

CSS3

作者: 王猛

Email : bietushiwo@gmail.com

微博: : [@王猛](#)

QQ : 672725440

说明: 本文档用于上课教案和学员复习, 可传播可分享, 如有错误, 请联系老王, 感谢矫正与探讨。

第1章 CSS3 概念

CSS 概念: 层叠样式表(英文全称: Cascading Style Sheets)是一种用来表现 [HTML](#) ([标准通用标记语言](#) 的一个应用) 或 [XML](#) (标准通用标记语言的一个子集) 等文件样式的计算机语言。CSS 不仅可以静态地修饰网页, 还可以配合各种脚本语言动态地对网页各元素进行格式化。

CSS3 是 CSS 的第三个版本, 于 1999 年开始制订, 2001 年 5 月 23 日 W3C 完成了 CSS3 的工作草案, 主要包括盒子模型、列表模块、[超链接](#)方式、语言模块、背景和边框、文字特效、多栏布局等模块, 但有些新功能浏览器厂商还处于懵逼状态, 暂时无法支持。

主要作用:

- (1) 增强控制页面的样式信息。
- (2) 运行样式信息与页面内容分离的一种标记语言。

第2章 CSS3 的引入 ***

2.1 css3 引入方式

1) 内联样式表, 行内引入

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title></title>
  </head>
  <body>
    <p style="color:white">哈哈</p>
    <i>猜谜语：两米男生，打一个吃的东西。</i>
  </body>
</html>

```

特点: 内联的样式表比其他的方法更加灵活 需要和展示的内容混合在一起 样式表的一些优点

2) 内嵌样式表，内部引入

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title></title>
    <style>
      i{
        color:green;
      }
    </style>
  </head>
  <body>
    <p style="color:white">哈哈</p>
    <i>猜谜语：两米男生，打一个吃的东西。</i>
  </body>
</html>

```

注意：一般内嵌样式表写在 head 标签中，其他地方不推荐。

3) 外部链接样式表，外部引入

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title>外部链接样式表</title>
    <!--
      rel 表示当前文档与连接文档的关系
      type 被连接文档的类型
      href 被连接文档的地址
      注意：css文件就是以.css结尾
    -->
    <link rel="stylesheet" type="text/css" href="./css.css"/>
  </head>
  <body>
    <b>是金子才会发光，你又不是。</b>
  </body>

```

```
</html>
```

CSS.CSS

```
b{  
    color:yellow;  
}
```

4) 导入式 (不推荐使用)

在一个css文件里，还可以引入其他的css

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="utf-8"/>  
    <title>外部链接样式表</title>  
    <link rel="stylesheet" type="text/css" href="./css.css"/>  
  </head>  
  <body>  
    <b>是金子才会发光，你又不是。</b>  
    <p>问题是，你是石头，不会发光。</p>  
  </body>  
</html>
```

CSS.CSS

```
/* @import "css2.css"; */  
@import url(css2.css);  
b{  
    color: red;  
}
```

CSS2.CSS

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

2.2 css3优先级

1.就近原则

2.理论上：行内>内嵌>链接>导入

3.实际上：内嵌、链接、导入在同一个文件头部，谁离相应的代码近，谁的优先级高。

第3章 CSS3 语法 ***

3.1 css3 基本语法

1) css 定义分别由 选择器 属性 属性值 组成。

格式: p{color:red;background:yellow}

但一般我们为了方便查看，会有如下写法：

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

2) css 大小写不敏感，推荐使用小写。

3) 属性和属性值之间用冒号分开 多个属性之间用分号隔开。

3.2 css3中的注释

html 注释 <!-- 注释内容 -->

css 注释： /* 写你要注释的内容 */

注意: 注释中不能再包含注释。

第4章 CSS3 选择器 ***

所谓的CSS选择器，其实就是如何选中HTML标签所提供的不同的方式，使用选择器，可以更加精准的实现程序员的需求。

4.1 基本选择器 ***

1) 类选择器

HTML 中使用 class="名称"

CSS 中使用 .名称{属性}

可以在 class 属性中通过空格分隔起多个名字

```
<!DOCTYPE html>  
<html>
```

```

<head>
  <meta charset="utf-8"/>
  <title>基本选择器</title>
  <style>
    .one{
      color:pink;
    }
  </style>
</head>
<body>
  <p class="one">卖家多赚钱，买家少花钱，问到底谁赚了？谁赔了？</p>
  <p class="one">世界本没有小三，都是女人跟女人过不去，男的一点事儿没有。</p>
  <p class="two one">北京不欢迎外地人。</p>
</body>
</html>

```

2) 唯一选择器

HTML 中使用 `id="名称"` 属性

CSS 中使用 `#名称{属性}`

当前 id 属性只能有一个名字，区分开 class，用于精准查找。

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title>基本选择器</title>
    <style>
      #gun{
        color:pink;
      }
    </style>
  </head>
  <body>
    <p id="gun">卖家多赚钱，买家少花钱，问到底谁赚了？谁赔了？</p>
    <p>世界本没有小三，都是女人跟女人过不去，男的一点事儿没有。</p>
    <p>北京不欢迎外地人。</p>
  </body>
</html>

```

3) 标签选择器

根据标签名字匹配所有标签

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title>基本选择器</title>
    <style>

```

```

        #gun{
            color: pink;
        }

        p{
            color: red;
        }

        b{
            color: yellow;
        }
    </style>
</head>
<body>
    <p id="gun">卖家多赚钱，买家少花钱，问到底谁赚了？谁赔了？</p>
    <p>世界本没有小三，都是女人跟女人过不去，男的一点事儿没有。</p>
    <p>北京不欢迎外地人。</p>
    <b>为啥都叫小姐姐呢？人家也不小啊</b>
</body>
</html>

```

4) 通用符选择器

* 表示匹配所有的标签

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8"/>
        <title>基本选择器</title>
        <style>
            *{
                color: green;
            }
        </style>
    </head>
    <body>
        <p id="gun">卖家多赚钱，买家少花钱，问到底谁赚了？谁赔了？</p>
        <p>世界本没有小三，都是女人跟女人过不去，男的一点事儿没有。</p>
        <p>北京不欢迎外地人。</p>
        <b>为啥都叫小姐姐呢？人家也不小啊</b>
    </body>
</html>

```

4.2 层级选择器 ***

1) 组合选择器

p,b 等多个选择器中间都用逗号隔开

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title></title>
    <style>
      /*组合选择器*/
      p,.er{
        color:red;
      }

      #one,.er{
        color:red
      }
    </style>
  </head>
  <body>
    <p id="one">11111111111111</p>
    <b class="er">22222222222222</b>
    <i>33333333333333</i>
  </body>
</html>

