标移动过去有反应。

如果你觉得 label 标签加 id，然后使用的其它标签加 id 太麻烦，那么你就直接用 label 标签去包裹你想要处理的标签文字。

```
<tr>
  <td>性别</td> <td><label>男<input type="radio" name="nnn"
id="lab"></label>
  &nbsp;&nbsp;女<input type="radio" name="nnn"></td>
```

</tr>

性别 男 ♂ 女 ♀

这就是 label 标签的使用

重置按钮

```
<head>
  <title></title>
  <meta charset=UTF-8>
</head>
<body>
  <form>
    <tr>
      <td>昵称</td> <td><input type="text" name="user"></td>
    </tr>
    <input type="reset" value="重置按钮"/>
  </form>
</body>
</html>
```

重置按钮一定要在 form
里面才有效果

重置按钮 type = "reset"

昵称

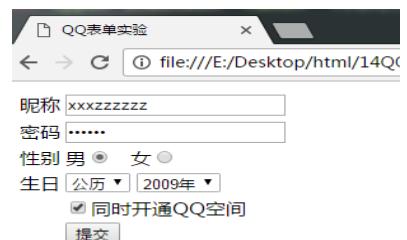
我在页面输入了 111，然后
点击重置按钮

昵称 页面数据就清空了，要你重新写。

QQ 用户注册案例：

```
<table>
    <tr>
        <td>昵称</td> <td><input type="text" name=""></td>
    </tr>
    <tr>
        <td>密码</td> <td><input type="password" name=""></td>
    </tr>
    <tr>
        <td>性别</td> <td><label>男<input type="radio" name="nnn" id="lab"></label>
        &nbsp;&nbsp;女<input type="radio" name="nnn"></td>
    </tr>
    <tr>
        <td>生日</td>
        <td>
            <select>
                <option value="">农历</option>
                <option value="">公历</option>
            </select>
            <select>
                <option value="">2005年</option>
                <option value="">2006年</option>
                <option value="">2007年</option>
                <option value="">2008年</option>
                <option value="">2009年</option>
                <option value="">2010年</option>
                <option value="">2011年</option>
                <option value="">2012年</option>
                <option value="">2013年</option>
                <option value="">2014年</option>
            </select>
        </td>
    </tr>
    <tr>
        <td></td>
        <td><input type="checkbox" name="">同时开通QQ空间</td>
    </tr>
    <tr>
        <td></td>
        <td><input type="submit" name="注册"></td>
    </tr>
</table>
</form>
```

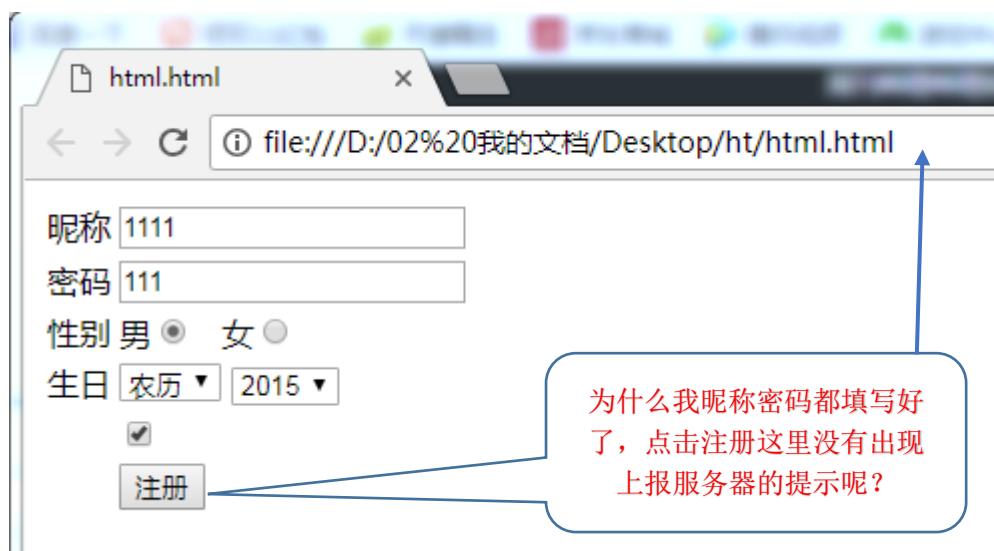
浏览器显示为



在文本框表单框布局上，要用<tr>和<td>标签先布好局，然后再里面写新的标签。

from 标签上报服务器功能

```
html.html
<body>
  <table>
    <tr>
      <td>昵称</td> <td><input type="text" name="user"></td>
    </tr>
    <tr>
      <td>密码</td> <td><input type="text" name="passwd"></td>
    </tr>
    <tr>
      <td>性别</td> <td><label>男<input type="radio" name="nnn" id="lab"></label>
      &nbsp;&nbsp;女<input type="radio" name="nnn"></td>
    </tr>
    <tr>
      <td>生日</td>
      <td>
        <select>
          <option value="">农历</option>
          <option value="">公历</option>
        </select>
        <select>
          <option value="">2015</option>
          <option value="">2014</option>
          <option value="">2013</option>
          <option value="">2012</option>
        </select>
      </td>
    </tr>
    <tr>
      <td></td>
      <td><input type="checkbox" name=""></td>
    </tr>
    <tr>
      <td></td>
      <td><input type="submit" value = "注册"></td>
    </tr>
  </table>
</body>
</html>
```



这是因为你代码里面没有加 from

```

<body>
<form> 在整个注册表单开始处这里加 from
    <table>
        <tr>
            <td>昵称</td> <td><input type="text" name="user"></td>
        </tr>
        <tr>
            <td>密码</td> <td><input type="text" name="passwd"></td>
        </tr>
        <tr>
            <td>性别</td> <td><label>男<input type="radio" name="boy" id="lab"></label>
            &nbsp;&nbsp;女<input type="radio" name="girl"></td>
        </tr>
        <tr>
            <td>生日</td>
            <td>
                <select>
                    <option value="">农历</option>
                    <option value="">公历</option>
                </select>
                <select>
                    <option value="">2015</option>
                    <option value="">2014</option>
                    <option value="">2013</option>
                    <option value="">2012</option>
                </select>
            </td>
        </tr>
        <tr>
            <td></td>
            <td><input type="checkbox" name=""></td>
        </tr>
        <tr>
            <td></td>
            <td><input type="submit" value = "注册"></td>
        </tr>
    </table>
</form> 在整个注册表单结束处用/from 结尾

```



分析地址

file:///D:/02%20%E6%88%91%E7%9A%84%E6%96%87%E6%A1%A3/Desktop/ht/html.html?user=111&passwd=111&boy=on

User=111 : User 就是名称, 111 就是这个名称对应的值

Passwd = 111 也是一样的

boy = on: boy 是名称, on 是值

中间的&就是分割符

现在又有一个问题了

我页面这么多数据, 为什么只有三个名称提交了

```

<form>
  <table>
    <tr>
      <td>昵称</td> <td><input type="text" name="user"></td>
    </tr>
    <tr>
      <td>密码</td> <td><input type="text" name="passwd"></td>
    </tr>
    <tr>
      <td>性别</td> <td><label>男<input type="radio" name="boy" id="lab"></label>
      &nbsp;&nbsp;女<input type="radio" name="girl"></td>
    </tr>
    <tr>
      <td>生日</td> <td>只有这里给 name 属性加了名称</td>
    </tr>
    <td>
      <select>
        <option value="">农历</option>
        <option value="">公历</option>
      </select>
      <select>
        <option value="">2015</option>
        <option value="">2014</option>
        <option value="">2013</option>
        <option value="">2012</option>
      </select>
    </td>
  </tr>
  <tr>
    <td></td>
    <td><input type="checkbox" name=""></td>
  </tr>
  <tr>
    <td></td>
    <td><input type="submit" value = "注册"></td>
  </tr>
  </table>
</form>

```

如果要提交更多数据需要在每一个标签那里加 name

```

<form>
  <table>
    <tr>
      <td>昵称</td> <td><input type="text" name="user"></td>
    </tr>
    <tr>
      <td>密码</td> <td><input type="text" name="passwd"></td>
    </tr>
    <tr>
      <td>性别</td> <td><label>男<input type="radio" name="boy" id="lab"></label>
      &nbsp;&nbsp;女<input type="radio" name="girl"></td>
    </tr>
    <tr>
      <td>生日</td>
      <td>
        <select name="huangli">
          <option value="nong">农历</option>
          <option value="gong">公历</option>
        </select>
        <select name="number">
          <option value="2015">2015</option>
          <option value="2014">2014</option>
          <option value="2013">2013</option>
          <option value="2012">2012</option>
        </select>
      </td>
    </tr>
    <tr>
      <td></td>
      <td><input type="checkbox" name="box"></td>
    </tr>
    <tr>
      <td></td>
      <td><input type="submit" value = "注册"></td>
    </tr>
  </table>
</form>

```

因为只有这里给 name 属性加了名称

只有这里给 name 属性加了名称

列表标签只能在 select 这里加 name

给 name 属性的值只能在 option 的 value 上加

Input 标签直接加 name 就可以

html.html?user=111&passwd=222&girl=on&huangli=nong&number=2015&box=on

昵称
 密码
 性别 男 女
 生日 农历

现在有个问题，就是我现在点击注册提交数据，但是数据没有指定提交到哪个网站的服务器，所以我们现在要加入提交网站的服务器地址。

action 属性：填入你要提交的网站地址

我们先用两个 html 文件来模拟提交过程



同级目录下有两个 html 文件，我的 action 是写在 html 文件下的

```
<form action="html2.html">
  <table>
    <tr>
      <td>昵称</td> <td><input type="text" name="user"></td>
    </tr>
    <tr>
      <td>密码</td> <td><input type="password" name="password"></td>
    </tr>
  </table>
</form>
```

直接在 from 头加 action，然后指定提交的地址，我直接提交给本目录的 html2 文件

html.html

昵称
 密码
 性别 男 女
 生日 农历

点击注册

页面转入到 html2 了，数据提交到这里

html2.html?user=111&passwd=111&boy=on&huangli=nong&number=2015&box=on

这是html2文件

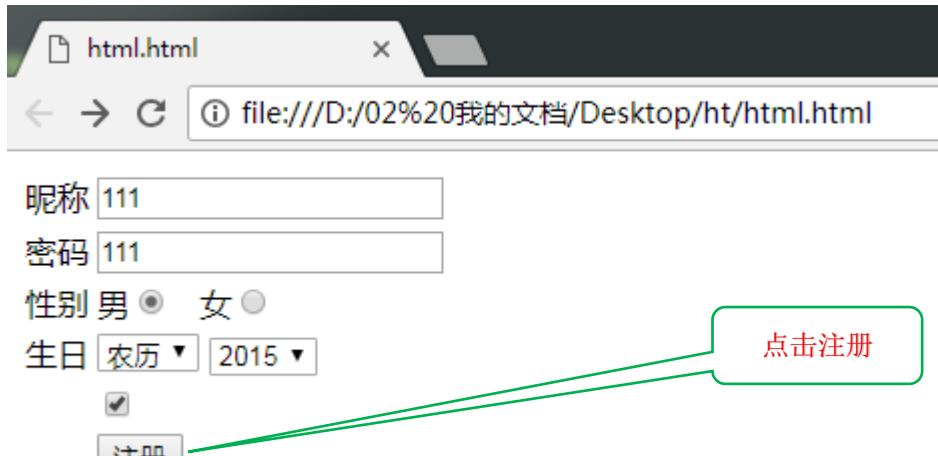
from 标签默认的是 get 请求，get 请求有几个问题

1. 地址是明文传输

① file:///D:/02%20我的文档/Desktop/ht/html2.html?user=111&passwd=111&boy=on&huangli=nong&number=2015&box=on

2.get 请求有数据量大小的限制，不能超出地址栏的容纳范围，所以 get 一般提交小数据
改成 post 请求

```
<form action="html2.html" method="post">
  <table>
    <tr>
      <td>昵称</td> <td><input type="text" name="user"></td>
    </tr>
    <tr>
      <td>密码</td> <td><input type="text" name="passwd"></td>
    </tr>
```



Post 相比 get 请求的好处

1.post 提交的数据不是明文，密码这些在地址栏上看不见

2.post 提交的数据量可以很大，不是浏览器地址栏容量限制，可以用来做文件上传

一般 get 可以在做网页过程中检查你提交数据格式对不对，项目做好发布网页时改成 post

From 标签里用图片当做按钮来提交数据

```
52           </select>
53           </td>
54     </tr>
55     <tr>
56       <td></td>
57       <td><input type="checkbox" name="box"></td>
58     </tr>
59     <tr>
60       <td></td>
61       <td><input type="submit" value="注册"></td>
62     </tr>
63   </table>
64   <input type="image" src="timg.jpg">
```

在 from 里加个有 image 属性的
input 就行了，图片就在当前目录下



勾选框和选择标签如何设置页面打开默认选择

昵称

密码

性别 男 女

生日 农历 2015

我们打开网页，如果不点击选择属性，默认是不选择的

```

<tr>
    <td>性别</td> <td><label>男<input type="radio" name="boy" id="lab" checked="checked"></label>
    &nbsp;&nbsp;女<input type="radio" name="girl"></td>
</tr>
<tr>
    <td>生日</td>
    <td>
        <select name="huangli">
            <option value="nong">农历</option>
            <option value="gong">公历</option>
        </select>
        <select name="number">
            <option value="2015">2015</option>
            <option value="2014">2014</option>
            <option value="2013">2013</option>
            <option value="2012">2012</option>
        </select>
    </td>
</tr>
<tr>
    <td></td>
    <td><input type="checkbox" name="box" checked="checked"></td>
</tr>

```

在你使用默认选择功能的 input 标签里加 checked="checked"

只要在 input 里面加了，选择标签就有默认功能，不管哪个 input 标签都可以加

昵称

密码

性别 男 女

生日 农历 2015

你看打开网页就要默认选择

表单分组

为什么要表单分组？

```
<body>
<form action="">

    用户名: <input type="text" name="">
    <br>
    密码:   <input type="text" name="">

    用户名2: <input type="text" name="">
    <br>
    密码2:   <input type="text" name="">
</form>
</body>
```

浏览器显示为下面这样，看起来是不是很乱，如果你想用哪些复杂的标签让它对齐也可以，但是很麻烦。

用户名：
密码： 用户名2：
密码2：

<fieldset></fieldset>标签：该标签让你对浏览器想要的部分进行分组。

```
<fieldset>
    用户名: <input type="text" name="">
    <br>
    密码:   <input type="text" name="">
</fieldset>
```

浏览器页面的层次感就出来了

用户名：
密码：

用户名2：
密码2：

如果你想给这个分组的块加上名字，请在分组里面加**<legend></legend>**标签

```
<fieldset>
<legend>用户组1</legend>
    用户名: <input type="text" name="">
    <br>
    密码:   <input type="text" name="">
</fieldset>
```

我们将两个 input 填空块都分开。

```
<fieldset>
<legend>用户组1</legend>
    用户名: <input type="text" name="">
    <br>
    密码:   <input type="text" name="">
</fieldset>

<fieldset>
<legend>用户组2</legend>
    用户名2: <input type="text" name="">
    <br>
    密码2:   <input type="text" name="">
</fieldset>
```

两个 fieldset 标签

用户组1
用户名：
密码：

用户组2
用户名2：
密码2：

表单综合实例：

```
</head>
<body>
<form action="">
<input type="text" name="" value="111">
<input type="radio" name="">
<input type="checkbox" name="">
<input type="button" name="" value="普通按钮">
<input type="submit" name="" value="提交表单">
<input type="reset" name="" value="重置">
</form>
</body>
</html>
```

<input type="选择那种控件" value="打开网页默认状态"
 text 填写框
 radio 选择
 checkbox 勾选
 button 按钮
 submit 提交服务器
 reset 清空前面操作

浏览器显示为：



操作后是下面这样



重置后又是浏览器默认样式



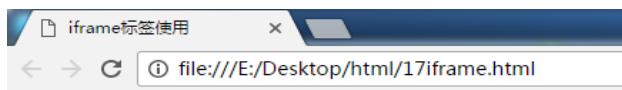
如果我想在我的网页里面显示其他网页的内容怎么办？

<iframe></iframe>内联标签

```
</head>
<body>
<h1>内联标签使用</h1>
<iframe src="https://www.taobao.com/"></iframe>
</body>
```

在 `iframe` 标签里的 `src` 属性里面写要展现的地址

浏览器显示为：



内联标签使用



增加框框的宽度和长度

```
</head>
<body>
<h1>内联标签使用</h1>
<iframe width="1000" height="1000" src="https://www.taobao.com/"></iframe>
</body>
```

加上 width 和 height 属性就可以了，填入长 1000 宽 1000

浏览器显示为：



如果要多加几个内联标签

```
<h1>内联标签使用</h1>
<iframe width="400" height="400" src="https://www.taobao.com/"></iframe>
<iframe width="400" height="400" src="https://www.jd.com/"></iframe>
</body>
```