```

2) 包含选择器

p 空格 u 只要 u 标签被 p 标签包含

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title></title>
    <style>
      /*包含选择器*/
      p li{color:blue;}
    </style>
  </head>
  <body>

    <ul>
      <li>aaaaaaaa</li>
      <li>aaaaaaaa</li>
      <li>aaaaaaaa</li>
      <li>aaaaaaaa</li>
      <li>aaaaaaaa</li>
      <li>aaaaaaaa</li>
    </ul>

    <ol>
      <li>!!!!</li>
      <li>!!!!</li>
      <li>!!!!</li>
      <li>!!!!</li>
      <li>!!!!</li>
      <li>!!!!</li>
    </ol>
  </body>
</html>

```

```

        <li>!!!!</li>
    </ol>
</ul>

</body>
</html>

```

3) > 选择器

选择父元素 ul 的每一个 li 元素

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title></title>
    <style>

        ul>li{
            color:blue;
        }

    </style>
  </head>
  <body>

    <ul>
      <li>aaaaaaaa</li>
      <li>aaaaaaaa</li>
      <li>aaaaaaaa</li>
      <li>aaaaaaaa</li>
      <li>aaaaaaaa</li>
      <li>aaaaaaaa</li>
      <ol>
        <li>!!!!</li>
        <li>!!!!</li>
        <li>!!!!</li>
        <li>!!!!</li>
        <li>!!!!</li>
        <li>!!!!</li>
        <li>!!!!</li>
        <li>!!!!</li>
      </ol>
    </ul>

  </body>
</html>

```

4) + 选择器

紧贴着的兄弟元素(p 和 b 就是兄弟元素，而且只有一个是紧贴着的)。


```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title></title>
    <style>
      p+b{
        color:red;
      }
    </style>
  </head>
  <body>
    <b>我是大哥大</b>
    <p>zzzzzzzzzz</p>
    <p>zzzzzzzzzz</p>
    <p>zzzzzzzzzz</p>
    <b>111111111111111111111111111111</b>
    <p>zzzzzzzzzz</p>
    <p>zzzzzzzzzz</p>
    <p>zzzzzzzzzz</p>
    <p>zzzzzzzzzz</p>
    <i>bbbbbbbbbb</i>
    <b>aaaaaaaaaa</b>
    <b>aaaaaaaaaa</b>
    <b>aaaaaaaaaa</b>
    <b>aaaaaaaaaa</b>
    <b>aaaaaaaaaa</b>
    <b>aaaaaaaaaa</b>
    <b>aaaaaaaaaa</b>
    <b>aaaaaaaaaa</b>

  </body>
</html>

```

5) ~ 选择器

选择 p 元素之后的所有兄弟元素 b (p和b是同级关系)。

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title></title>
    <style>
      p~b{
        color:purple;
      }
    </style>
  </head>
  <body>

    <b>我是大哥大</b>
    <p>zzzzzzzzzz</p>
    <p>zzzzzzzzzz</p>

```

```

    <p>zzzzzzzzzz</p>
    <b>111111111111111111111111111111</b>
    <p>zzzzzzzzzz</p>
    <p>zzzzzzzzzz</p>
    <p>zzzzzzzzzz</p>
    <p>zzzzzzzzzz</p>
    <i>bbbbbbbbbb</i>
    <b>aaaaaaaaaa</b>
    <b>aaaaaaaaaa</b>
    <b>aaaaaaaaaa</b>
    <b>aaaaaaaaaa</b>
    <b>aaaaaaaaaa</b>
    <b>aaaaaaaaaa</b>
    <b>aaaaaaaaaa</b>
    <b>aaaaaaaaaa</b>

    </body>
</html>

```

4.3 属性选择器 * *

属性选择器就是根据属性名或者属性值来找到元素

E element 元素首字母 标签 选择器

ATT attribute 属性名

VAL value 值

1) E[ATT] 匹配所有具有 ATT 属性的 E 元素，不考虑属性值。

注意: 如果把元素E去掉表示匹配所有ATT属性的元素。

2) E[ATT=VAL] 匹配所有 ATT 属性等于 VAL 的 E 元素。

3) E[ATT~=VAL] 匹配所有 ATT 属性具有多个空格分隔的值其中一个值等于 VAL 的 E 元素。

~ 英文模式下 shift+反引号

4) E[ATT^=VAL] 匹配 ATT 属性值以 VAL 开头的 E 元素。

^ 英文模式下 shift+6

5) E[ATT\$=VAL] 匹配 ATT 属性值以 VAL 结尾的 E 元素。

\$ 英文模式下 shift+4

6) E[ATT*=VAL] 匹配 ATT 属性的值包含 VAL 字符的 E 元素。

* 英文模式下 shift+8

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>

```

```

</title></title>
<style>
    /*p[name]{color:red;} p标签有name属性就选中 */
    /*[name]{color:green;} 不管什么标签，有name属性就选中 */
    /*p[name='mi']{color:purple;} p标签，并且name等于mi的选中 */
    /*p[name~='ha']{color:red;} p标签，并且name属性中有多个空格分隔的值被选中
*/

    /*p[name^='q']{color:red;} p标签，并且name属性以p开头的选中 */
    /*p[name$='i']{color:gold;} p标签，并且name属性以p结尾的选中 */
    /*p[name*='i']{color:#ddddd;} p标签，并且name属性中含有i的选中 */
</style>
</head>
<body>
    <p name="mi">aaaaaaa</p>
    <p name="hahaha">bbbbbb</p>
    <p class="baomihua" name="ha hehe">cccccccccccccccc</p>
    <p name="qiji">ddddddddddd</p>
    <p aaa="yiyuan">eeeeeeeeeeeeeeee</p>
    <b name="xueguang">刚才我看见主管老师，看见他印堂发黑，弄不好今晚有"血光之灾",我有一良方可以破解，请使用白巾!!!</b>

    </body>
</html>

```

4.4 伪类选择器 *

伪类对元素进行分类是基于特征(characteristics)而不是它们的名字、属性或者内容；特征的分类一般叫伪类。

所谓伪类选择器，其实并不是对真正的元素进行选择，顾称为伪类选择器。

`E:first-line` E 元素的第一行内容

`E:first-letter` E 元素的第一个字母

`E:before` 在 E 内容之前

`E:before{content: 要添加的内容}`

`E:after` 在 E 内容之后

`E:after{content: 要添加的内容}`

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8"/>
        <title></title>
        <style>
            /*#one:first-line{
                font-size:30px;
            }
            #one:first-letter{

```

```

        font-size:50px;
        color:yellow;
    }*/
    /*#two:first-line{font-size:40px;}
    #two:first-letter{color:red;}*/
    #two:before{
        content: '小母牛';
        color:red;
    }
    #two:after{
        content: '->凉快';
        color:blue;
    }
}
</style>
</head>
<body>
    <div id="one">
        法海懂个球的爱情，瞎管闲事。<br/>
        唐僧喜欢的女人，女儿国国王
    </div>
    <div id="two">当年逆风尿三丈，如今顺风湿了鞋</div>

</body>
</html>

```

2) 结构性伪类选择器

a. `:root` 将样式绑定到页面的跟元素

所谓的根就是位于文档最顶部的结构元素，在 html 页面中最顶端就是 html，也就是改变整个html 的样式

b. `E:not()` 除了某个元素

对某个元素使用样式，但是想要排除这个元素的下的某个元素

c. `E:empty` 所有为空的E元素

d. `:target` 连接目标

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title></title>
    <style>
      /*:root{
        color:red;
      }*/
      /*li:not(#one){
        color:yellow;
      }*/
      /*li:empty{background-color:pink;}*/
      :target{
        font-size:40px;
        color:pink;
      }
    </style>
  </head>
  <body>
    <div id="one">
      法海懂个球的爱情，瞎管闲事。<br/>
      唐僧喜欢的女人，女儿国国王
    </div>
    <div id="two">当年逆风尿三丈，如今顺风湿了鞋</div>
  </body>
</html>

```

```

    </style>
</head>
<body>

    <ul>
        <li>饭岛爱</li>
        <li></li>
        <li>武藤兰</li>
        <li></li>
        <li id="one">波多野结衣</li>
        <li></li>
        <li>苍井空</li>
    </ul>
    <a href="#minmin">李老大</a><br/>
    <a href="#chaochao">刘老大</a><br/>
    <a name="minmin">敏敏</a>
    <a name="chaochao">超超</a>
</body>
</html>

```

4.5 其他选择器 *

1) `E:first-child` 对一个父元素中的第一个子元素 E 设置样式。

2) `E:last-child` 对一个父元素中的最后一个子元素 E 设置样式。

3) `E:nth-child()` 对指定需要的子元素E设置样式(正序)。

(数字) 获取某个一个内容

(odd) 所有正序下来的奇数个子元素

(even) 所有正序下来的偶数个子元素

4) `nth-last-child` 对指定需要的子元素E设置样式(倒序)。

(数字) 获取某个内容

(odd) 所有倒序下来的奇数个子元素

(even) 所有倒序下来的偶数个子元素

5) `nth-child(an+b)` 循环样式。

a 表示循环间隔的次数

b 表示循环开始的样式

6) `E:first-of-type()` 与 `first-child` 作用类似，但是他仅匹配使用的同种标签。

7) `E:last-of-type()` 与 `last-child` 作用类似，但是他仅匹配使用同种标签。

8) `E:nth-of-type()` 对指定需要的元素E设置样式(正序)。

(数字) 获取某个元素内容

(odd) 所有正序下来的奇数个元素

(even) 所有正序下来的偶数个元素

9) E:nth-last-of-type() 对指定需要的元素E设置样式(倒序)。

(数字)获取某个元素内容

(odd) 所有倒序下来的奇数个元素

(even) 所有倒序下来的偶数个元素

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title></title>
    <style>
      /*li:first-child{
        background-color:red;
      }*/
      /*li:last-child{
        background-color:green;
      }
      li:nth-child(10){
        background-color:pink;
      }*/
      /*奇数*/
      /*li:nth-child(odd){
        background-color:yellow;
      }*/
      /*偶数*/
      /*li:nth-child(even){
        background-color:red;
      }*/
      /*li:nth-last-child(4){
        background-color:green;
      }*/
      /*li:nth-last-child(odd){
        background-color:teal;
      }
      li:nth-last-child(even){
        background-color:#cccccc;
      }*/
      /*li:nth-child(3n+1){
        background-color:#43bcd4;
      }
      li:nth-child(3n+2){
        background-color:#ffff00;
      }
      li:nth-child(3n+3){
        background-color:#ff00ff;
      }*/

      /*li:first-child{//孩子
        background-color:red;
      }*/
      /*li:first-of-type{//兄弟
        background-color:blue;
```

```

    */

    /*li:last-child{
        background-color:blue;
    }*/
    /*li:last-of-type{
        background-color:yellow;
    }*/
    /*li:nth-of-type(4){
        background-color:yellow;
    }*/
    /*li:nth-of-type(odd){
        background-color:purple;
    }
    li:nth-of-type(even){
        background-color:teal;
    }*/
    /*li:nth-last-of-type(1){
        background-color:lightgreen;
    }*/
    /*li:nth-last-of-type(odd){
        background-color:pink;
    }
    li:nth-last-of-type(even){
        background-color:lightgreen;
    }*/
    */
</style>
</head>
<body>
<ul>
    <p>孩</p>
    <li>南风</li>
    <li>西风</li>
    <li>北风</li>
    <li>东风</li>
    <li>你电  你是光  你是闪电</li>
    <li>你电  你是光  你是闪电</li>
    <li>你电  你是光  你是闪电</li>
    <li>你电  你是光  你是闪电</li>
    <li>你电  你是光  你是闪电</li>
    <li>你电  你是光  你是闪电</li>
    <p>哈哈哈哈哈，我没词了</p>
</ul>
</body>
</html>

```

10) E:only-child 匹配父元素下仅有一个子元素并且这个子元素是E元素。

11) E:only-of-type() 匹配父元素下使用同中标签的为一个子元素。

12) E:enabled 可用状态的。

13) E:disabled 禁用状态的。

14) E:checked 匹配表单中被选中的radio或者checkbox元素。

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title></title>
    <style>
      /*p:only-child{
        background-color:red;
      }*/
      p:only-of-type{
        background-color:green;
      }

    </style>
  </head>
  <body>
    <div>
      <p>后面的朋友你们好吗? </p>
    </div>
    <div>
      <p>举起你们的双手? 跟我一起摇</p>
      <b>朋友说出你的故事。</b>
    </div>
    <div>
      <p>谎言的可怕</p>
      <p>下次你还信不信? </p>
    </div>
  </body>
</html>