这种内联标签也不利于网站的
SEO

但是现在都不用 iframe 标签，现在都有 div 来做

<base target="_blank"> base 标签就是让你的 a 标签打开网页多一个 table 框

```
<a href="https://www.taobao.com/">淘宝</a>
<a href="https://www.baidu.com/">百度</a>
```

没有 base 标签浏览器显示为：



点击淘宝



你看没有增加 tab 拦打开网页。

如果有 base

```
<base target="_blank">
<a href="https://www.taobao.com/">淘宝</a>
<a href="https://www.baidu.com/">百度</a>
</body>
```

打开浏览器显示为：

点击后对一个 tab 栏



淘宝网

其实 a 标签加属性也可以做到这个效果，但是 base 标签可以将下面的 a 标签全部包裹为 tab 样式功能，不用每个 a 标签都去加一句，方便而已。

<pre></pre>让文本的空格不合并，有多少空格就留多少空格

```
<pre>
    我这里有空格的
        真的有空格
</pre>
</body>
```

我这里有空格的
真的有空格

字符实体

我想在网页上显示一些符号，比如<HTML5> 和<p>这些字符

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>字符实体</title>
</head>
<body>

    <HTML5>
    <p>这是p标签的写法</p>

</body>
</html>
```

浏览器并没有显示出这些字符



如果你想在浏览器显示这些字符就必须加入一些东西。

```
<body>

    <HTML5>
    <p>这是p标签的写法</p>

    &lt;HTML5&gt;
    &lt;p&gt;必须用特殊字符标记来取代HTML标签特殊符号&lt;/p&gt;

</body>
```

这样浏览器才能显示出来

这是p标签的写法

<HTML5> <p>必须用特殊字符标记来取代HTML标签特殊符号</p>

符号	代码
"	"
&	&
<	<
>	>
?/TD>	€
æ	æ
	¦
,	¸
©	©
¤	¤
°	°
÷	÷
?/TD>	€
f	ƒ
½	½
·	·
ó	ó
'	‘
'	’
"	“
"	”
%o	‰
X	χ
±	±
£	£
§	§
¹	¹

CSS 与 HTML 的四种结合方式

目录:

HTML 标签的 style 属性，能在 HTML 代码下写 CSS.....	48
Style 将 CSS 分离出来不放在标签里， html 和 CSS 结合的一种方式.....	48
将 html 和 CSS 文件分离， CSS 写一个文档， html 写一个文档.....	49
用 link 将 css 代码文件和 html 代码文件分离.....	50
如何避免两个 div 一个 css 颜色修改器，两个 div 颜色改成一样.....	50
Class 选择器.....	51
id 选择器.....	51
id 选择器， class 选择器和标签选择器优先级区别.....	52
我想修改标签里面嵌套的标签颜色怎么办？使用关联选择器.....	54
不同的标签设置同样的颜色，解决在 style 里面写过多同样标签颜色的 css.....	54
如何实现一个图标根据鼠标的移动有多种颜色表现，伪类选择器.....	55
动态伪类选择器 hover, active 多种用法，鼠标移动到 a 标签让字体变大.....	56
表单伪类选择器，在输入框加颜色.....	58
伪类选择器 target 属性项目实例.....	62
让 div 里面的字体或者元素居中.....	64
动态伪类选择器.....	66
结构性伪类选择器，根据编号指定某行标签.....	66
CSS 盒子模型.....	72
在 CSS 里设置 div 区域块大小，和边框颜色.....	73
CSS 设置 div 边框单个的颜色和样式.....	74
div 内边距设置，就是类似 word 文字居中.....	74
div 文字内边距指定方向设置.....	75
float CSS 给 div 窗口布局定位，就是可以让 div 窗口横着页面放.....	75
CSS 各种属性使用案例.....	84
后代选择器设置 html 里面指定标签的颜色.....	85
>子元素选择器，只选择第一层指定的标签，标签内部嵌套相同标签不会被影响.....	87
+相邻标签选择器.....	87
~兄弟选择器，选择同一个标签下的 div，.....	88
标签[...] 属性选择器。比如 div 标签，我让 div 标签有同名的选择，没同名的不选择..	89
属性选择器之值选择器.....	89
自定义字体.....	91
用别人做好的字体图标.....	92
在 CSS 中给图标增加 opacity，可以让图标透明.....	93
Background:rgb(..) CSS 设置标签背景颜色和透明.....	94
text-shadow CSS 设置字体阴影.....	95

text-align:center 字体居中.....	95
transition:1s.....	97
图片标签后面背景如何嵌入模糊图片.....	97
position:relative 把标签定死在页面.....	98
margin:10px 10px 10px 10px 设置标签离父容器框的上下左右距离.....	98
文字描边.....	100
盒子模型阴影，也就是标签块阴影.....	101
margin:auto 图片水平居中.....	103
标签图片倒影.....	105
鼠标拖动标签块或者 div 块大小的功能实现.....	106
让标签块或者 div 块在页面居中的一种方法.....	107
div 块方形倒圆角.....	109
如何将 div 方形制作成椭圆形，借鉴 div 方形倒圆角知识.....	111
用 border-image-source 建立边框图片.....	113

HTML 标签的 style 属性，能在 HTML 代码下写 CSS

```
<!DOCTYPE html>
<html>
<head>
    <title></title>
    <meta charset=UTF-8>
</head>
<body>
    <div>html与css的结合</div>
</body>
</html>
```

字体没有颜色

```
</head>
<body>
    <div style="background-color: red;">html与css的结合</div>
</body>
```

给字符添加背景颜色

Style: 就是 CSS 属性

background-color: 就是添加背景颜色

red 变量: 就是背景的颜色为红色

```
<body>
    <div style="background-color: red;color: green;">html与css的结合</div>
</body>
```

Color 是前景色属性
设置为 green 绿色

Style 将 CSS 分离出来不放在标签里，html 和 CSS 结合的一种方式

```
<html>
<head>
    <title></title>
    <meta charset=UTF-8>

    <style type="text/css">
        div{
            background-color: yellow;
            color: blue;
        }
    </style>
</head>
<body>
    <div>html与css的结合</div>
</body>
</html>
```

在 head 标签下加 style 标签，指定标签属于 CSS 代码段，固定模式

因为我是要给 div 标签里面的内容加颜色，所以这里写 div，然后用大括号将要写颜色的属性包括起来

将 html 和 CSS 文件分离，CSS 写一个文档，html 写一个文档

D: css-html.css
D: All Files (*.*) 在本地目录下建立一个 css 文件

```
css-html.html * css-html.css  
/*CSS文件*/  
div{ background-color:aqua; color: black;  
}  
其实和 style 一样，指定标签的颜色，只不过是在另外一个 CSS 文件给 div 标签上色
```

```
css-html.html * css-html.css *  
<!DOCTYPE html>  
<html>  
<head>  
    <title></title>  
    <meta charset=UTF-8>  
  
    <style type="text/css">  
        @import url(css-html.css)  
    </style>  
</head>  
<body>  
    <div>html与CSS的结合</div>  
</body>  
</html>  
Html 标签还是在 style 标签里面用@import url 将本地路径的 css 文件 引用进来
```

html与CSS的结合

最后结果也是一样的，只是 css 和 html 文件分离方式容易管理代码，所以是比较流行的 html 与 CSS 结合的一种方法。

但是这种方式有个问题，就是在有些浏览器下根本无法兼容，报错什么的。

所以我们用下面的 link 来代替@import 方法

用 link 将 css 代码文件和 html 代码文件分离

还是要单独建立个 css 文件

```
/*CSS文件*/
div{
    background-color:aqua;
    color: black;
}
```

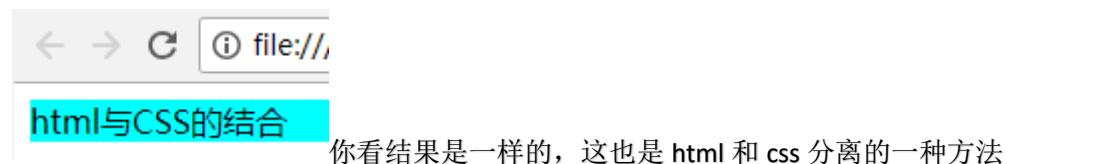
就用上面这个 css 文件

```
<html>
<head>
    <title></title>
    <meta charset=UTF-8>

    <link rel="stylesheet" type="text/css" href="css-html.css">
</head>
<body>
    <div>html与CSS的结合</div>
</body>
```

取消 style，用 link 代替

这里指定 css 文件路径，因为是当前路径所以没有/斜杠



如何避免两个 div 一个 css 颜色修改器，两个 div 颜色改成一样

```
<head>
    <title></title>
    <meta charset=UTF-8>

    <style type="text/css">
        div
        {
            background-color: aqua;
            color: red;
        }
    </style>

</head>
<body>
    <div>html与CSS的结合</div>
    <div>html与CSS的结合第二行</div>
</body>
```

我想两个 div 颜色不一样，但是现在一样了，因为一个 css 的 div 包含了所有 div 标签的颜色

Class 选择器

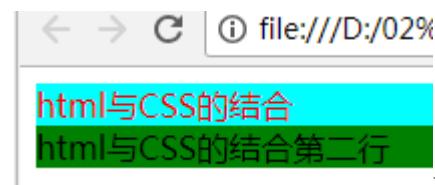
所以我们要用 class 选择器来解决多个 div 实现颜色不同的问题

```
<head>
    <title></title>
    <meta charset=UTF-8>

    <style type="text/css">
        div.first1{
            background-color: aqua;
            color: red;
        }
        div.first2{
            background-color: green;
            color: black;
        }
    </style>
</head>
<body>
    <div class="first1">html与CSS的结合</div>
    <div class="first2">html与CSS的结合第二行</div>
</body>
```

在 CSS 里用 div . (class 名字) 来决定给指定的 div 赋色

用 class 给每个 div 取独有的名字



这就实现了不同的 div 不同的颜色

id 选择器

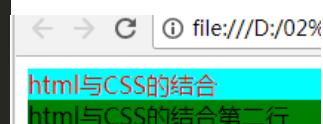
还有一个和 class 一样的选择器， id 选择器

```
<head>
    <title></title>
    <meta charset=UTF-8>

    <style type="text/css">
        div#first1{
            background-color: aqua;
            color: red;
        }
        div#first2{
            background-color: green;
            color: black;
        }
    </style>
</head>
<body>
    <div id="first1">html与CSS的结合</div>
    <div id="first2">html与CSS的结合第二行</div>
</body>
```

在 CSS 里用 div # (class 名字) 来决定给指定的 div 赋色

用 id 给每个 div 设置唯一的名字



id 选择器，class 选择器和标签选择器优先级区别

```
<head>
    <title></title>
    <meta charset=UTF-8>

    <style type="text/css">

        div {
            background-color: green;
            color: blue;
        }
    </style>
</head>
<body>
    <div class="class1">xxxxxx</div>
    <div>pppppppppppppppppppppppp</div>
</body>
```

xxxxxx
pppppppppppppppppppppp

正常情况下两个 div 标签都是同样的颜色

```
<head>
    <title></title>
    <meta charset=UTF-8>

    <style type="text/css">

        div {
            background-color: green;
            color: blue;
        }
        div.class1 {
            background-color: aqua;
            color: blue;
        }
    </style>
</head>
<body>
    <div class="class1">xxxxxx</div>
    <div>pppppppppppppppppppppp</div>
</body>
```

程序是从上而下执行的，所以第一个 CSS 是设置 div 标签的颜色

第 2 个 CSS 是设置 class 属性的 div 标签

xxxxxx
pppppppppppppppppppppp

如果我把 div.class1 放在前面呢？

```

<head>
    <title></title>
    <meta charset=UTF-8>

    <style type="text/css">
        div.class1 {
            background-color: aqua;
            color: blue;
        }
        div {
            background-color: green;
            color: blue;
        }
    </style>
</head>
<body>
    <div class="class1">xxxxxx</div>
    <div>pppppppppppppppppppppppp</div>
</body>

```

因为程序从上而下执行，所以第一个 div.class1 修改了具备 class1 属性的 div 标签

但是第二个 CSS 的 div 具备全局属性，应该要覆盖 div.class1 设置的颜色，全部背景色都是绿色

但是实际情况是并没有全部变成背景绿色

这就证明了 class 选择器比普通 div 标签的优先级高

下面我们加入 id 选择器

```

<head>
    <title></title>
    <meta charset=UTF-8>

    <style type="text/css">
        div#id1 {
            background-color: orange;
            color: blue;
        }
        div.class1 {
            background-color: aqua;
            color: blue;
        }
        div {
            background-color: green;
            color: blue;
        }
    </style>
</head>
<body>
    <div class="class1" id="id1">xxxxxx</div>
    <div>pppppppppppppppppppppppp</div>
</body>

```

id 选择器放在最上面，所以 id 选择器先加载，背景颜色应该是橙色，但是下面又遇到了 class 选择器，那么背景颜色应该是灰色

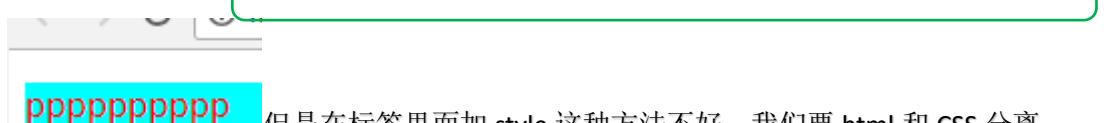
但是实际的结果还是橙色，这说明了什么问题呢？其实就是 id 选择器优先级最高

所以 id 选择器优先级 > class 选择器 > 标签选择器

我想修改标签里面嵌套的标签颜色怎么办？使用关联选择器

```
<body>
    <div><p style="background-color: aqua; color: red;">pppppppppp</p></div>
</body>
```

我想修改 div 标签里面 p 标签包裹的文字颜色，我们可以使用 style



pppppppppp

但是在标签里面加 style 这种方法不好，我们要 html 和 css 分离

```
<head>
    <title></title>
    <meta charset=UTF-8>

    <style type="text/css">
        div p
        {
            background-color: aqua;
            color: red;
        }
    </style>
</head>
<body>

    <div><p>pppppppppp</p></div>

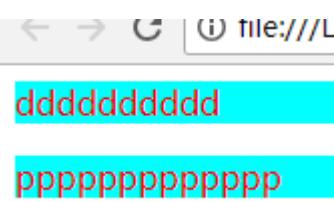
</body>
```

这就是关联标
签，先指定
div，然后空格
加 p，就表明了
div 标签内 p 标
签的颜色

不同的标签设置同样的颜色，解决在 style 里面写过多同样标签颜色的 css

```
7   <style type="text/css">
8       div,p
9       {
10           background-color: aqua;
11           color: red;
12       }
13
14   </style>
15 </head>
16 <body>
17
18   <div>ddddddddd</div>
19   <p>pppppppppppp</p>
```

用逗号将两个标签
链接起来，这样就
可以设置同样的颜
色了(逗号一定是英
文模式下书写)



```

<head>
    <title></title>
    <meta charset=UTF-8>

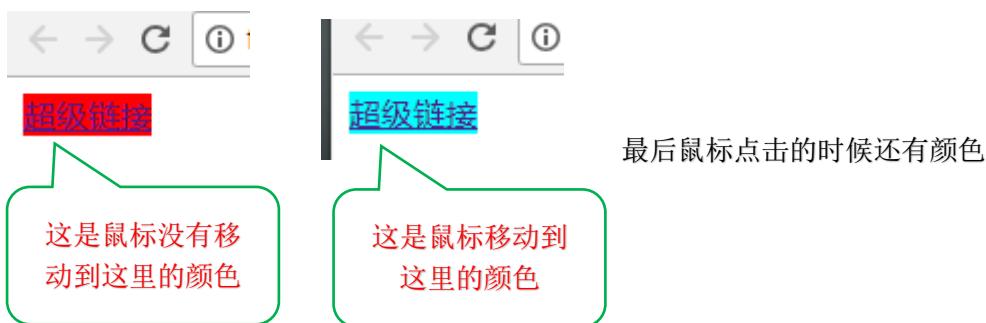
    <style type="text/css">
        div,p,h1
        {
            background-color: aqua;
            color: red;
        }
    </style>
</head>
<body>

    <div>ddddddddd</div>
    <p>ppppppppppppp</p>
    <h1>hhhhhhhhhhhhhh</h1>

</body>

```

如何实现一个图标根据鼠标的移动有多种颜色表现，伪类选择器



```

5
6     <style type="text/css">
7
8         a:link{
9             background-color: red;
10        }
11        a:hover{
12            background-color: aqua;
13        }
14        a:active{
15            background-color: green;
16        }
17
18     </style>
19 </head>
20 <body>
21
22     <a href="https://www.baidu.com/" target="_blank">超级链接</a>
23
24 </body>
25

```

所以你只需要记得 link, hover, active 这三个属性设置颜色就是了，我们下面用 div 试试

```

7   <style type="text/css">
8
9     div:link{
10       background-color: red;
11     }
12    div:hover{
13      background-color: aqua;
14    }
15    div:active{
16      background-color: green;
17    }
18
19  </style>
20 </head>
21 <body>
22   <div>超级链接</div>
23 </body>

```



动态伪类选择器 `hover`, `active` 多种用法，鼠标移动到 `a` 标签让字体变大

```

7   <style type="text/css">
8     a:hover{
9       font-size: 24px;
10    }
11  </style>
12 </head>
13 <body>
14   <a href="#div1">选择div1图框</a>
15   <a href="#div2">选择div2图框</a>
16   <a href="#div3">选择div3图框</a>
17   <a href="#div3">选择div4图框</a>
18   <a href="#div3">选择div5图框</a>
19   <a href="#div3">选择div6图框</a>
20
21 </body>

```

选择div1图框 选择div2图框 选择div3图框 选择div4图框 选择div5图框 选择div6图框

`hover` 就是个鼠标移动到标签的属性，一旦鼠标移动到了标签，就执行 `hover` 里面的 `css` 代码，至于里面写什么代码可以自己随意定义

我感觉标签横着排列我觉得不舒服

```
6
7     <style type="text/css">
8         a{
9             display: block;
10            }
11            a:hover{
12                font-size: 24px;
13            }
14        </style>
15    </head>
16    <body>
17        <a href="#div1">选择div1图框</a>
18        <a href="#div2">选择div2图框</a>
19        <a href="#div3">选择div3图框</a>
20        <a href="#div3">选择div4图框</a>
21        <a href="#div3">选择div5图框</a>
22        <a href="#div3">选择div6图框</a>
23
24    </body>
```

设置 display 就可以让标签竖行排列

选择div1图框
选择div2图框
选择div3图框
选择div4图框
选择div5图框
选择div6图框

除了 a 标签，div 标签也支持 hover 属性

```
6
7
8     <style type="text/css">
9         div{
10            display: block;
11            }
12            div:hover{
13                font-size: 24px;
14            }
15        </style>
16    </head>
17    <body>
18        <div>选择div1图框</div>
19        <div>选择div2图框</div>
20        <div>选择div3图框</div>
21        <div>选择div4图框</div>
22        <div>选择div5图框</div>
23        <div>选择div6图框</div>
```

选择div1图框
选择div2图框
选择div3图框
选择div4图框
选择div5图框
选择div6图框

```
<style type="text/css">
    div{
        display: block;
    }
    div:hover{
        font-size: 24px;
    }
    div:active{
        font-size: 50px;
    }
</style>
</head>
<body>
    <div>选择div1图框</div>
    <div>选择div2图框</div>
    <div>选择div3图框</div>
    <div>选择div4图框</div>
    <div>选择div5图框</div>
    <div>选择div6图框</div>
</body>
```

选择div1图框
选择div2图框
选择div3图框
选择div4图框
选择div5图框
选择div6图框

给 div 加入 active 让点击的时候字体变更大

表单伪类选择器，在输入框加颜色

```
3 <head>
4     <title></title>
5     <meta charset=UTF-8>
6
7     <style type="text/css">
8         input:enabled{
9             background: pink;
10            }
11
12     </style>
13 </head>
14 <body>
15     <input type="text" >
16 </body>
```

css-html.html#div3

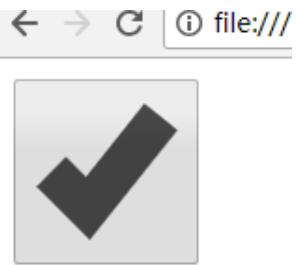
file:///D:/02%20我的

给可写的表单加颜色

```
6      <style type="text/css">
7          input:enabled{
8              background: pink;
9          }
10     </style>
11 </head>
12 <body>
13     <input type="text" disabled="disabled">
14 </body>
```

加入了 disabled 后
表单就不可写了

```
5         <meta charset=UTF-8>
6
7         <style type="text/css">
8             加入 checked
9             伪类，就是指
10            定 input 标签
11            的 checkbox
12            点击后的样式
13             input:checked{
14                 width: 100px;
15                 height: 100px;
16             }
17         </style>
18     </head>
19     <body>
20         <input type="checkbox" >
21     </body>
```



自定义单选按钮

```
6
7         <style type="text/css">
8
9             </style>
10    </head>
11    <body>
12        <input type="radio" name="xzz" >
13        <input type="radio" name="xzz" >
14        <input type="radio" name="xzz" >
15    </body>
```

默认 name 名字不一样是
多选按钮。设置 name 名
字一样才是单选按钮

下面我来修改选择按钮的样式



```
<style type="text/css">
    label{
        width: 500px;
        height: 500px;
        border:1px solid pink;
    }
</style>
</head>
<body>
    <label>
        <input type="radio" name="xzz" >
        <span></span>
    </label>
    <label>
        <input type="radio" name="xzz" >
        <span></span>
    </label>
    <label>
        <input type="radio" name="xzz" >
        <span></span>
    </label>
</body>
```

2.为什么无法设置选择按钮边框的大小呢?

1.用 label 标签包裹选择按钮 radio, 这样点击 label 标签下的字或者边框都可以选择上按钮

不能在 CSS 指定 width, height 的标签是内敛元素

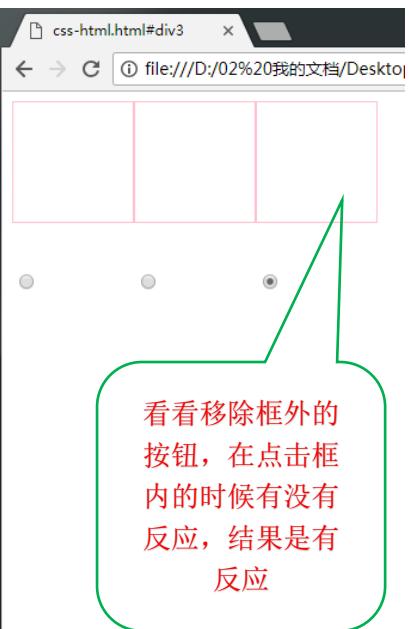
```
<style type="text/css">
    label{
        float: left;
        width: 100px;
        height: 100px;
        border:1px solid pink;
    }
</style>
</head>
<body>
    <label>
        <input type="radio" name="xzz" >
        <span></span>
    </label>
    <label>
        <input type="radio" name="xzz" >
        <span></span>
    </label>
    <label>
        <input type="radio" name="xzz" >
        <span></span>
    </label>
</body>
```

我让 label 标签浮动起来就可以修改 width 和 height 了

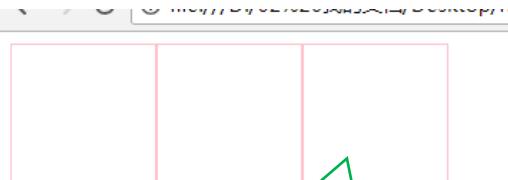
但是按钮在框里面好丑哦

```
7     <style type="text/css">
8         label{
9             float: left;
10            width: 100px;
11            height: 100px;
12            border:1px solid pink;
13        }
14        input{
15            position: absolute;
16            top: 150px;
17        }
18    </style>
19 </head>
20 <body>
21     <label>
22         <input type="radio" name="xzz" >
23         <span></span>
24     </label>
25     <label>
26         <input type="radio" name="xzz" >
27         <span></span>
28     </label>
29     <label>
30         <input type="radio" name="xzz" >
31         <span></span>
32     </label>
33     <label>
34         <input type="radio" name="xzz" >
35         <span></span>
36     </label>
37 </body>
```

我先把单选按钮移除框外



```
input{
    position: absolute;
    left: -50px;
    top: -50px;
}
```



伪类选择器 target 属性项目实例

```
<head>
    <title></title>
    <meta charset=UTF-8>

    <style type="text/css">

        </style>
</head>
<body>
    <a href="#div1">选择div1图框</a>
    <a href="#div2">选择div2图框</a>
    <a href="#div3">选择div3图框</a>
</body>
```

选择div1图框 选择div2图框 选择div3图框

我们建立了 3 个 div 超级
链接标签，但是感觉怎么
没有填满浏览器的边框

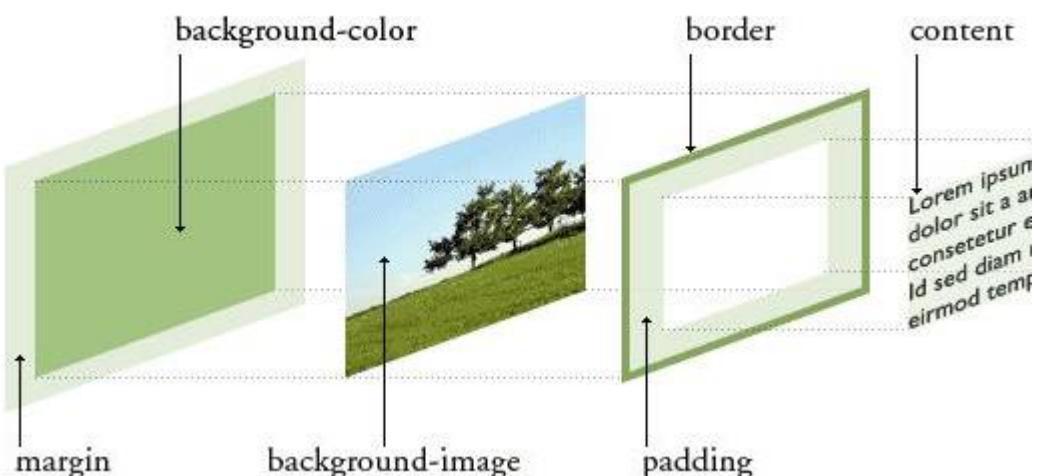
```
3 <head>
4     <title></title>
5     <meta charset=UTF-8>
6
7     <style type="text/css">
8         *{
9             margin: 0;
10            padding: 0;
11        }
12
13     </style>
14 </head>
15 <body>
16     <a href="#div1">选择div1图框</a>
17     <a href="#div2">选择div2图框</a>
18     <a href="#div3">选择div3图框</a>
19 </body>
```

选择div1图框 选择div2图框 选择div3图框

我们发现标签填满浏览器
了，没有空行

那么这个 margin 和 padding 到底是什么？

THE BOX MODEL



```
margin: 层的边框以外留的空白  
background-color: 背景颜色  
background-image: 背景图片  
padding: 层的边框到层的内容之间的空白  
border: 边框  
content: 内容
```

表示标签边框以外的区域
要不要空出一定的距离

我们这里选择不准留距离

选择div1图框 选择div2图框 选择div3图框

这个标签下划线看着不舒服，想法去掉

```
7   <style type="text/css">  
8     *{  
9       margin: 0;  
10      padding: 0;  
11    }  
12    a{  
13      text-decoration: none;  
14    }  
15  </style>  把所有 a 标签下划线去掉  
16 </head>  
17 <body>  
18   <a href="#div1">选择div1图框</a>  
19   <a href="#div2">选择div2图框</a>  
20   <a href="#div3">选择div3图框</a>  
21 </body>
```

选择div1图框 选择div2图框 选择div3图框

你看下划线去掉了

```
7   <style type="text/css">
8     *{
9       margin: 0;
10      padding: 0;
11    }
12    a{
13      text-decoration: none;
14    }
15    div{
16      width: 200px;
17      height: 200px;
18      background-color: pink;
19    }
20  </style>          给 div 上色
21 </head>
22 <body>
23   <a href="#div1">选择div1图框</a>
24   <a href="#div2">选择div2图框</a>
25   <a href="#div3">选择div3图框</a>
26
27   <div id="div1">div1被选择上</div>
28   <div id="div2">div2被选择上</div>
29   <div id="div3">div3被选择上</div>
30 </body>
```

1. 建立 3 个 div, a 标签点击的时候会自动执行 id 对应的 div

现在有个问题 div 的字体无法居住

让 div 里面的字体或者元素居中

```
div{
  width: 200px;
  height: 200px;
  background-color: pink;
  text-align: center;
```

center 是水平居中

```
5 <meta charset=UTF-8>                                         选择div1图框 选择div2图框 选择div3图框
6
7 <style type="text/css">
8   *{
9     margin: 0;
10    padding: 0;
11  }
12  a{
13    text-decoration: none;
14  }
15  div{
16    width: 200px;
17    height: 200px;
18    background-color: pink;
19    text-align: center;
20    display: none;
21  }
22 </style>
23 </head>
24 <body>
25   <a href="#div1">选择div1图框</a>          我们先把网页的 div 标签关闭了 none, 等
26   <a href="#div2">选择div2图框</a>
27   <a href="#div3">选择div3图框</a>
```

```
<title></title>
5 <meta charset=UTF-8>
6
7 <style type="text/css">
8   *{
9     margin: 0;
10    padding: 0;
11  }
12  a{
13    text-decoration: none;
14  }
15  div{
16    width: 200px;
17    height: 200px;
18    background-color: pink;
19    text-align: center;
20    display: none;
21  }
22  :target{           target 使用必须要有 a 标
23    display: block;  签, 点击任何一个 a 标
24  }
25 </style>
26 </head>
27 <body>
28   <a href="#div1">选择div1图框</a>          标签, 然后根据点击 a 标签
29   <a href="#div2">选择div2图框</a>          的 uri, 也就是 href 得到
30   <a href="#div3">选择div3图框</a>          的 ID 值, 去找对应 id 的
31
32   <div id="div1">div1被选择上</div>        div 来执行
33   <div id="div2">div2被选择上</div>
34   <div id="div3">div3被选择上</div>
```



动态伪类选择器

结构性伪类选择器，根据编号指定某行标签

```
<style type="text/css">
    #warp li:nth-child(1){
        color: red;
    }
</style>
</head>
<body>
    <ul id="warp">
        <li>1</li>
        <li>2</li>
        <li>3</li>
        <li>4</li>
        <li>5</li>
    </ul>
</body>
```

用后代选择器选择 warp 下的 li 标签，然后用 nth-child 关键字指定标签行号，对其赋值颜色

```
7     <style type="text/css">
8         #warp li:nth-child(2){
9             color: red;
10        }
11    </style>
12 </head>
13 <body>
14     <ul id="warp">
15         <li>1</li>
16         <li>2</li>
17         <li>3</li>
18         <li>4</li>
19         <li>5</li>
20     </ul>
21 </body>
```

也可以选择 li 标签第 2 行

```
7     <style type="text/css">
8         #warp li:nth-child(1){
9             color: red;
10        }
11    </style>
12 </head>
13 <body>
14     <ul id="warp">
15         <p></p>
16         <li>1</li>
17         <li>2</li>
18         <li>3</li>
19         <li>4</li>
20         <li>5</li>
21     </ul>
22 </body>
```

但是在后代选择器指定的 li 前面加其他标签，就不行了

所以 nth-child 要求 warp 下面的子元素第一个必须是 li，因为你的后代选择器选择的 li

```
7     <style type="text/css">
8         #warp dev:nth-child(1){
9             color: red;
10            }
11    </style>
12 </head>
13 <body>
14     <ul id="warp">
15         <dev>1</dev>
16         <dev>2</dev>
17         <dev>3</dev>
18         <dev>4</dev>
19         <dev>5</dev>
20     </ul>
21 </body>
```

如果 nth-child 属性的后代选择器是 dev，那么 id=warp 下面的第一个标签必须是 dev

```
7     <style type="text/css">
8         #warp li:nth-child(1){
9             color: red;
10            }
11    </style>
12 </head>
13 <body>
14     <ul id="warp">
15         <p></p>
16         <li>1</li>
17         <li>2</li>
18         <li>3</li>
19         <li>4</li>
20         <li>5</li>
21     </ul>
22 </body>
```

所以这样是不行的，因为 nth-child 要求 warp 下第 1 个子元素是 li，你却是 p 标签所以不行

```
7     <style type="text/css">
8         #warp p:nth-of-type(1){
9             color: red;
10        }
11    </style>
12 </head>
13 <body>
14     <ul id="warp">
15         <p>ppp</p>
16         <li>1</li>
17         <li>2</li>
18         <li>3</li>
19         <li>4</li>
20         <li>5</li>
21     </ul>
22 </body>
```

nth-of-type 属性是指定的标签不管在前在后都按照序号设置

```
7     <style type="text/css">
8         #warp p:nth-of-type(1){
9             color: red;
10        }
11    </style>
12 </head>
13 <body>
14     <ul id="warp">
15
16         <li>1</li>
17         <li>2</li>
18         <li>3</li>
19         <li>4</li>
20         <li>5</li>
21         <p>ppp</p>
22         <p>p222</p>
23     </ul>
24 </body>
```

你看放在后面