```

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title></title>
    <style>
      input[type="text"]:enabled{background-color:green;}
      input[type="text"]:disabled{background-color:yellow; color:#000000;}
      input[type="radio"]:checked+b{font-size:30px;color:green;}
    </style>
  </head>
  <body>
    <form action="">
      <input type="text" value="readonly" readonly><br/>
      <input type="text" value="普通"><br/>
      <input type="text" value="disabled" disabled><br/>
      性别:
      <label><input type="radio" name="sex" value="0"><b>女</b></label>
      <label><input type="radio" name="sex" value="1"><b>男</b></label>
      <label><input type="radio" name="sex" value="2"><b>妖</b></label>
    </form>
  </body>
</html>

```

::selection 用鼠标选中的时候触发。

在不同的浏览器中，需要加浏览器私有的前缀，以识别到底是哪个浏览器在做测试。

Gecko内核 CSS前缀为 `-moz-`

火狐浏览器

WebKit内核 CSS前缀为 `-webkit-`

Comodo Dragon(科摩多龙)，苹果，搜狗高速浏览器3，快快浏览器，枫树浏览器，云游浏览器，360极速浏览器，世界之窗极速版，SRWare Iron，猎豹浏览器，RockMelt，QQ浏览器

Presto内核 CSS前缀为 `-o-`

Opera(欧朋),NDSBrowser

Trident内核 CSS前缀为 `-ms-`

IE，360极速浏览器，猎豹安全浏览器，傲游浏览器，百度浏览器，世界之窗浏览器，2345浏览器，腾讯TT，淘宝浏览器，采编读浏览器，搜狗高速浏览器，阿云浏览器，瑞星安全浏览器，Slim Browser，GreenBrowser、爱帆浏览器，115浏览器，155浏览器，114浏览器，N氧化碳浏览器，糖果浏览器，彩虹浏览器，瑞影浏览器，勇者无疆浏览器，闪游浏览器，蚂蚁浏览器，飞腾浏览器，速达浏览器，佐罗浏览器，海豚浏览器（iPhone/iPad/Android），UC浏览器

Trident内核 CSS前缀为 `-khtml-`

苹果浏览器之前的版本，后改为WebKit内核

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title></title>
    <style>
      ::-moz-selection{
        background-color:yellow;
        color:purple;
      }
      ::selection{
        background-color:green;
        color:red;
      }
    </style>
  </head>
  <body>
    <div>
```

上课不认真听课,原因有五个:

- 第一个: 手里有个手机
- 第二个: 心里有个姐弟
- 第三个: 旁边有个逗比
- 第四个: 智商不在服务区
- 第五个: 窗外有台挖掘机

```
    </div>
  </body>
</html>
```

a标签伪类

定义好的，不能随便改名，是为固定的某个元素在不同状态上的定义，目前a标签有几种不同状态。

:link 设置超链接a在未被访问前的样式

:visited 设置超链接a在其他链接地址被访问过的样式

:active 设置元素被用户激活(鼠标点击与释放之间)时的样式

:hover 设置元素在鼠标悬停时的样式

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title></title>
    <style>
      /*设置超链接a在未访问前的样式*/
      a:link{color:red;}
      /*设置超链接a在其他链接地址被访问过*/
      a:visited{color:green;}
      /*设置元素在鼠标悬停的时候的样式*/
      a:hover{color:pink;}
      /*设置元素被用户激活(数据点击与释放)*/
      a:active{color:purple;}

      /*文本标签只有下面两种内容*/
      /*设置元素在鼠标悬停的样式*/
      p:hover{color:pink;}
      /*设置元素被用户激活(鼠标点击与释放)*/
      p:active{color:purple;}

    </style>
  </head>
  <body>
    <a href="http://www.douyu.tv">斗鱼虎牙早期怎么火的? </a>
    <p>当然是看球啊，不看球难道看球啊? </p>
  </body>
</html>
```

第5章 CSS3常用属性 * * *

5.1 字体属性

`font-style` 设置字体样式

normal 正常字体

italic 斜体

`font-variant` 设置字体为小型的大写字母

normal 默认值

small-caps 将小写字母变为小型大写字母

`font-weight` 设置字体粗细

bold 粗体

normal 默认值

`font-size` 设置字体大小

常见单位:

px 像素 屏幕中一个点就是一个像素(相对单位)

em 相当于一个汉字的大小

`font-family` 设置字体类型

注意: 页面中该属性设置的字体必须存在于客户端电脑中有的字体才可以使用

`font` 字体样式统一这个标签

但是要注意: 使用该属性时, 值具有顺序关系必须按照顺序来写, 不存在可以省略, 其中 `font-size` 和 `font-family` 必须写。

font统一样式详解

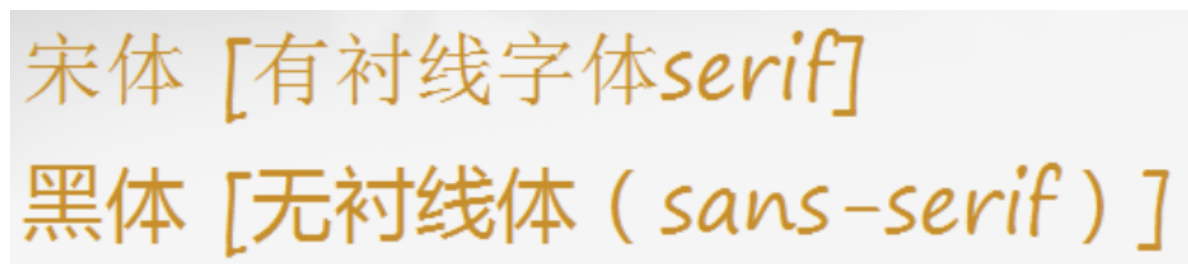
font属性顺序: [斜体 小型字体 粗体] <字体大小[/行高]> <类型>

使用font统一设置上面所有属性的时候注意顺序是: `font-style` `font-variant` `font-weight` `font-size` `line-height` `font-family`

中文字体的2大分类:

无衬线: 黑体, 雅黑, 文泉驿正黑 sans-serif

有衬线: 宋体, 新宋, serif (像衬衫一样有棱角)



英文字体样式很大

等宽字体 (Monaco, Courier, Monospace)

不等宽字体 (KodchiangUPC)



我们编程时，如果用不等宽字体，很可能给我们的对齐带来麻烦。

因为良好的缩进和对齐，非常的便于我们调试错误，发现错误。

编程时我们用的都是程序员专用的等宽字体。

安全字体：