```
7     <style type="text/css">
8         #warp p:nth-of-type(2){
9             color: red;
10        }
11    </style>
12 </head>
13 <body>
14     <ul id="warp">
15
16         <li>1</li>
17         <li>2</li>
18         <li>3</li>
19         <li>4</li>
20         <li>5</li>
21         <p>ppp</p>
22         <p>p222</p>
23     </ul>
24 </body>
```

指定后面第 2 个

所以 nth-of-type 灵活性比较大

nth-child 和 nth-of-type 的坑

```
7      <style type="text/css">
8          #wrap .inner:nth-child(1)
9          {
10             border:1px solid;
11         }
12     </style>
13 </head>
14 <body>
15     <div id="wrap">
16         <div class="inner">div</div>
17         <p class="inner">p</p>
18         <span class="inner">span</span>
19         <h1 class="inner">h1</h1>
20         <h2 class="inner">h1</h2>
21     </div>
22 </body>
```

nth-child 不管什么标签类型只要 class 元素名是一样，就只选择第 1 行元素名的标签

div
p
span
h1
h1

```
7      <style type="text/css">
8          #wrap .inner:nth-of-type(1)
9          {
10             border:1px solid;
11         }
12     </style>
13 </head>
14 <body>
15     <div id="wrap">
16         <div class="inner">div</div>
17         <p class="inner">p</p>
18         <span class="inner">span</span>
19         <h1 class="inner">h1</h1>
20         <h2 class="inner">h1</h2>
21     </div>
22 </body>
```

nth-of-type 是以元素名为选择的，所以下面标签类型不一样，只要元素名一样，每个标签都是第 1 行

div
p
span
h1
h1

```
6
7     <style type="text/css">
8         #wrap .inner:nth-of-type(1)
9     {
10         border:1px solid;
11     }
12 </style>
13 </head>
14 <body>
15     <div id="wrap">
16         <div class="inner">div</div>
17         <p class="inner">p</p>
18         <p class="inner">ppp</p>
19         <p class="inner">ppppp</p>
20         <span class="inner">span</span>
21         <span class="inner">spanspan</span>
22         <h1 class="inner">h1</h1>
23         <h2 class="inner">h1</h2>
24     </div>
25 </body>
```

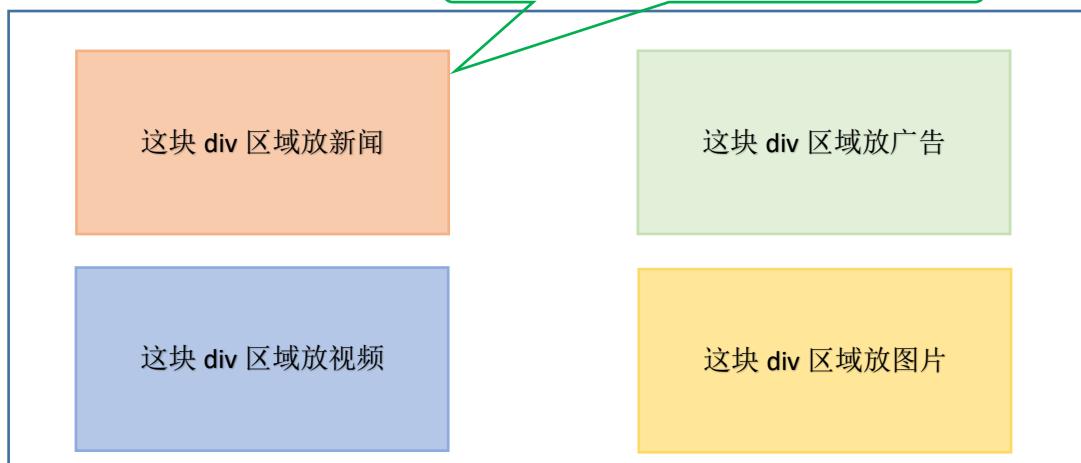
The diagram illustrates the CSS selector `#wrap .inner:nth-of-type(1)`. It shows a tree structure of elements: a `div` containing a `p`, two `ppp` spans, two `spanspan` spans, and two `h1` elements. A green bracket highlights the first `h1` element, which is enclosed in a red box. A callout bubble points to this `h1` with the text "选择的是同一个类型标签的第 1 行". Another callout bubble points to the first `h1` with the text "我同一个类型的标签多加几个就看出来了".

```
6
7     <style type="text/css">
8         #wrap .inner:nth-of-type(2)
9     {
10         border:1px solid;
11     }
12 </style>
13 </head>
14 <body>
15     <div id="wrap">
16         <div class="inner">div</div>
17         <p class="inner">p</p>
18         <p class="inner">ppp</p>
19         <p class="inner">ppppp</p>
20         <span class="inner">span</span>
21         <span class="inner">spanspan</span>
22         <h1 class="inner">h1</h1>
23         <h2 class="inner">h1</h2>
24     </div>
25 </body>
```

The diagram illustrates the CSS selector `#wrap .inner:nth-of-type(2)`. It shows the same tree structure as the previous diagram. A green bracket highlights the second `h1` element, which is also enclosed in a red box. A callout bubble points to this `h1` with the text "选择的是同一个类型标签的第 2 行".

CSS 盒子模型

盒子模型就是一个 div 当做一个盒子



就类似与下面 B 站这样



```
</head>
<body>
    <div id="id1">第一个div第一个div第一个div</div>
    <div id="id2">第二个div第二个div第二个div</div>
    <div id="id3">第三个div第三个div第三个div</div>
</body>
```

← → ⏪ ⓘ file:///D:/02%20我的文

第一个div第一个div第一个div
第二个div第二个div第二个div
第三个div第三个div第三个div

这是 div 默认情况

在 CSS 里设置 div 区域块大小，和边框颜色

```
<head>
    <title></title>
    <meta charset=UTF-8>

    <style type="text/css">
        div {
            width: 200px;
            height: 100px;
        }
    </style>
</head>
<body>
    <div id="id1">第一个div第一个div第一个div</div>
    <div id="id2">第二个div第二个div第二个div</div>
    <div id="id3">第三个div第三个div第三个div</div>
</body>
```

← → C | ⓘ file:///D:/02%20我的文

第一个div第一个div第一个
div

第二个div第二个div第二个
div

第三个div第三个div第三个
div

这是 div 区域的宽度

这是 div 区域的高度，px 单位是像素

这就是 div 区域宽度，因
为没有 div 边框线条看不
出来，我们来给 div 加边
框

```
5     <meta charset=UTF-8>
6
7     <style type="text/css">
8         div {
9             width: 200px;
10            height: 100px;
11            border: 2px solid blue;
12        }
13    </style>
14 </head>
15 <body>
16    <div id="id1">第一个div第一个div第一个div</div>
17    <div id="id2">第二个div第二个div第二个div</div>
18    <div id="id3">第三个div第三个div第三个div</div>
19 </body>
```

边框颜色蓝色

边框宽度 2 个像素

边框用实心线

第一个div第一个div第一个
div

第二个div第二个div第二个
div

第三个div第三个div第三个
div

CSS 设置 div 边框单个的颜色和样式

前面的 border 是设置 div 边框四条边的颜色和样式

div 边框单边设置要用：

border-top	边框上
border-bottom	边框下
border-left	边框左
border-right	边框右

The screenshot shows a browser window with three stacked div elements. The top div has a solid blue border. The middle div has a dashed green border on its right side. The bottom div has solid yellow borders on all four sides. Each div contains the text '第一个div' followed by its respective id ('id1', 'id2', or 'id3'). A callout bubble points to the border-right rule in the CSS with the text '设置单边' (Set single side). Another callout bubble points to the border-right rule in the #id3 block with the text '设置四边' (Set all four sides).

```
<head>
    <title></title>
    <meta charset=UTF-8>

    <style type="text/css">
        div {
            width: 200px;
            height: 100px;
            border: 2px solid blue;
        }
        #id2 {
            border-right: 10px dashed green;
        }
        #id3{
            border-top: 5px solid yellow;
            border-bottom: 5px solid yellow;
            border-left: 5px solid yellow;
            border-right: 5px solid yellow;
        }
    </style>
</head>
<body>
    <div id="id1">第一个div第一个div第一个div</div>
    <div id="id2">第二个div第二个div第二个div</div>
    <div id="id3">第三个div第三个div第三个div</div>
</body>
```

div 内边距设置，就是类似 word 文字居中

The screenshot shows a browser window with three stacked div elements. Each div has a solid blue border and contains the text '第一个div' followed by its respective id ('id1', 'id2', or 'id3'). A callout bubble points to the padding rule in the #id2 block with the text '设置文本内容距离 div 四条边框的最小距离' (Set the minimum distance between text content and the four border edges). Another callout bubble points to the same padding rule with the text '这里设置的 20' (The value 20 is set here).

```
<head>
    <title></title>
    <meta charset=UTF-8>

    <style type="text/css">
        div {
            width: 200px;
            height: 100px;
            border: 2px solid blue;
        }
        #id2{
            padding: 20px;
        }
    </style>
</head>
<body>
    <div id="id1">第一个div第一个div第一个div</div>
    <div id="id2">第二个div第二个div第二个div</div>
    <div id="id3">第三个div第三个div第三个div</div>
</body>
```

就是让文本距离边框 20 个像素

div 文字内边距指定方向设置

```
<title></title>
<meta charset=UTF-8>

<style type="text/css">
    div {
        width: 200px;
        height: 100px;
        border: 2px solid blue;
    }

    #id3{
        padding-left: 50px;
    }
</style>
</head>
<body>
    <div id="id1">第一个div第一个div第一个div</div>
    <div id="id2">第二个div第二个div第二个div</div>
    <div id="id3">第三个div第三个div第三个div</div>
</body>
```

文本离左边距离 50 像素

第一个div第一个div第一个div

第二个div第二个div第二个div

第三个div第三个div第三个div

你看文本只与左
边保持 50px 距离

float CSS 给 div 窗口布局定位，就是可以让 div 窗口横着页面放

这是 div 窗口



做情趣内衣赚5千万

这个 div 窗口是怎么横
着放的，不是 div 窗口
都是按照代码顺序向下
放吗？

- 重庆捣毁特大跨境诈骗团伙 女孩误
- ... 次... 次... 次...

第一种方法是使用 CSS 的漂浮属性

float 漂浮属性

参数: left 让 div 保持左边

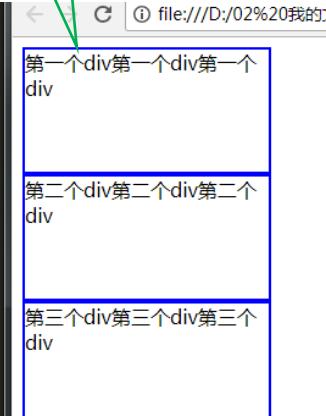
right 让 div 保持右边

本来是这样

```
<head>
    <title></title>
    <meta charset=UTF-8>

    <style type="text/css">
        div{
            width: 200px;
            height: 100px;
            border: 2px solid blue;
        }

    </style>
</head>
<body>
    <div id="id1">第一个div第一个div第一个div</div>
    <div id="id2">第二个div第二个div第二个div</div>
    <div id="id3">第三个div第三个div第三个div</div>
</body>
```



```

7     <style type="text/css">
8         div{
9             width: 200px;
10            height: 100px;
11            border:2px solid blue;
12        }
13        #id1{
14            float: left;
15        }
16
17     </style>
18 </head>
19 <body>
20     <div id="id1">第一个div第一个div第一个div</div>
21     <div id="id2">第二个div第二个div第二个div</div>
22     <div id="id3">第三个div第三个div第三个div</div>
23 </body>

```

增加 float 悬浮属性，然后指定悬浮位置在页面 left 左边

这里不知道为什么出现 bug

```

1     <style type="text/css">
2         div{
3             width: 200px;
4             height: 100px;
5             border:2px solid blue;
6         }
7         #id1{
8             float: left;
9         }
10        #id2{
11            float: left;
12        }
13        #id3{
14            float: left;
15        }
16
17     </style>
18 </head>
19 <body>
20     <div id="id1">第一个div第一个div第一个div</div>
21     <div id="id2">第二个div第二个div第二个div</div>
22     <div id="id3">第三个div第三个div第三个div</div>
23 </body>

```

为了解决 bug 我就直接将三个 div 都设置在左边

所以 float 这种属性很少用，因为在有些浏览器下面不好使，有 bug

但是为了说明整个问题我们还是来详细介绍一下

```

<body>
    <div class="box">
        <div class="float1">
        </div>
        <div class="float2">
        </div>
        <div class="float3">
        </div>
    </div>
</body>

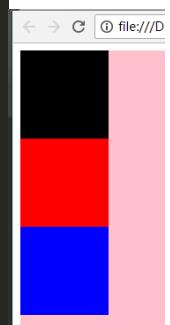
```

建立三个小的 div，这三个小的 div 块放在 box 大的 div 块上

```
<style type="text/css">
    .box{
        width: 600px;
        height: 400px;
        background: pink;
    }
    .float1
    {
        width: 100px;
        height: 100px;
        background: black;
    }
    .float2
    {
        width: 100px;
        height: 100px;
        background: red;
    }
    .float3
    {
        width: 100px;
        height: 100px;
        background: blue;
    }
</style>
```

给大的
div 块加
长宽颜色

给大的 div
块上面的
三个小 div
块加长宽
颜色



```

.float1
{
    width: 100px;
    height: 100px;
    background: black;
    float: left;

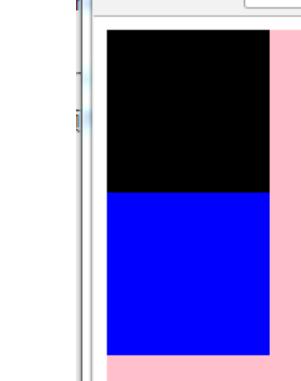
}

.float2
{
    width: 100px;
    height: 100px;
    background: red;

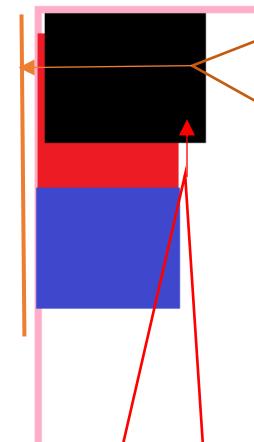
}

.float3
{
    width: 100px;
    height: 100px;
    background: blue;
}

```



为什么变成这样子了？红色呢？



1. 把黑色提取起来然后按照左边线对齐

2. 红色根据浏览器特性自动向上移动，所以被漂浮在上面的黑色遮住了

```

}
.float1
{
    width: 100px;
    height: 100px;
    background: black;
    float: left;

}

.float2
{
    width: 100px;
    height: 100px;
    background: red;
    float: left;

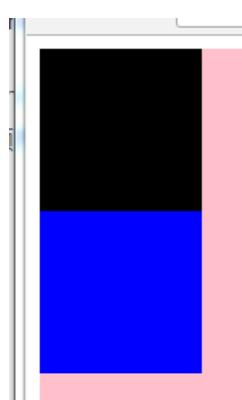
}

.float3
{
    width: 100px;
    height: 100px;
    background: blue;
}

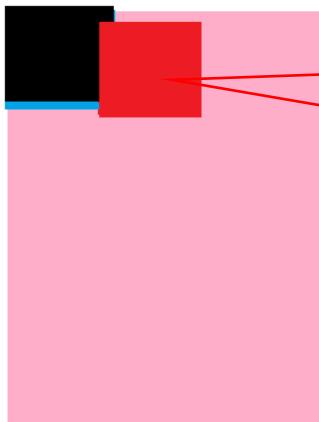
```



3. 我又把红色悬浮起来，然后向左对齐，但是为什么红色没有把黑色覆盖呢？



所以就成这样了



4.这是因为红色本来浮起来后和黑色就在同一层，然后发现黑色已经把左边第一格占用了，所以红色想占用左边第一格就不行了，但是因为属性是 left，所以只有挨着黑色放

```
.float1
{
    width: 100px;
    height: 100px;
    background: black;
    float: left;
}

.float2
{
    width: 100px;
    height: 100px;
    background: red;
    float: left;
}

.float3
{
    width: 100px;
    height: 100px;
    background: blue;
    float: left;
}
```

蓝色原理
是一样的

```
.float1
{
    width: 100px;
    height: 100px;
    background: black;
    float: left;
}

.float2
{
    width: 100px;
    height: 100px;
    background: red;
    float: left;
}

.float3
{
    width: 100px;
    height: 100px;
    background: blue;
    float: right;
}
```

蓝色向右对齐就是
这样，原理差不多

```

    .float1
    {
        width: 100px;
        height: 100px;
        background: black;
        float: left;
    }
    .float2
    {
        width: 100px;
        height: 100px;
        background: red;
        float: right;
    }
    .float3
    {
        width: 100px;
        height: 100px;
        background: blue;
        float: right;
    }

```

因为红色右对齐先执行，所以红色先占坑，蓝色就只有跟着挨着了

下面我们来试试并排两行 div