```
font-family: '微软雅黑';
```

别人网站设置的字体是微软雅黑，我们打开显示也是微软雅黑字体。

那是因为我们的计算机上有微软雅黑这个字体，如果是一个国外的用户打开之后，他的机器上并没有微软雅黑这种字体，会报错吗？也就是，我们设置的字体，客户(浏览器)机器上未必有。

但是也不会出错，电脑会尽量在自己本地有的字体中找个相似的字体显示出来。

网页设计者如果用比较稀奇古怪的字体，用户机器上没有，字体效果也显示不出来。

所以要注意，字体的常用和不常用，也就是安全字体。

什么是安全字体？安全字体就是几乎大家都有的字体

字体顺序：

win7的系统'微软雅黑'

xp是'黑体'

linux是'文泉驿正黑体'

```
'font-family: '静蕾体','微软雅黑','文泉驿正黑体','黑体',sans-serif;
```

字体顺序: 独特字体,安全字体,字体大类;

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title></title>
    <style>
      #one{
        /*font-size:40px;*/
        font-size:1.5em;
        font-style:italic;
        font-family: 黑体, 宋体;
        font-weight:normal;
        font-variant:small-caps;
```

```

    }
    #two{
        font:italic bold 50px/20px 微软雅黑,宋体,黑体;
    }
</style>
</head>
<body>
    <div id="one">Small文思三千，不如胸脯四两，才高八斗，不如胯下半斤，人大北大,不如波大</div>
    <div>文思三千，不如胸脯四两,才高八斗，不如胯下半斤，人大北大,不如波大</div>

    <div id="two">
        One car come one car go , two car
        pengpeng people die;
        一辆车来，一辆车走，两辆车碰碰,人死了
    </div>
</body>
</html>

```

5.2 文本属性

text-indent 首行缩进

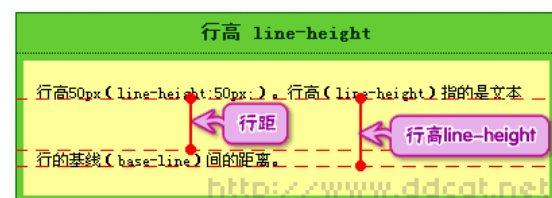
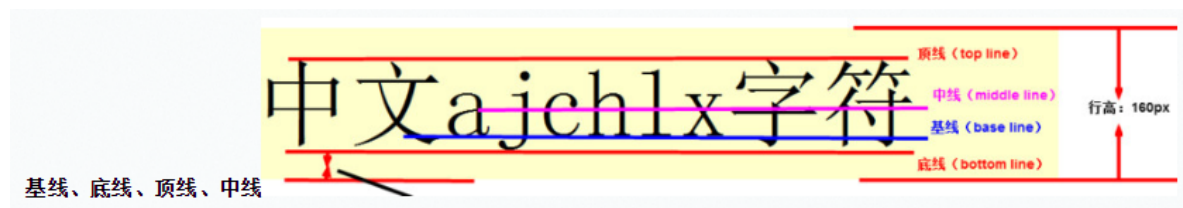
常见单位: em 一个汉子的大小

line-height 用于设置当前文本的行高

常见单位:

px 像素 屏幕中一个点就是一个像素(相对单位)

em 相当于一个汉字的大小



行高 (line-height): 包括内容区与以内容区为基础对称拓展的空白区域, 我们称之为行高。一般情况下, 也可以认为是相邻文本行基线间的距离。

text-align 设置元素内容的水平对齐方式

取值: left center right

text-decoration 设置文本修饰

取值:

none 不修饰

underline 下划线

line-through 删除线

word-spacing 设置单词之间的距离（对于中文需要加上每个字加上空格才有效果）取值可以是负数

letter-spacing 设置字母之间的间距 取值可以是负数

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title></title>
    <style>
      #one{text-indent:2em;}
      ul>li{
        /*边框属性*/
        border:1px solid red;
        line-height:40px;
        text-align:right;
      }
      a{
        text-decoration:none;
        /*text-decoration:underline;*/
        /*text-decoration:line-through;*/
      }
      div{
        word-spacing:100px;
      }
      b{
        letter-spacing:-100px;
      }
    </style>
  </head>
  <body>
    <p id="one">
      和谐社会，就是不和谐的地方你看不到。
    </p>
    <p>
      和谐社会，就是不和谐的地方你看不到。
    </p>
    <ul>
      <li>最有魅力的代表人物</li>
      <li>康师傅</li>
      <li>因为每天都有人泡他</li>
    </ul>
    <a href="http://www.baidu.com">百度一下你就知道</a>

    <div>love zai li ming po small money 爱在黎明破晓前</div>
    <b>this is a island 这是一个岛</b>
  </body>
</html>
```

5.3 背景属性

`background-color` 设置背景颜色

可以使用颜色的表达方式: 英文/六位的16进制/rgb()

在css中六位十六进制如果六位每两位都是相同的数字或者字母 我们可以将他们使用为三位数字或者字母代替: 例如: # ffff00 ----> #ff0

rgb(255,255,0)

rgb 表达式中: r 0-255 g 0-255 b 0-255 中间用逗号隔开

rgb(255,255,255)

transparent 透明色, 背景透明, 默认就是透明, 写不写都一样

`background-image` 设置背景图片

background-image:url(图片地址)

url('text.jpg')

url("text.jpg")

url(test.jpg)

url(https://www.baidu.com/img/bd_logo1.png)

`background-repeat` 设置背景重复

repeat-x 水平重复

repeat-y 垂直重复

repeat 默认值 水平方向和垂直方向重复

no-repeat 不重复

`background-position` 设置背景位置

格式1: background-position:水平位置和垂直位置

格式2: background-position:水平位置和垂直位置

英文单词: left,center,right/top,center,bottom

长度单位: 只要长度单位都可以 px em 常用像素

`background-attachment` 设置背景是否滚动

默认值: scroll 背景图片随着滚动条滚动

fixed 绑定页面 图片不随着页面而滚动

`background` 简写属性 推荐使用 没有顺序关系

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title></title>
    <style>
      #one{
```

```

        width:800px;
        height:10000px;
        /*background-color:red;*/
        /*background-color:#ffff00;*/
        /*background-color:#ff0;*/
        background-color:#abcdef;
        /*background-color:rgb(44,109,46);*/
        background-image:url(1.gif);
        background-repeat:no-repeat;
        /*background-position:center top;*/
        /*background-position:center;*/
        background-position:100px 300px;
        background-attachment:fixed;
    }
</style>
</head>
<body>
    <div id="one"></div>
</body>
</html>

```

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8"/>
        <title></title>
        <style>
            #san{
                width:500px;
                height:500px;
                background: transparent no-repeat url('./1.gif') 150px 200px
fixed;

                /*background-color:red;*/
            }
        </style>

    </head>
    <body>
        <div id="san">
            身高不够用鞋补
            身高不够用鞋补
            身高不够用鞋补
            身高不够用鞋补
            身高不够用鞋补
            身高不够用鞋补
            身高不够用鞋补
            身高不够用鞋补
            身高不够用鞋补
            身高不够用鞋补
            身高不够用鞋补
            身高不够用鞋补
            身高不够用鞋补
            身高不够用鞋补
        </div>
    </body>
</html>

```


5.4 尺寸属性

`width` 设置宽度

`height` 设置高度

`min-height` 最小高度

`max-height` 最大高度

备注：

最小的高度属性，当前高度如果没有达到最小高度那么高度将设置为最小高度

最大高度属性，允许当前元素根据内容自动减少高度，但是如果达到最大高度值，将不在添加高度

说人话：

假如我同时设置了宽100和高100，那你这个div就按照宽高给你设置，但如果我设置了最大高度200，那高度就会根据你的内容自动延伸高度，直到你设置的最大高度拉倒。注意：设置高了，就不要设置最大高度，无效。

最小高度同理，假如我设置宽100，高100，那这个div就这么大，100*100，但如果设置了最小高度50，那就根据你的内容无限的拉长，但随着内容的缩小，小到你的最小高度50就拉倒了。

使用情况：

一般在实际的开发中，只设置设计的宽和高，不固定的自动拉伸不怎么使用，因为会导致布局不明确，从而影响实际的网页效果。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title></title>
    <style>
      #one{
        width:100px;
        min-height:100px;
        border:1px solid red;
      }
    </style>
  </head>
  <body>
    <div id="one">
      我是内容
    </div>
  </body>
</html>
```

5.5 列表属性

列表属性用于设置 `ul` 和 `ol` 的列表样式

`list-style-type`: 设置列表的符号样式(固定样式 必须在手册中出现的才能使用, 参考手册)

`list-style-image`: 使用指定的图片来代替列表的序号

注意: 如果同时存在image和type 以用户定义的image为准

格式: `list-style-image:url('图片地址')`

`list-style-position`: 设置列表符号需要的位置

取值:

inside 需要在内容中

outside 需要在内容外

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title></title>
    <style>
      ul>li{
        /*list-style-type:none;*/
        list-style-type:decimal;
      }
      ol>li{
        list-style-type:disc;
        list-style-image:url(1.gif);
        border:1px solid red;
        /*list-style-position:inside;*/
        list-style-position:outside;
      }
    </style>
  </head>
  <body>
    <h2>无序列表</h2>
    <ul>
      <li>做人要学孟获</li>
      <li>七擒七纵</li>
      <li>不要学赵云</li>
      <li>七进七出</li>
    </ul>

    <h2>有序列表</h2>
    <ol>
      <li>锄禾日当午</li>
      <li>汗滴禾下土</li>
      <li>谁知盘中餐</li>
      <li>粒粒皆辛苦</li>
    </ol>
  </body>
</html>
```

5.6 字体颜色补充

`color` 设置字体颜色

取值: 英文单词 16进制 rgb()

```
<!DOCTYPE html>
<html>

  <head>
    <meta charset="utf-8"/>
    <title></title>
    <style>
      p{
        font-size:40px;
        color:rgb(255,0,255);
        background-color:blue;
      }
      #one{
        width:200px;
        background-color:pink;
        max-height:200px;
        color:#00ffff;
      }
    </style>
  </head>
  <body>
    <p>
      生活，哪有不累的，你说的是瑞典吧。
    </p>
    <div id="one">
      再努努力，你就坐在星巴克喝咖啡了，别忘了就着大蒜，煎饼果子也可以来一套。
      再努努力，你就坐在星巴克喝咖啡了，别忘了就着大蒜，煎饼果子也可以来一套。
      再努努力，你就坐在星巴克喝咖啡了，别忘了就着大蒜，煎饼果子也可以来一套。
      再努努力，你就坐在星巴克喝咖啡了，别忘了就着大蒜，煎饼果子也可以来一套。
      再努努力，你就坐在星巴克喝咖啡了，别忘了就着大蒜，煎饼果子也可以来一套。
      再努努力，你就坐在星巴克喝咖啡了，别忘了就着大蒜，煎饼果子也可以来一套。
      再努努力，你就坐在星巴克喝咖啡了，别忘了就着大蒜，煎饼果子也可以来一套。
    </div>
  </body>
</html>
```

5.7 定位属性

1) `position` 设置元素的定位方式

`static` 默认值。没有定位

`absolute` 绝对定位

`relative` 相对定位

`fixed` 定位到窗口

详解：

absolute 绝对定位，以其他元素作为参照物移动指定距离的定位方式，默认是（body，也就是整个文档的开头）

注意: 绝对定位以后，该元素不会占据原来的物理位置

如果只使用 `position:absolute` 没有任何意义，只是脱离文档流，需要配合方位使用

`left`: 设置定位元素的水平移动位置(距离左边的距离)

`top`: 设置定位元素的垂直移动位置(距离顶部的距离)

`right`: 设置定位元素的水平移动位置(距离右边的距离)

`bottom`: 设置定位元素的垂直移动位置(距离底部的距离)

关于绝对定位：

1. 如果该元素的外元素没有任何的一个元素采用position定位 那么此时的定位的参照元素是body或者说页面的参照点0,0
2. 如果元素的外层为非static(有了除默认值以外的定位位置)那么这个外层元素的成为该元素的参照点 这个外层元素的左上角才是我们的0,0
3. 如果元素的外层元素没有设置任何的position的值 那么该元素将寻找距离自己最近的其他设置过position的元素作为参照物(有个要求:必须是嵌套关系)

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title></title>
    <style>
      /*现在你不清楚他是干什么的但是你记住他是用来清除小风凤的可以*/
      *{margin:0px;padding:0px;}
      div{width:100px;height:100px;}
      /*#one{

          background:red;
          position:absolute;
          left:100px;
          top:100px;
        }
      #two{background:green;}
      #three{
          background:blue;
          position:absolute;
          left:100px;
          top:100px;
        }
      #si{background:yellow;}*/
    /*

      #die{
        width:200px;height:200px;
        background:lightblue;
        position:absolute;
        left:100px;
        top:100px;
      }
    */
  </head>
</html>
```

```

        #son{
            width:100px;height:100px;
            background:red;
            position:absolute;
            left:100px;
            top:100px;
        }*/

#yeye{
    width:400px;
    height:400px;
    background:yellowgreen;
    position:absolute;
    left:100px;
    top:100px;
}

#daidi{
    width:200px;height:200px;
    background:lightblue;
    position:absolute;
    left:200px;
    top:200px;
}

#son{
    width:100px;height:100px;
    background:yellow;
    position:absolute;
    top:200px;
    left:200px;
}

</style>
</head>
<body>
    <!-- <div id="one"></div>
    <div id="two"></div>
    <div id="three"></div>
    <div id="si"></div> -->
    <div id="yeye">
        <div id="daidi">
            <div id="son"></div>
        </div>
    </div>
</body>
</html>