```

    .box{
        width: 600px;
        height: 400px;
        background: pink;
    }
    .float1
    {
        width: 100px;
        height: 100px;
        background: black;
        float: left;
    }
    .float2
    {
        width: 100px;
        height: 100px;
        background: red;
        float: left;
    }
    .float3
    {
        width: 100px;
        height: 100px;
        background: blue;
        float: left;
    }
    .float4
    {
        width: 100px;
        height: 100px;
        background: green;
        float: left;
        clear: left;
    }
    .float5
    {
        width: 100px;
        height: 100px;
        background: gray;
        float: left;
    }
    .float6
    {
        width: 100px;
        height: 100px;
        background: white;
        float: left;
    }

```

这里第二行一定要用 clear:left 让该 div 锁死在本行左边或者右边

这样第一行 div 全部浮空了，但是第二行 div 不会根据浏览器特性自动填补到第一行去被遮挡。

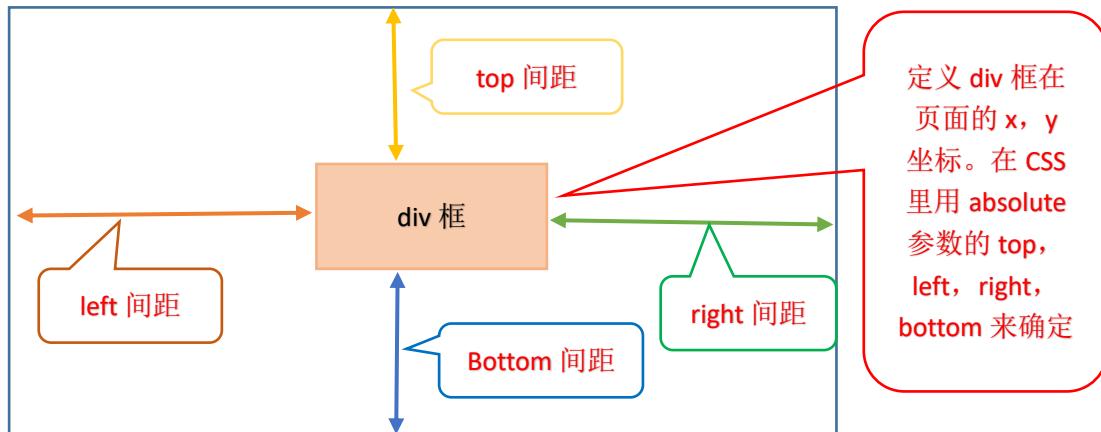
```

</head>
<body>
    <div class="box">
        <div class="flo1">
            <div class="float1">
            </div>
            <div class="float2">
            </div>
            <div class="float3">
            </div>
        </div>          第 1 行 div
        <div class="flo2">
            <div class="float4">
            </div>
            <div class="float5">
            </div>
            <div class="float6">
            </div>
        </div>          第 2 行 div
    </div>
</body>

```

第二种方法是用 position 属性方法, 将 div 框指定定位到页面的某个位置

参数: absolute



我们先把 div 框建立起来

```

7   <style type="text/css">
8     div{
9       width: 200px;
10      height: 100px;
11      border:2px solid blue;
12    }
13    #id1{
14      background-color: red;
15    }
16    #id2{
17      background-color: aqua;
18    }
19    #id3{
20      background-color: green;
21    }
22
23   </style>
24 </head>
25 <body>
26   <div id="id1">第一个div第一个div第一个div</div>
27   <div id="id2">第二个div第二个div第二个div</div>
28   <div id="id3">第三个div第三个div第三个div</div>
29 </body>

```

```

1   <style type="text/css">
2     div{
3       width: 200px;
4       height: 100px;
5       border:2px solid blue;
6     }
7     #id1{
8       background-color: red;
9       position: absolute;
10    }
11    #id2{
12      background-color: aqua;
13    }
14    #id3{
15      background-color: green;
16    }
17
18   </style>
19 </head>
20 <body>
21   <div id="id1">第一个div第一个div第一个div</div>
22   <div id="id2">第二个div第二个div第二个div</div>
23   <div id="id3">第三个div第三个div第三个div</div>
24 </body>

```

怎么第二个 div 不见了?

```

1   <style type="text/css">
2     div{
3       width: 200px;
4       height: 100px;
5       border:2px solid blue;
6     }
7     #id1{
8       background-color: red;
9       position: absolute;
10    }
11    #id2{
12      background-color: aqua;
13      width: 400px;
14      height: 100px;
15    }
16
17   </style>
18 </head>
19 <body>
20   <div id="id1">第一个div第一个div第一个div</div>
21   <div id="id2">第二个div第二个div第二个div</div>
22   <div id="id3">第三个div第三个div第三个div</div>
23 </body>

```

其实是第一个 div 悬浮起来了，然后根据 div 的属性，第一个 div 离开网页固定位置，第二个 div 就会自动移动上来

```

7   <style type="text/css">
8     div{
9       width: 200px;
10      height: 100px;
11      border:2px solid blue;
12    }
13    #id1{
14      background-color: red;
15      position: absolute;
16      top: 50px;
17      left: 100px;
18    }
19    #id2{
20      background-color: aqua;
21      width: 400px;
22      height: 100px;
23    }
24    #id3{
25      background-color: green;
26    }
27
28  </style>
29 </head>
30 <body>
31   <div id="id1">第一个div第一个div第一个div</div>
32   <div id="id2">第二个div第二个div第二个div</div>
33   <div id="id3">第三个div第三个div第三个div</div>
34

```

我设置红色 div 离网页顶边距离 50 个 px

我设置红色 div 离网页左边距离 100 个 px

这样也就证明了我上面说的，第一个 div 悬浮起来了，可以根据 top, left 选择位置浮动，下边的第二个 div 自动顶上来

如果我不想第二个 div 自动顶上来把第一个 div 的位置占了，该怎么做？

参数: relative

```

7   <style type="text/css">
8     div{
9       width: 200px;
10      height: 100px;
11      border:2px solid blue;
12    }
13    #id1{
14      background-color: red;
15      position: relative;
16      top: 50px;
17      left: 100px;
18    }
19    #id2{
20      background-color: aqua;
21      width: 400px;
22      height: 100px;
23    }
24

```

将 absolute 改为 relative

这样我第一个 div 是在页面定死了的，就算设置 div 在页面的位置也是定死了的，不是浮空的效果，所以第二个 div 无法自动占第一个 div 位置

这是 div 在页面布局的常用方法。

CSS 各种属性使用案例

```
<!DOCTYPE HTML>
<html>
<head>
    <title></title>
    <meta charset=UTF-8>

    <style type="text/css">

        </style>
    </head>
    <body>
        <div id="jw"></div>
        <div>九尾爆发</div>
    </body>
</html>
```

现在时这样感觉太丑，想给图文区域弄个边框

```
css-html.html
<!DOCTYPE html>
<html>
<head>
    <title></title>
    <meta charset=UTF-8>

    <style type="text/css">
        #bk1{
            width: 350px;
            height: 300px;
            border: 2px dashed orange;
        }
    </style>
</head>
<body>
<div id="bk1">
    <div id="jw"></div>
    <div>九尾爆发</div>
</div>
</body>

```

3. 设置边框
样式和颜色

2. 加个 div 把图片和
文字用边框包裹起来

1. 把图片设
置小点

position: absolute 把 div 或者标签，图片悬浮起来

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

    <style type="text/css">
        #bk1{
            width: 350px;
            height: 300px;
            border: 2px dashed orange;
        }
        #text{
            position: absolute;
            top: 70px;
            left: 70px;
            color: green;
        }
    </style>
</head>
<body>
    <div id="bk1">
        <div id="jw"></div>
        <div id="text">迪达拉</div>
    </div>

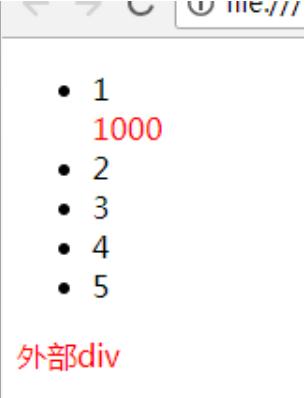
```



把文字悬浮在图片上

后代选择器设置 html 里面指定标签的颜色

```
7         <style type="text/css">
8
9             div{
10                 color: red;
11             }
12         </style>
13     </head>
14     <body>
15         <ul id="wrap">
16             <li>1
17                 <div>1000</div>
18             </li>
19             <li>2</li>
20             <li>3</li>
21             <li>4</li>
22             <li>5</li>
23         </ul>
24         <div>外部div</div>
25     </body>
```



正常情况下我 CSS 设置 div，那么不管是标签嵌套的 div 还是外部本身的 div 都会变色

除非给 div 加 id 属性或者 class 属性来解决

外部div

但是我不想加太多的 id 和 class，因为太难记住

标签 标签{} 这种就是后代选择器

```
7     <style type="text/css">
8
9         li div{
10             color: red;
11         }
12     </style>
13 </head>
14 <body>
15     <ul id="wrap">
16         <li>1
17             <div>1000</div>
18         </li>
19         <li>2</li>
20         <li>3</li>
21         <li>4</li>
22         <li>5</li>
23     </ul>
24     <div>外部div</div>
25 </body>
```

指定 li 标签里面嵌套的 div 标签，这就是后代选择器

这样就会指定到 li 标签里面的所以 div 标签，不会影响外部的 div 标签

外部div

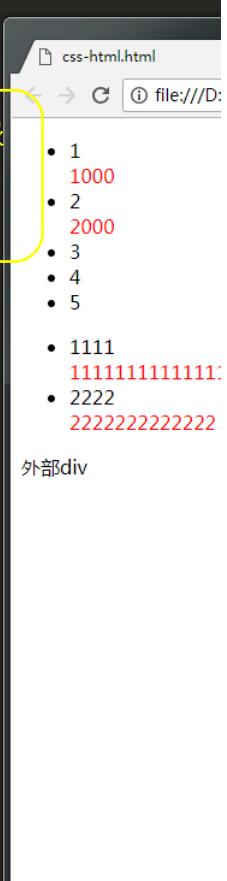
```
7     <style type="text/css">
8
9         li div{
10             color: red;
11         }
12     </style>
13 </head>
14 <body>
15     <ul id="wrap">
16         <li>1
17             <div>1000</div>
18         </li>
19         <li>2
20             <div>2000</div>
21         </li>
22         <li>3</li>
23         <li>4</li>
24         <li>5</li>
25     </ul>
26     <div>外部div</div>
27 </body>
```

但是后代选择器有个问题就是 li 指定的 div 嵌套标签都会受颜色影响

外部div

>子元素选择器，只选择第一层指定的标签，标签内部嵌套相同标签
不会被影响

```
7     <style type="text/css">
8         li > div{
9             color: red;
10        }
11    </style>
12 </head>
13 <body>
14     <ul>
15         <li>1
16             <div>1000</div>
17         </li>
18         <li>2
19             <div>2000</div>
20         </li>
21         <li>3</li>
22         <li>4</li>
23         <li>5</li>
24     </ul>
25
26     <ul>
27         <li>1111
28             <div>1111111111111111</div>
29         </li>
30         <li>2222
31             <div>22222222222222</div>
32         </li>
33     </ul>
34     <div>外部div</div>
35 </body>
```



指定 li 标签里面嵌套的 div 标签，需用>符号指定

外部div

+相邻标签选择器

```
6
7     <style type="text/css">
8
9         #ull > #first +.cla4{
10            color: red;
11        }
12    </style>
13 </head>
14 <body>
15     <ul id="ull">
16
17         <div class="cla1">1000</div>
18         <div class="cla2">2000</div>
19         <div id="first">3000</div>
20         <div class="cla4">4000</div>
21         <div class="cla5">5000</div>
22         <div>4</div>
23         <div>5</div>
24     </ul>
25     <div>外部div</div>
26 </body>
```



+号表示指定 ull 里面的 first 下面的 cla4 编号标签，这就是相邻选择器

```
<style type="text/css">  
    #ull > #first +.cla2{  
        color: red;  
    }  
</style>  
</head>  
<body>  
    <ul id="ull">  
  
        <div class="cla1">1000</div>  
        <div class="cla2">2000</div>  
        <div id="first">3000</div>  
        <div class="cla4">4000</div>  
        <div class="cla5">5000</div>  
    </ul>  
    <div>外部div</div>
```

first 编号的相邻选择器必须在 first 的下面，不允许是上面

~兄弟选择器，选择同一个标签下的 div，

```
7   <style type="text/css">  
8  
9     #ull #first ~ div{  
10       color: red;  
11     }  
12   </style>  
13 </head>  
14 <body>  
15   <div id="ull">  
16  
17     <div id="first">3000 </div>  
18     <div class="cla1">1000</div>  
19     <div class="cla2">2000</div>  
20  
21     <div class="cla4">4000 <div>真全栈少年</div> </div>  
22     <div class="cla5">5000</div>  
23     <div>4</div>  
24     <div>5</div>  
25   </div>  
26   <div>外部div</div>
```

选择 ull 选择器内 first 选择器以下的所有 div

标签[...] 属性选择器。比如 div 标签，我让 div 标签有同名的选择，没同名的不选择

```
7      <style type="text/css">
8
9          div[name]{
10              color: red;
11          }
12      </style>
13  </head>
14  <body>
15      <div id="ull">
16
17          <div name="xzz">1000</div>
18          <div name="xzz">2000</div>
19          <div name="xzz">3000</div>
20          <div>4000</div>
21          <div>5000</div>
22          <div>6000</div>
23          <div>7000</div>
24      </div>
25      <div>外部div</div>
26  </body>
```

1.给 div 属性 name 赋上值 xzz

2.指定 div 标签，然后在[]写入要修改颜色的部分 div 标签

1000
2000
3000
4000
5000
6000
7000
外部div

属性选择器之值选择器

```
<head>
    <title></title>
    <meta charset=UTF-8>

    <style type="text/css">
        div[name]{
            color: red;
        }
    </style>
</head>
<body>
    <div id="ull">
        <div name="xzz0">1000</div>
        <div name="xzz1">2000</div>
        <div name="xzz2">3000</div>
        <div>4000</div>
    </div>
</body>
```

属性选择器我们直接指定 div 的 name，就选择了 div 带有 name 的标签

1000
2000
3000
4000
5000
6000
7000
外部div

```
<head>
    <title></title>
    <meta charset=UTF-8>

    <style type="text/css">
        div[name="xzz0"]{
            color: red;
        }
    </style>
</head>
<body>
    <div id="ull">

        <div name="xzz0">1000</div>
        <div name="xzz1">2000</div>
        <div name="xzz2">3000</div>
        <div>4000</div>
        <div>5000</div>
        <div>6000</div>
    
```

值选择器在 name 后面加入具体字符，就选择指定字符的标签

```
6
7     <style type="text/css">
8
9         div[name~="xzz"]{
10             color: red;
11         }
12     </style>
13 </head>
14 <body>
15     <div id="ull">

16
17         <div name="xzz 0">1000</div>
18         <div name="xzz 1">2000</div>
19         <div name="xzz2">3000</div>
20         <div>4000</div>
21         <div>5000</div>
22         <div>6000</div>
    
```

~=符号是选择 name 具有 xzz 字符，而且还带有空格的标签

自定义字体

```
<body>
  <div id="wrap">
    <div id="test">TTF字体</div>
  </div>
</body>
```

这默认的字体

```
<title></title>
<meta charset=UTF-8>

<style type="text/css">
  @font-face{
    font-family: "xzz";
    src:url(ht/XZZ.TTF);
  }
  #test{
    font:50px "xzz";
  }

</style>
</head>
<body>
  <div id="wrap">
    <div id="test">TTF字体</div>
  </div>
</body>
```

怎么实现的？



TTF字体

这是自定义字体

```
7   <style type="text/css">
8     @font-face{
9       font-family: "xzz";
10      src:url(ht/XZZ.TTF);
11    }
12    #test{
13      font:50px "xzz";
14    }
15
16  </style>
17 </head>
18 <body>
19   <div id="wrap">
20     <div id="test">TTF字体</div>
21   </div>
22 </body>
```

去网上下载个 TTF 格式的字体在同级目录下

给@font-face 取个可以被调用的名字，这样才可以被 font 调用

@font-face 格式的代码会使用 src:url 去找指定目录下的字体样式来嵌入到你的字体内

设置字体样式的时候调用定义好名字的 xzz 元素

这种 url 调用字体库的方式，增加网络负担，但是先实现最重要，至于网络问题再优化。

用别人做好的字体图标

<https://icomoon.io/> 去这个网站下载喜欢的图标

The screenshot shows the IcoMoon App interface. At the top, there's a purple header with the text "IcoMoon App". Below it is a sidebar with four orange-outlined icons: a house, a play button, a file, and a price tag. A green callout points to the house icon with the text "点击选择你要的图标" (Click to select the icon you want). To the right of the sidebar is a main area with a "Generate Font" button and a preview of the generated font characters. A green callout points to the "Generate Font" button with the text "然后点击右下角这个" (Then click this at the bottom right). Below this is a preview of the generated font characters: "home", "play", "file-empty", and "price-tags". A green callout points to this area with the text "生成图标" (Generate icon). At the bottom, there's a "Font" tab, a "Download" button, and a gear icon. A green callout points to the "Download" button with the text "点击右下角下载" (Click the bottom right to download). On the left side, there's a file browser interface showing a folder named "新建文件夹" (New Folder) containing files: "demo-files", "fonts", "demo", "Read Me", "selection.json", and "style". A green callout points to the "style" file with the text "解压后生成 SVG 类型的文件，放在 html 目录下，然后 html 调用 style 文件就可以了" (Extract after generating an SVG type file, place it in the html directory, and then use the style file in the html).