```

2) relative 相对定位

相对定位，以当前元素作为参照移动指定的距离的定位方式，该元素在什么位置那么他就是以自己的左上角为0,0

注意: 该元素依旧占据原来的位置，也就是你当初占了多大，给你留着，不管你飘到哪里，我都等你。

fixed 固定位置

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title></title>
    <style>
      *{margin:0px;padding:0px;}
      div{width:100px;height:100px;}
      #one{
        background: pink;
      }
      #two{
        background: lightblue;
      }
      #three{
        background: lightyellow;
        position: relative;
        top: 100px;
        left: 100px;
      }
      #four{
        background: lightgreen;
      }
    </style>
  </head>
  <body>
    <div id="one"></div>
    <div id="two"></div>
    <div id="three"></div>
    <div id="four"></div>
  </body>
</html>
```

3) fixed 固定定位，相对于窗口的定位

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Document</title>
    <style>
      #one{
        width:100px;
        height:100px;
        border:1px solid red;
        position:fixed;
        bottom:50px;right:50px;
      }
      #one>a{
        text-decoration:none;
      }
    </style>
  </head>
```

[illegible]

导航案例

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style type="text/css">
    *{margin:0px;padding:0px;}
    #nav{
      height: 40px;
      width: 100%;
      background:#000;
      position: fixed;
    }
    #nav>ul>li{
      width: 120px;
      height: 40px;
      background: red;
      text-align: center;
      list-style-type: none;
      line-height: 40px;
    }

    #nav>ul>li>a{
      color: #fff;
      text-decoration: none;
    }
    #nav>ul{
      position: absolute;
      left: 100px;
```

```

}
#nav>ul>li:nth-child(2){
    position:absolute;
    left:120px;
    top:0px;
}
#nav>ul>li:nth-child(3){
    position:absolute;
    left:240px;
    top:0px;
}
#nav>ul>li:nth-child(4){
    position:absolute;
    left:360px;
    top:0px;
}
#nav>ul>li:nth-child(5){
    position:absolute;
    left:480px;
    top:0px;
}
#nav>ul>li:nth-child(6){
    position:absolute;
    left:600px;
    top:0px;
}
#nav>ul>li>a:hover{
    color:yellow;
}
#nav>ul>li:hover{
    background:green;
}
#content{
    height: 1500px;
    background: pink;
    width: 1200px;
    margin: 0px auto;
}
</style>
</head>
<body>
<div id="nav">
<ul>
<li><a href="">网站首页</a></li>
<li><a href="">发票专区</a></li>
<li><a href="">刻章专区</a></li>
<li><a href="">二手专区</a></li>
<li><a href="">耍贱专区</a></li>
<li><a href="">老王写真</a></li>
</ul>
</div>
<div id="content">
<br />
<br />
<br />
<br />
<br />

```



```
    </div>
</div>
</body>
</html>
```

5.8 显示层级属性

z-index 设定定位元素z轴的距离(就是元素的显示顺序)

注意：必须配合绝对定位使用。

取值

auto 自动

number 数字

数字越大就会在最上面显示

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title></title>
    <style>
      #one{
        background:pink;
        width:300px;
        height:300px;
        position:absolute;
        z-index:auto;
      }
      #two{
        background:lightgreen;
        width:200px;
        height:200px;
        position:absolute;
        z-index:auto;
      }
    </style>
  </head>
  <body>
    <div id="one"></div>
    <div id="two"></div>

  </body>
</html>
```

第6章 布局属性 ***

6.1 块状元素和行内元素

1) `<div></div>` 无意义的块状元素

什么是块级元素：具有宽度高度属性并且独占一行

块级元素的特点：

1. 独占一行
2. 宽度和浏览器宽度一样，与内容无关
3. 宽度和高度都可以调整
4. 可以容纳行内元素和其他的块元素

2) `` 无意义的行内元素

什么是行内元素：没有宽度高度属性并且不独占一行

行内元素的特点：

1. 和其他标签在同一行
2. 宽度与内容有关 内容改变宽度也改变
3. 宽度和高度不能调整
4. 容纳文本和其他的行内元素

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title></title>
    <style>
      div{
        height:100px;
        width:100px;
        background:pink;
      }
      span{
        height:100px;
        width:100px;
        background:lightgreen;
      }
      p{
        background:lightyellow;
        width:100px;
        height:100px;
      }
    </style>
  </head>
  <body>
    <div>块级标签</div>
```



```

        #one{
            width:100px;
            height:100px;
            background:pink;
        }
        #two{
            width:50px;
            height:50px;
            background:lightgreen;
            display:none;
        }
        #one:hover>#two{
            display:block;
        }