点击右上角 APP

点击选择你要的图标

然后点击右下角这个

生成图标

Font

Download

点击右下角下载

解压后生成 SVG 类型的文件，放在 html 目录下，然后 html 调用 style 文件就可以了

html 调用方式如下

```
css-html.html    style.css
7   url('fonts/icomoon.svg?anbfls#icomoon') format('svg');
8   font-weight: normal;
9   font-style: normal;
0 }
1
2 [class^="icon-"], [class*=" icon-"] {
3   /* use !important to prevent issues with browser extensions that change fonts */
4   font-family: 'icomoon' !important;
5   speak: none;
6   font-style: normal;
7   font-weight: normal;
8   font-variant: normal;
9   text-transform: none;
0   line-height: 1;
1
2   /* Better Font Rendering ===== */
3   -webkit-font-smoothing: antialiased;
4   -moz-osx-font-smoothing: grayscale;
5 }
6
7 .icon-home:before {
8   content: "\e900";
9 }
10 .icon-play:before {
11   content: "\e912";
12 }
13 .icon-file-empty:before {
14   content: "\e924";
15 }
16 .icon-price-tags:before {
17   content: "\e936";
18 }
```

这就是生成的 style 代码，这个 icon-home 名字就代表一个图标

```
<html>
<head>
  <title></title>
  <meta charset=UTF-8>
  <link rel="stylesheet" type="text/css" href="SVG/style.css">
</head>
<body>
  <div class="icon-home"></div>
</body>
```

用 link 包含生成 style.css 文件路径

在标签里面用 class 包含图标名称就指定显示什么图标

我感觉这个图标太小

```
4   <title></title>
5   <meta charset=UTF-8>
6
7   <link rel="stylesheet" type="text/css" href="SVG/style.css">
8   <style type="text/css">
9     .xzz{
10       font-size: 100px;
11     }
12   </style>
13 </head>
14 <body>
15   <div class="icon-home xzz"></div>
16 </body>
```

2. 设置字体大小

1. 给字体多增加一个名称

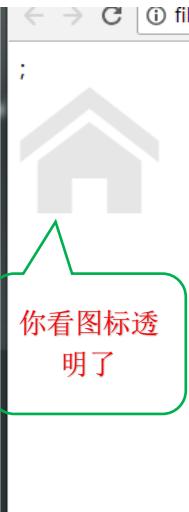


因为图标也归为字体，所以就变大了

在 CSS 中给图标增加 opacity，可以让图标透明

```
5      <meta charset=UTF-8>
6
7      <link rel="stylesheet" type="text/css" href="style.css"/>
8      <style type="text/css">
9          .xzz{
10              font-size: 100px;
11              opacity: 0.1;
12          }
13      </style>;
14  </head>
15  <body>
16      <div class="icon-home xzz"></div>
17  </body>
```

图标透明度 0~1 区间



你看图标透
明了

```
5      <meta charset=UTF-8>
6
7      <link rel="stylesheet" type="text/css" href="SVG/style.css">
8      <style type="text/css">
9          .xzz{
10              font-size: 100px;
11              opacity: 0.5;
12          }
13      </style>;
14  </head>
15  <body>
16      <div class="icon-home xzz"></div>
17  </body>
```

透明度减少



Background:rgb(...) CSS 设置标签背景颜色和透明

```
7      <link rel="stylesheet" type="text/css" href="style.css"/>
8      <style type="text/css">
9          .xzz{
10              font-size: 100px;
11              background:rgb(255,149,202,0.5);
12          }
13      </style>;
14  </head>
15  <body>
16      <div class="icon-home xzz"></div>
17  </body>
```

这是 RGB 颜色

这是透明度

这个颜色怎
么来的?



windows 画图调色板选择的

```

8     <style type="text/css">
9         .xzz{
10             font-size: 100px;
11             background:rgb(255,149,202,0.5);
12             color: rgba(128,0,64,1);
13         }
14     </style>;
15 </head>
16 <body>
17     <div class="icon-home xzz"></div>
18 </body>

```

这是设置图标
的前景颜色

这是前景颜色

参数设置和背景颜色一样，在 windows 调色板找

```

8     <style type="text/css">
9         .xzz{
10             font-size: 100px;
11             background:rgb(255,149,202,0.5);
12             color: rgba(128,0,64,0.5);
13         }
14     </style>;
15 </head>
16 <body>
17     <div class="icon-home xzz"></div>
18 </body>

```

设置前景颜色透明度

text-shadow CSS 设置字体阴影

```

8     <style type="text/css">
9         *{
10             margin: 0;
11             padding: 0;
12         }
13     .xzz{
14         text-align: center; /*字体居中*/
15         font:100px/200px "微软雅黑";
16         text-shadow: pink 10px 10px;
17     }
18     </style>
19 </head>
20 <body>
21     <div class="xzz">字体阴影设置</div>
22 </body>

```

字体阴影颜色

字体阴影偏移量，必须要有，因为阴影是根据原字体偏移出来的

感觉这种阴影不是很好看，给阴影加模糊效果。

text-align:center 字体居中

```

8     <style type="text/css">
9         *{
10             margin: 0;
11             padding: 0;
12         }
13     .xzz{
14         text-align: center; /*字体居中*/
15         font:100px/200px "微软雅黑";
16         text-shadow: pink 10px 10px 10px;
17     }
18     </style>
19 </head>
20 <body>
21     <div class="xzz">字体阴影设置</div>
22 </body>

```

第三个 px 就是模糊效果

```
8     <style type="text/css">
9         *{
10             margin: 0;
11             padding: 0;
12         }
13         .xzz{
14             text-align: center; /*字体居中*/
15             font: 100px/200px "微软雅黑";
16             text-shadow: pink 10px 10px 5px;
17         }
18     </style>
19 </head>
20 <body>
21     <div class="xzz">字体阴影设置</div>
22 </body>
```

这是两层模糊效果，其实还可以增加多层

```
8     <style type="text/css">
9         *{
10             margin: 0;
11             padding: 0;
12         }
13         .xzz{
14             text-align: center; /*字体居中*/
15             font: 100px/200px "微软雅黑";
16             text-shadow: pink 10px 10px 5px, gray 20px 20px 10px;
17         }
18     </style>
19 </head>
20 <body>
21     <div class="xzz">字体阴影设置</div>
22 </body>
```

数字越小模糊效果越小

字体阴影设置

用逗号分隔开，后面就是第3层的模糊偏移量和样式

```
8     <style type="text/css">
9         *{
10             margin: 0;
11             padding: 0;
12         }
13         .xzz{
14             text-align: center; /*字体居中*/
15             font: 100px/200px "微软雅黑";
16             color: white; /*1.先让字体为白色*/
17             text-shadow: black 1px 1px 10px;
18         }
19     </style>
20 </head>
21 <body>
22     <div class="xzz">浮雕文字</div>
23 </body>
```

本来白色是看不见的，但是用阴影把白色分离出来了

浮雕文字

设置鼠标移动到字体后的模糊效果

```
8     <style type="text/css">
9         *{
10             margin: 0;
11             padding: 0;
12         }
13         .xzz{
14             text-align: center; /*字体居中*/
15             font: 100px/200px "微软雅黑";
16             color: black;
17         }
18         div:hover{
19             color: white;
20             text-shadow: black 0 0 100px;
21         }
22     </style>
23 </head>
24 <body>
25     <div class="xzz">浮雕文字</div>
26 </body>
```

这是鼠标没有移动到字体时候的情况

浮雕文字

这是鼠标移动到字体后的情况

但是感觉字体变化太突然了，能不能字体变化慢一点

transition:1s

```
8   <style type="text/css">
9     *{
10       margin: 0;
11       padding: 0;
12     }
13     .xzz{
14       text-align: center; /*字体居中*/
15       font: 100px/200px "微软雅黑";
16       color: black;
17       transition: 1s;
18     }
19     div:hover{
20       color: white;
21       text-shadow: black 0 0 100px;
22     }
23   </style>
24 </head>
25 <body>
26   <div class="xzz">浮雕文字</div>
27 </body>
```



在字体样式这里增加了 1 秒
变化的延时，这个变化时间
必须配合 hover 才行，因为
hover 里有变化的样式

图片标签后面背景如何嵌入模糊图片

```
7
8   <style type="text/css">
9     *{
10       margin: 0;
11       padding: 0;
12     }
13
14
15   </style>
16 </head>
17 <body>
18   <div id="xzz">
19     
20   </div>
21 </body>
```



先导入图片

```
8   <style type="text/css">
9
10    #xzz{
11      height: 100px;
12      background: pink;
13    }
14
15
16   </style>
17 </head>
18 <body>
19   <div id="xzz">
20     
21   </div>
22 </body>
```



修改图片尺
寸后发现没
有居中

```
8     <style type="text/css">
9
10    #xzz{
11        height: 100px;
12        background: pink;
13    }
14    img{
15        margin: 24px 0 0 24px;
16    }
17
18 </style>
19 </head>
20 <body>
21     <div id="xzz">
22         
23     </div>
24 </body>
```



position:relative 把标签定死在页面

```
8     <style type="text/css">
9
10    #xzz{
11        height: 100px;
12        background: rgb(247,134,221,0.7);
13        position: relative;
14    }
15    img{
16        margin: 20px 0 0 10px;
17    }
18
19 </style>
20 </head>
21 <body>
22     <div id="xzz">
23         
24         <div id="bg">
25             </div>
26         </div>
27     </div>
28 </body>
```



margin:10px 10px 10px 10px 设置标签离父容器框的上下左右距离

```
4     }
5
6     img{
7         margin: 25px 0 0 20px;
8     }
```



```

15      #xzz #bg{
16          position: absolute; /*让这个id为bg的div悬空*/
17          left: 30px; /*bg的div离左边距离30px*/
18          right: 0;
19          top: 0;
20          bottom: 0;
21          background: black;
22      }
23
24      img{
25          margin: 25px 0 0 20px;
26      }
27
28
29  
```

```

30 </head>
31 <body>
32     <div id="xzz">
33         
34         <div id="bg">←
35
36     </div>
37 </div>
38 </body>

```

很明显这里设置的是内嵌套的div，这个div用来放模糊背景

所以我们应该设置该div的left为0

```

#xzz #bg{
    position: absolute; /*让这个id为bg的div悬空*/
    left: 0; /*bg的div离左边距离0*/
    right: 0;
    top: 0;
    bottom: 0;
    background: black;
}

```

但是我们发现内嵌div的黑色把外部div的粉色遮挡住了。

```

#xzz #bg{
    position: absolute; /*让这个id为bg的div悬空*/
    left: 0; /*bg的div离左边距离0*/
    right: 0;
    top: 0;
    bottom: 0;
    background: black;
    z-index: -1; /*内嵌bg id的div在xzz id之下，所以写-1*/
}

```

因为粉色div是0.7半透明的，所
以后面的黑色也能看到点

```

#xzz{
    height: 100px;
    background: rgb(247,134,221,0.7);
}

```

```

9      #xzz{
10         height: 100px;
11         background: rgb(247,134,221,0.7);
12         position: relative;
13     }
14     #xzz #bg{
15         position: absolute; /*让这个id为bg的div悬空*/
16         left: 0; /*bg的div离左边距离0*/
17         right: 0;
18         top: 0;
19         bottom: 0;
20         background: url(git.jpg);
21         z-index: -1; /*内嵌bg id的div在xzz id之下，所以写-1*/
22     }
23
24     img{
25         margin: 25px 0 0 20px;
26     }
27
28     </style>
29 </head>
30 <body>
31     <div id="xzz">
32         
33         <div id="bg">
34
35             </div>
36         </div>
37     </div>
38 </body>

```



```

#xzz #bg{
    position: absolute; /*让这个id为bg的div悬空*/
    left: 0; /*bg的div离左边距离0*/
    right: 0;
    top: 0;
    bottom: 0;
    background: url(git.jpg);
    z-index: -1; /*内嵌bg id的div在xzz id之下，所以写-1*/
    background-size: 100% 100%;
}

```



但是我们背景没有模糊啊

```

#xzz #bg{
    position: absolute; /*让这个id为bg的div悬空*/
    left: 0; /*bg的div离左边距离0*/
    right: 0;
    top: 0;
    bottom: 0;
    background: url(git.jpg);
    z-index: -1; /*内嵌bg id的div在xzz id之下，所以写-1*/
    background-size: 100% 100%;
    filter: blur(5px);
}

```



文字描边

```

8 <style type="text/css">
9     h1{
10         color: white;
11         -webkit-text-stroke:pink 3px;
12     }
13 </style>
14 </head>
15 <body>
16     <h1>文字描边</h1>
17 </body>

```

设置描边颜色粗细

文字描边

盒子模型阴影，也就是标签块阴影

```
8 <style type="text/css">
9   *{
10     margin: 0;
11     padding: 0;
12   }
13   #test{
14     position: absolute;
15     right: 0;
16     left: 0;
17     top: 0;
18     bottom: 0;
19     margin: auto; /*该属性是让div块居中*/
20     width: 100px; /*但是div块居中必须有指定div块的宽，高才行*/
21     height: 100px;
22     background: pink;
23   }
24 </style>
25 </head>
26 <body>
27   <div id="test">盒模型阴影</div>
28 </body>
```

盒模型阴影

```
8 <style type="text/css">
9   *{
10     margin: 0;
11     padding: 0;
12   }
13   #test{
14     position: absolute;
15     right: 0;
16     left: 0;
17     top: 0;
18     bottom: 0;
19     margin: auto; /*该属性是让div块居中*/
20     width: 100px; /*但是div块居中必须有指定div块的宽，高才行*/
21     height: 100px;
22     background: pink;
23     text-align:center; /*div里面的字体水平居中*/
24     line-height: 100px; /*div里面的字体垂直居中*/
25   }
26 </style>
27 </head>
28 <body>
29   <div id="test">盒模型阴影</div>
30 </body>
```

盒模型阴影

```
#test{
  position: absolute;
  right: 0;
  left: 0;
  top: 0;
  bottom: 0;
  margin: auto; /*该属性是让div块居中*/
  width: 100px; /*但是div块居中必须有指定div块的宽，高才行*/
  height: 100px;
  background: pink;
  text-align:center; /*div里面的字体水平居中*/
  line-height: 100px; /*div里面的字体垂直居中*/

  box-shadow: 10px 10px black; /*在div框背后增加一层阴影*/
}
```

盒模型阴影

这是阴影相对于 div 块的距离和颜色

用 box-shadow
给 div 块增加
相同大小的阴
影

```

margin: auto; /*该属性是在div块居中*/
width: 100px; /*但是div块居中必须有指定div块的宽，高才行*/
height: 100px;
background: pink;
text-align:center; /*div里面的字体水平居中*/
line-height: 100px; /*div里面的字体垂直居中*/

box-shadow: 10px 10px 10px black; /*在div框背后增加一层阴影*/

```

盒模型阴影

多加一个产生就是给阴影增加模糊效果

```

margin: auto; /*该属性是在div块居中*/
width: 100px; /*但是div块居中必须有指定div块的宽，高才行*/
height: 100px;
background: pink;
text-align:center; /*div里面的字体水平居中*/
line-height: 100px; /*div里面的字体垂直居中*/

box-shadow: 10px 10px 10px 10px black; /*在div框背后增加一层

```

盒模型阴影

第4个参数是增加阴影向外扩充的大小

```

background: pink;
text-align:center; /*div里面的字体水平居中*/
line-height: 100px; /*div里面的字体垂直居中*/

box-shadow: inset 10px 10px 10px 0px black; /*在div框背后增加一

```

盒模型阴影

Inset 增加阴影向顶层，div 向内侧扩散

第4个参数就变成了阴影
向顶层扩散的大小

```

text-align:center; /*div里面的字体水平居中*/
line-height: 100px; /*div里面的字体垂直居中*/

box-shadow: inset 10px 10px 10px 10px black; /*在div框背后增加一

```

盒模型阴影

```

height: 100px;
background: pink;
text-align:center; /*div里面的字体水平居中*/
line-height: 100px; /*div里面的字体垂直居中*/

box-shadow: 10px 10px 10px black, 20px 20px 10px 10px deeppink; /*在div框背后增加多层阴影*/

```

盒模型阴影

用，逗号分割可以多加几层阴影

```

width: 100px; /*但是div块居中必须有指定div块的宽，高才行*/
height: 100px;
background: pink;
text-align:center; /*div里面的字体水平居中*/
line-height: 100px; /*div里面的字体垂直居中*/

box-shadow: -10px -10px 10px 10px black, 20px 20px 10px 10px deeppink; /*在div框背后增加多层阴影*/

```

盒模型阴影

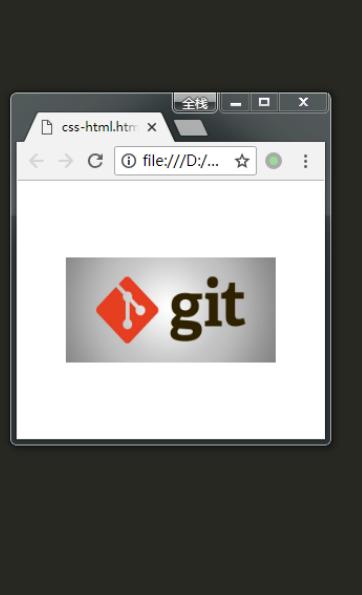
值是负号，那么块就是向
反方向扩散

这个负号是什么方向，网上查找 CSS 坐标系统基础

margin:auto 图片水平居中

```
8     <style type="text/css">
9         *{
10             margin: 0;
11             padding: 0;
12         }
13         img{
14             position: absolute;
15             left: 0;
16             right: 0;
17             bottom: 0;
18             top: 0;
19             margin: auto;
20         }
21     </style>
22 </head>
23 <body>
24     
25 </body>
```

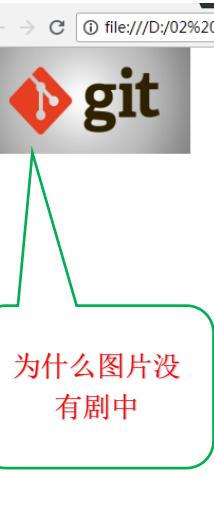
这个 auto 除了让盒子 div 块居中外，也能让图片居中



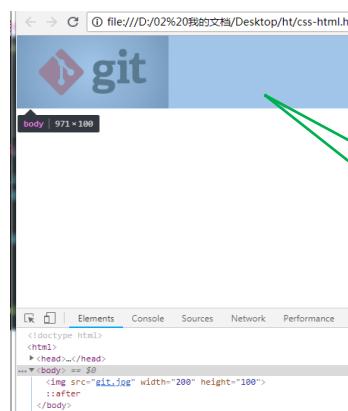
但是这种方法不能用，因为动了层，先用 absolute 将图片悬浮起来，然后在居中。

有个好方法，我们把 body 假当做一个容器，然后让图片相对于容器来居中

```
8     <style type="text/css">
9         *{
10             margin: 0;
11             padding: 0;
12         }
13
14         body:after{
15             height: 100%;
16             content: "";
17             display: inline-block; /*指定行内块元素*/
18             vertical-align: middle; /*把body块放在页面中部*/
19         }
20         img{
21             vertical-align: middle;
22         }
23     </style>
24 </head>
25 <body>
26     
27 </body>
```



为什么图片没
有剧中

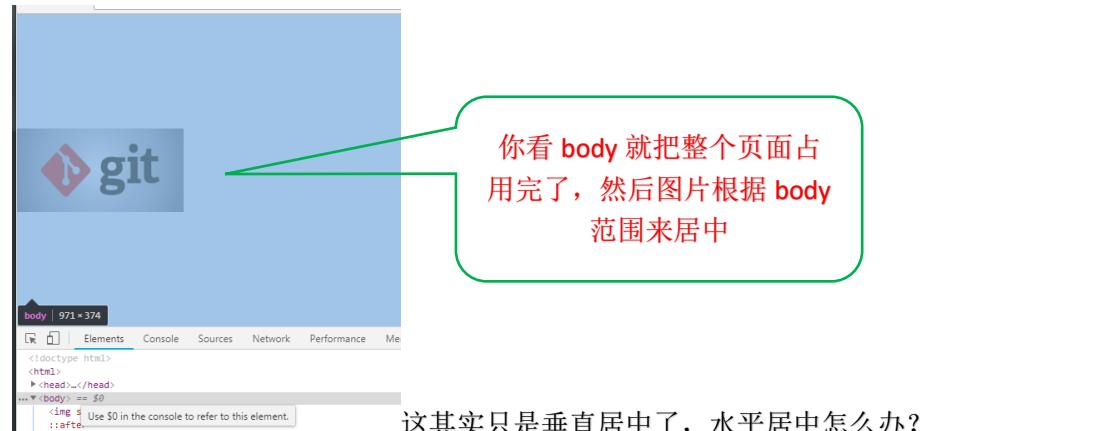


检查发现 body 容器是整个
页面的一行，而不是整个页
面

```

3     html,body{
4         height: 100%; /*body块容器占html页面的100%范围*/
5     }
6     body:after{
7         height: 100%;
8         content: "";
9         display: inline-block; /*指定行内块元素*/
10        vertical-align: middle; /*把body块放在页面中部*/
11    }
12    img{
13        vertical-align: middle;
14    }
15 
```

这个就让 **body** 占完了页面区域，
然后图片根据 **body** 来居中

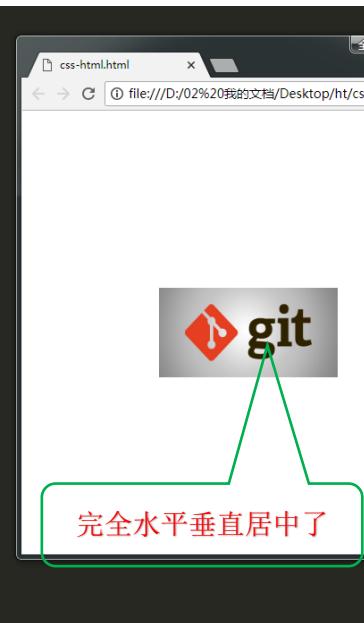


这其实只是垂直居中了，水平居中怎么办？

```

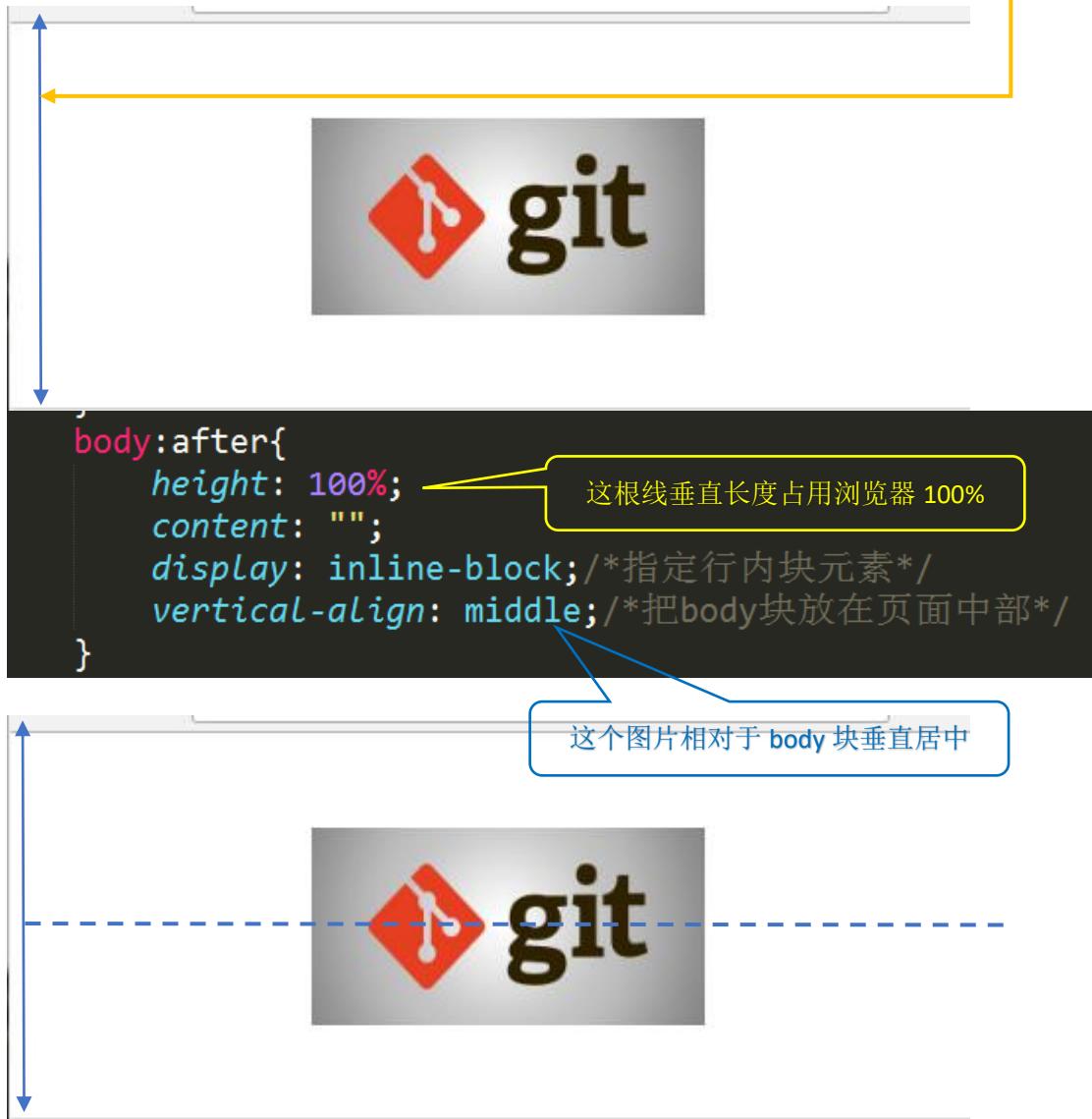
<style type="text/css">
*{
    margin: 0;
    padding: 0;
}
html,body{
    height: 100%; /*body块容器占html页面的100%范围*/
}
body{
    text-align: center; /*水平居中*/
}
body:after{
    height: 100%;
    content: "";
    display: inline-block; /*指定行内块元素*/
    vertical-align: middle; /*把body块放在页面中部*/
}
img{
    vertical-align: middle;
}
</style>
</head>
<body>
    
</body>

```



详解 **after** 伪元素原理

```
body:after{  
    height: 100%;  
    content: "";  
    display: inline-block; /*指定行内块元素*/  
    vertical-align: middle; /*把body块放在页面中部*/  
}
```



标签图片倒影

```
1     content: ,  
2     display: inline-block; /*指定行内块元素*/  
3     vertical-align: middle; /*把body块放在页面中部*/  
4   }  
5   img{  
6     vertical-align: middle;  
7     -webkit-box-reflect: above;  
8   }  
9   </style>  
0 </head>  
1 <body>  
2     
3 </body>
```

该属性是倒影， above 向上倒影



```

24     }
25     img{
26       vertical-align: middle;
27       -webkit-box-reflect:left;
28   }
29 </style>
30 </head>
31 <body>
32   
33 </body>

```

Left 向左倒影



至于其它方向倒影参数，查阅百度。

```

        img{
          vertical-align: middle;
          -webkit-box-reflect:left 10px;
      }
    </style>
</head>
<body>

```

如果你需要倒影间隔，在这里加间隔像素



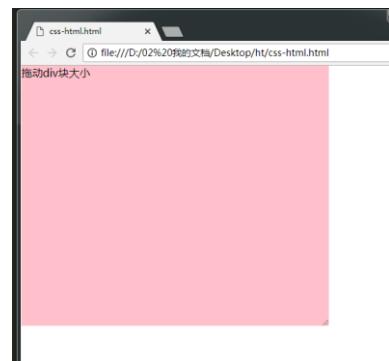
鼠标拖动标签块或者 div 块大小的功能实现

```

8   <style type="text/css">
9     *{
10       margin: 0;
11       padding: 0;
12     }
13     div{
14       display: inline-block;
15       width: 200px;
16       height: 200px;
17       background: pink;
18       vertical-align: middle;
19       resize: both;
20       overflow: auto;
21     }
22
23   </style>
24 </head>
25 <body>
26   <div>拖动div块大小</div>
27 </body>

```

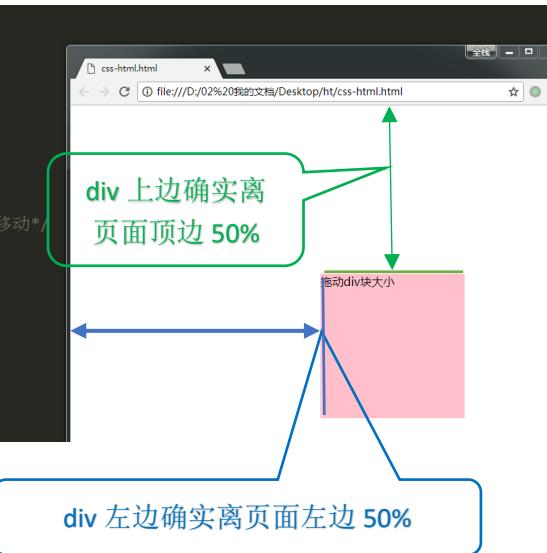
加 resize 属性的 both 值
还必须加入 overflow 的 auto 值



让标签块或者 div 块在页面居中的一种方法

```
8 <style type="text/css">
9   *{
10     margin: 0;
11     padding: 0;
12   }
13   div{
14     width: 200px;
15     height: 200px;
16     background: pink;
17     position: absolute; /*必须将div块悬浮起来才能移动*
18     left: 50%;/*div左边沿占用页面的左边50%*/
19     top: 50%;/*div上边沿占用页面的顶边50%*/
20   }
21 </style>
22 </head>
23 <body>
24   <div>拖动div块大小</div>
25 </body>
```

通常 `left` 单独在 [CSS](#) 中设置无效，
需要在使用 `position` 属性下使用才能生效起作用



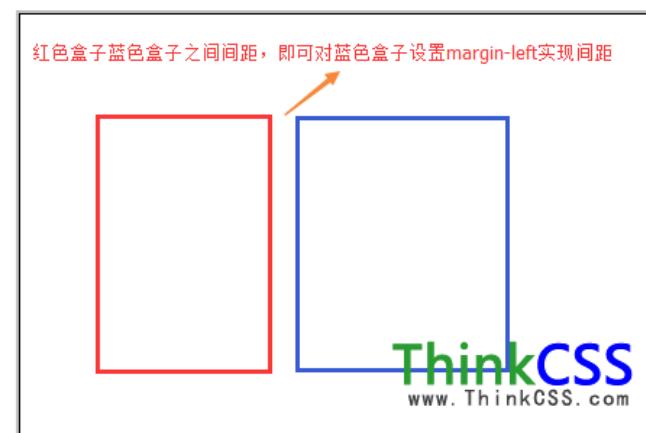
但是我看起 `div` 块并没有页面居中，那是因为 `left` 和 `top` 属性是对 `div` 边界居中的，块的大小就会像居中后面移动。

这种情况我就需要用 `margin` 属性来微调

```
13   <style type="text/css">
14     div{
15       width: 200px;
16       height: 200px;
17       background: pink;
18       position: absolute;
19       left: 50%;
20       top: 50%;
21       margin-left: -100px;
22       margin-top: -100px;
23     }
24   </style>
25 </head>
26 <body>
27   <div>拖动div块大小</div>
28 </body>
```

这样就对了

`margin-left`通常用于两个相邻对象，设置两个对象左右之间间距排版。（扩展教程[margin-right](#)）



可用`margin-left`设置两个盒子之间间隔间距

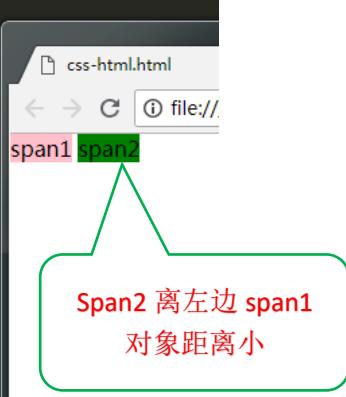
```
13     .div1{
14         width: 100px;
15         height: 100px;
16         background: pink;
17     }
18     .div2{
19         width: 100px;
20         height: 100px;
21         background: green;
22         margin-left:20px
23     }
24
25
26     </style>
27 </head>
28 <body>
29     <div class="div1">div1</div>
30     <div class="div2">div2</div>
31 </body>
```



让 div2 离左边对象 20px 距离

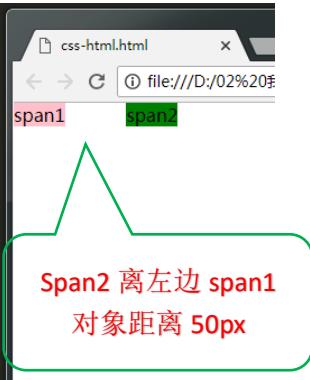
比如我们用 span 试一下

```
12
13     .span1{
14
15         background: pink;
16
17     }
18     .span2{
19
20         background: green;
21         /*margin-left:20px*/
22     }
23
24     </style>
25 </head>
26 <body>
27     <span class="span1">span1</span>
28     <span class="span2">span2</span>
29 </body>
```



Span2 离左边 span1 对象距离小

```
13     .span1{
14
15         background: pink;
16
17     }
18     .span2{
19
20         background: green;
21         margin-left:50px
22     }
23
24     </style>
25 </head>
26 <body>
27     <span class="span1">span1</span>
28     <span class="span2">span2</span>
29 </body>
```



Span2 离左边 span1 对象距离 50px

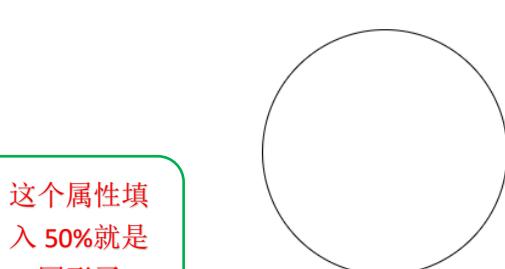
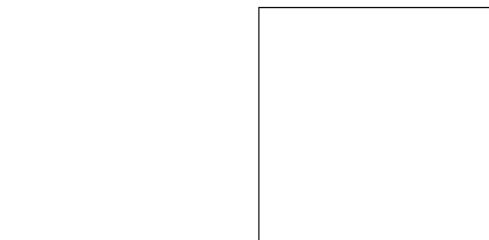
div 块方形倒圆角

```
<style type="text/css">
  *{
    margin: 0;
    padding: 0;
  }
  #test{
    width: 200px;
    height: 200px;
    border: 1px solid;
    position: absolute;
    left: 0;
    right: 0;
    top: 0;
    bottom: 0;
    margin: auto;
  }

</style>
</head>
<body>
  <div id="test"></div>
</body>
```

这是一个方形自动居中

```
#test{
  width: 200px;
  height: 200px;
  border: 1px solid;
  position: absolute;
  left: 0;
  right: 0;
  top: 0;
  bottom: 0;
  margin: auto;
  border-radius: 50%;
```

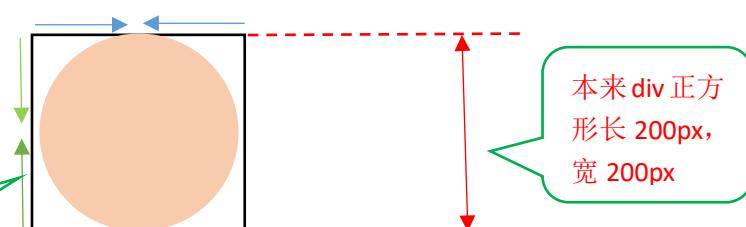


这个属性填入 50% 就是圆形了

原理是什么呢？

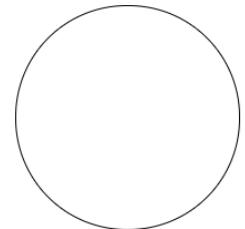
50%的意思是 width 和 height 的 px 值减小 50%

那么 width 和 height 就是 100px



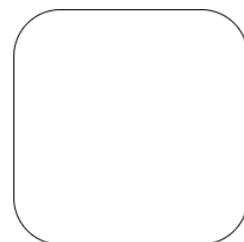
经过每条边向里面缩短 50%，也就是 100px，但是还要保持每条边相连，就形成了圆形了

```
13     #test{  
14         width: 200px;  
15         height: 200px;  
16         border: 1px solid;  
17         position: absolute;  
18         left: 0;  
19         right: 0;  
20         top: 0;  
21         bottom: 0;  
22         margin: auto;  
23         border-radius: 100px;  
24     }  
25
```



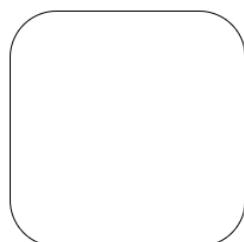
boder-radius 改成 100px 是一样的

```
3     #test{  
4         width: 200px;  
5         height: 200px;  
6         border: 1px solid;  
7         position: absolute;  
8         left: 0;  
9         right: 0;  
10        top: 0;  
11        bottom: 0;  
12        margin: auto;  
13        border-radius: 20%;  
14    }  
15
```



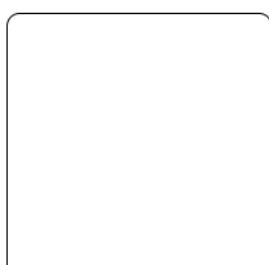
四边都向里缩进了 20%，200px 的 20%也就是向里缩进了 40px

```
#test{  
    width: 200px;  
    height: 200px;  
    border: 1px solid;  
    position: absolute;  
    left: 0;  
    right: 0;  
    top: 0;  
    bottom: 0;  
    margin: auto;  
    border-radius: 40px;  
}
```



一样的

```
#test{  
    width: 200px;  
    height: 200px;  
    border: 1px solid;  
    position: absolute;  
    left: 0;  
    right: 0;  
    top: 0;  
    bottom: 0;  
    margin: auto;  
    border-radius: 10px;  
}
```



10px 倒角就很小了

如何将 div 方形制作成椭圆形，借鉴 div 方形倒圆角知识

The screenshot shows a portion of a CSS file. It includes a style block for a div with id 'test' and a body section containing a div with id 'test'. The style block contains properties like width, height, border, position, left, right, top, bottom, margin, and border-radius. The border-radius property is set to 50px 50px 50px;.

Annotations in the code editor:

- A yellow callout points to the 'border-radius: 50px 50px 50px;' line with the text '右下角' (Bottom-right corner).
- A yellow callout points to the opening brace of the style block with the text '左上角' (Top-left corner).
- A yellow callout points to the closing brace of the style block with the text '右上角和左下角' (Top-right and bottom-left corners).

The screenshot shows a browser window displaying a single rounded square div with a black border and white background.

transform2D/3D 旋转属性，做一个旋转的花瓣案例

The screenshot shows a portion of a CSS file. It includes a style block for a div with id 'wrap' and a body section containing a div with id 'wrap'. The style block for #wrap contains properties like width, height, border, position, left, right, top, bottom, and margin. The style block for #wrap > div contains margin, width, height, border, background, float, and box-sizing: border-box.

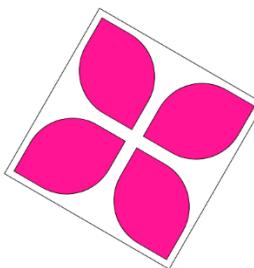
The screenshot shows a browser window displaying a 2x2 grid of four pink squares. Each square has a black border and is rotated 45 degrees clockwise.

Annotations in the code editor:

- A yellow callout points to the border and margin properties of the #wrap style with the text '先将顶层 div 加上边框和居中' (First add borders and center the top-level div).
- A green callout points to the box-sizing property of the #wrap > div style with the text '如果我们设置一个宽度为 200px，而实际呈现的盒子的宽度可能会大于 200px(除非没有左右边框和左右补白)。这可能看起来比较怪，CSS 设置的宽度仅仅是内容区的宽度，而非盒子的宽度' (If we set a width of 200px, the actual box width may be greater than 200px (unless there are no left and right borders and padding). This may look strange, as the CSS width is only the content area width, not the box width). It also includes the text '当你设置 box-sizing:border-box 以后，这就能达到你想要的目的。例如，我们想要一个宽度为 200px 的盒子，那么我们直接设置宽度为 200px，所以 box-sizing 是设置兼容属性' (After setting box-sizing: border-box, it can achieve the desired purpose. For example, if we want a box width of 200px, we directly set the width to 200px, so box-sizing is a compatibility attribute).

```
#wrap >div{  
    margin: 10px;  
    width: 130px;  
    height: 130px;  
    border: 1px solid;  
    background: deeppink;  
    float: left;  
    box-sizing: border-box;  
}  
#wrap >div:nth-child(1), #wrap >div:nth-child(4) /*选择大div内嵌入的第1行div和第4行div*/  
{  
    border-radius: 0 60%; /*设置第1行div和第4行div的图形*/  
}  
#wrap >div:nth-child(2), #wrap >div:nth-child(3) /*选择大div内嵌入的第2行div和第3行div*/  
{  
    border-radius: 60% 0; /*设置第2行div和第3行div的图形*/  
}  
#wrap:hover{  
    transform: rotate(120deg);  
}
```

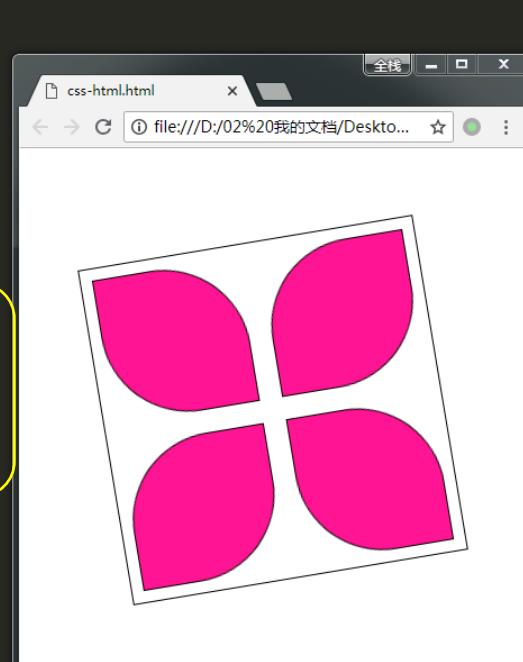
transform 属性向元素应用 2D 或 3D 转换。该属性允许您对元素进行旋转、缩放、移动或倾斜。



rotate 为 2d 旋转, 120deg 是选择角度参数

```
#wrap{  
    width: 300px;  
    height: 300px;  
    border: 1px solid;  
    position: absolute;  
    left: 0;  
    right: 0;  
    top: 0;  
    bottom: 0;  
    margin: auto;  
    transition: 2s  
}  
#wrap >div{  
    margin: 10px;  
    width: 130px;  
    height: 130px;  
    border: 1px solid;  
    background: deeppink;  
    float: left;  
    box-sizing: border-box;  
}  
#wrap >div:nth-child(1), #wrap >div:nth-child(4)
```

加这个参数
允许 120 度
不停旋转 2
秒



用 border-image-source 建立边框图片



```
<style type="text/css">
*{
    margin: 0;
    padding: 0;
}
#test
{
    position: absolute;
    left: 0;
    right: 0;
    top: 0;
    bottom: 0;
    margin: auto;
    width: 200px;
    height: 200px;
    border: 30px solid;
    border-image-source: url(jgg.jpg);
}

</style>
</head>
<body>
    <div id="test">
    </div>

```

但是发现不是我们想要的效果

```
#test
{
    position: absolute;
    left: 0;
    right: 0;
    top: 0;
    bottom: 0;
    margin: auto;
    width: 200px;
    height: 200px;
    border: 30px solid;
    border-image-source: url(jgg.jpg);
    border-image-slice: 33.333%;
}
```

用 border-image-slice 来切9宫格，这里的33.333%是什么意思？

这不是我们想要的结果



border-image-slice 应该有 4 个参数



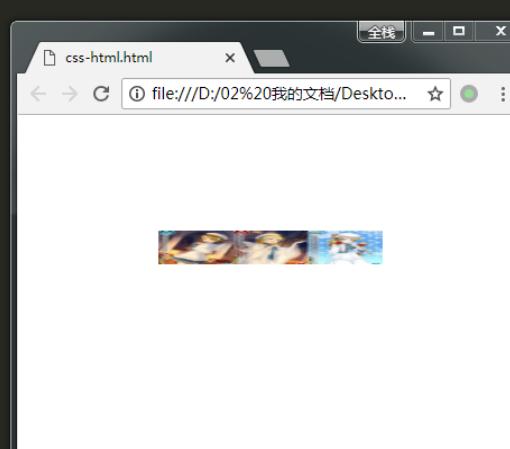
第 1 个%是红线从图片上
边向图片内部偏移的距离

第 2 个%是绿线右边向图
片内部偏移的距离

第 3 个%是紫线下边向
图片内部偏移的距离

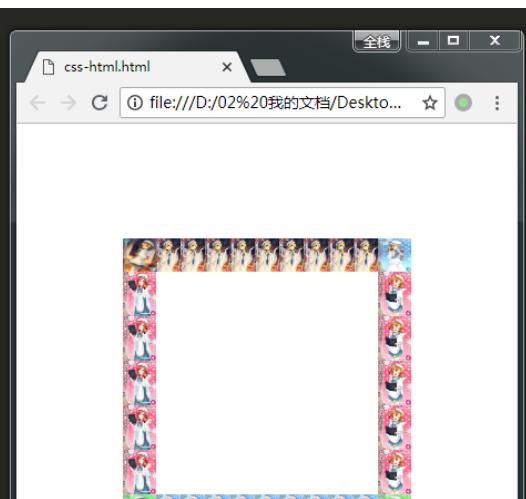
第 4 个%是蓝线左边向
图片内部偏移的距离

```
#test
{
    position: absolute;
    left: 0;
    right: 0;
    top: 0;
    bottom: 0;
    margin: auto;
    width: 200px;
    height: 200px;
    border: 30px solid;
    border-image-source: url(jgg.jpg);
    border-image-slice: 30% 0% 0% 0%;
}
```



第 1 个参数 30%把图片上边部分切出来了

```
#test
{
    position: absolute;
    left: 0;
    right: 0;
    top: 0;
    bottom: 0;
    margin: auto;
    width: 200px;
    height: 200px;
    border: 30px solid;
    border-image-source: url(jgg.jpg);
    border-image-slice: 33.333%;
    border-image-repeat: round;
}
```



让背景颜色变成图片

```
<style type="text/css">
  *{
    margin: 0;
    padding: 0;
  }
  #test
  {
    position: absolute;
    left: 0;
    right: 0;
    top: 0;
    bottom: 0;
    margin: auto;
    width: 200px;
    height: 200px;
    background-color: pink;
    background-image: url(jgg.jpg);
  }
</style>
</head>
<body>
  <div id="test">
  </div>
</body>
</html>
```

这就是给 div 框里面的元素加背景图片



其实是一大张图片，为什么只显示了其中一部分呢？这是因为 div 框大小被限制了

```

1  #test
2  {
3      position: absolute;
4      left: 0;
5      right: 0;
6      top: 0;
7      bottom: 0;
8      margin: auto;
9      /*width: 200px;
10     height: 200px;*/
11     background-color: pink;
12     background-image: url(jgg.jpg);
13  }
14  </style>
15 </head>
16 <body>
17     <div id="test">

```

这是把 div 限制区域大小去掉，背景图片就全部显示出来了

背景图片随着网页在滚动

```

1  }#test
2  {
3      position: absolute;
4      left: 0;
5      right: 0;
6      top: 0;
7      bottom: 0;
8      margin: auto;
9      width: 200px;
10     height: 200px;
11     background-color: pink;
12     background-image: url(jgg.jpg);
13  }
14  </style>
15 </head>
16 <body>
17     <div id="test">
18         <br/><br/>xzz<br/>xzz<br/>xzz<br/>xzz<br/>xzz<br/>xzz
19         <br/><br/>xzz<br/>xzz<br/>xzz<br/>xzz<br/>xzz<br/>xzz
20         <br/><br/>xzz<br/>xzz<br/>xzz<br/>xzz<br/>xzz<br/>xzz
21         <br/><br/>xzz<br/>xzz<br/>xzz<br/>xzz<br/>xzz<br/>xzz

```

你看除了背景图片随着网页滚动，字符也超出了 div 框

```

1  }#test
2  {
3      position: absolute;
4      left: 0;
5      right: 0;
6      top: 0;
7      bottom: 0;
8      margin: auto;
9      width: 200px;
10     height: 200px;
11     background-color: pink;
12     background-image: url(jgg.jpg);
13     overflow: auto;
14  }
15  </style>
16 </head>
17 <body>
18     <div id="test">
19         <br/><br/>xzz<br/>xzz<br/>xzz<br/>xzz<br/>xzz<br/>xzz
20         <br/><br/>xzz<br/>xzz<br/>xzz<br/>xzz<br/>xzz<br/>xzz
21         <br/><br/>xzz<br/>xzz<br/>xzz<br/>xzz<br/>xzz<br/>xzz
22         <br/><br/>xzz<br/>xzz<br/>xzz<br/>xzz<br/>xzz<br/>xzz

```

给 div 框加入 overflow 的 auto 属性，xzz 字符就在 div 框里面滚动