    </style>
</head>
<body>
    <!-- <div>法律是什么？两拨人打架，搞不定了。叫法律来帮忙</div>
    <span>
        法律搞不定怎么办？警察局就诞生了。
    </span> -->
    <div id="one">
        <p>显示</p>
        <div id="two">背景</div>
    </div>
</body>
</html>

```

6.2 显示与隐藏属性

visibility 设置元素是否显示

取值:

visible 显示(默认值)

hidden(隐藏)

注意: **visibility** 和 **display:none** 的区别, **visibility** 的隐藏方式仅隐藏内容的显示, 其占用的空间依旧占用。而 **display** 的隐藏方式是彻底隐藏该元素的内容和位置。

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8"/>
        <title></title>
        <style>
            .one{
                width:300px;
                height:250px;
                background:#ccc;
                text-align:center;
                border:1px solid #fff;
                display:inline-block;
            }
        </style>
    </head>
    <body>
        <div class="one">
            显示
        </div>
    </body>
</html>

```

```

    }
    .two{
        display:none;
        /* visibility:hidden; */
    }
</style>
</head>
<body>
<div class="one">
    <br/>
    <span>商品名称:社会我强哥人狠话不多1</span><br/>
    <span>商品价格:日元9.8</span><br/>
    <span>商品描述:真人发声,真实触感,用过都说好!! </span>
</div>
<div class="one">
    <br/>
    <span>商品名称:社会我强哥人狠话不多2</span><br/>
    <span>商品价格:日元9.8</span><br/>
    <span>商品描述:真人发声,真实触感,用过都说好!! </span>
</div>
<div class="one two">
    <br/>
    <span>商品名称:社会我强哥人狠话不多3</span><br/>
    <span>商品价格:日元9.8</span><br/>
    <span>商品描述:真人发声,真实触感,用过都说好!! </span>
</div>
<div class="one">
    <br/>
    <span>商品名称:社会我强哥人狠话不多4</span><br/>
    <span>商品价格:日元9.8</span><br/>
    <span>商品描述:真人发声,真实触感,用过都说好!! </span>
</div>
<div class="one">
    <br/>
    <span>商品名称:社会我强哥人狠话不多5</span><br/>
    <span>商品价格:日元9.8</span><br/>
    <span>商品描述:真人发声,真实触感,用过都说好!! </span>
</div>
<div class="one">
    <br/>
    <span>商品名称:社会我强哥人狠话不多6</span><br/>
    <span>商品价格:日元9.8</span><br/>
    <span>商品描述:真人发声,真实触感,用过都说好!! </span>
</div>
<div class="one">
    <br/>
    <span>商品名称:社会我强哥人狠话不多7</span><br/>
    <span>商品价格:日元9.8</span><br/>
    <span>商品描述:真人发声,真实触感,用过都说好!! </span>
</div>
</body>
</html>

```

6.3 设置内容超出部分

overflow 设置内容元素超出部分

取值：

visible 始终可见(默认值)

hidden 超出部分隐藏处理

scroll 滚动条，超出内容有滚动条显示

auto 自动，如果超出内容他显示滚动条，如果没有超出不显示滚动条

注意: 中文会自动换行，英文单词也会自动换行，但是如果是一串英文，没有空格的情况下，是不会自动换行，数字也是相同的。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title></title>
    <style>
      div{
        width:200px;
        height:200px;
        background:yellow;
        /*默认值*/
        /*overflow:visible;*/
        /*超出部分隐藏*/
        /*overflow:hidden;*/
        /*超出部分以滚动条方式显示*/
        /*overflow:scroll;*/
        /* overflow:auto; */
      }
    </style>
  </head>
  <body>
    <div>
      aaaaaaa aaaaaaaaaa aaaaaaaaaaaa aaaaaaa
      1111111111 111111111111111111
      常见的不会恋爱。
      在吗？在
      在干吗？没干嘛
      吃饭了吗？吃了
      哦。。。。
      蛐蛐叫。。。。

      你怎么还在线？ 嗯。。。
      又这么晚睡觉？ 嗯。。习惯了，你不也没睡吗？
      你不睡我能睡嘛？ 。。。。
      去照照镜子看看脸。 咋了？？
      黑眼圈是不是又大了？
      跟你说了这么晚睡觉容易黑眼圈，就是不听。
      那只能跟你说怎么样。。。。

    </div>
  </body>
</html>
```

6.4 float 浮动

float 设置浮动，脱离普通文档流

取值：

left 左浮动

right 右浮动

注意:

1. 如果父级没有给高度，那么他就会变成一条线，因为里面没有任何内容所有内容全部在浮动层面，不在普通文档流层面
2. 如果你使用浮动，请将你的元素计算精确，如果不精确会有弊端。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title></title>
    <style>
      div{
        width:100px;
        height:100px;
      }
      #one{
        background:yellow;
        float:left;
      }
      #two{
        background:yellowgreen;
        float:left;
      }
      #three{
        background:greenyellow;
        float:left;
      }
      #four{
        background:lightgreen;
        float:right;
      }
    </style>
  </head>
  <body>
    <div id="one"></div>
    <div id="two"></div>
    <div id="three"></div>
    <div id="four"></div>
  </body>
</html>
```

简单布局

```
<!DOCTYPE html>
```

[illegible]


```

    </div> -->

    <div id="main">
        <div id="left"></div>
        <div id="right">
            <div class="one"></div>
            <div class="one"></div>
            <div class="one"></div>
            <div class="one"></div>
            <div class="one"></div>
            <div class="one"></div>
            <div class="one"></div>
            <div class="one"></div>
        </div>
    </div>
</body>
</html>

```

clear 清除浮动，让你脱离文档流的内容(float) 回到文档流中

取值：

left 清除左边的浮动

right 清除右边的浮动

both 清除两端浮动，一般使用这个

说人话：

所谓清除浮动，其实非常容易理解，就是我不允许有人在我头上飘着。

推荐:

每次看见浮动都记得清除浮动什么地方出现就在什么地方清除(同一级别的清除一次)，但是不能说统一一次全部清除，只要出现浮动就要清除浮动，如果不清除浮动后期页面会非常乱，你会找不到哪些元素被压在低下了，哪些应该出来。

一定记得要清除浮动，一定记得要清除浮动，一定记得要清除浮动。

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
    <style type="text/css">
        #div1{
            height: 100px;
            width: 100px;
            float: left;
            background: red;
        }
        #div2{
            height: 100px;
            width: 100px;
            float: left;
            background: blue;
        }
    </style>

```

```

#div3{
    height: 100px;
    width: 100px;
    float: left;
    background: green;
}
#div4{
    height: 100px;
    width: 100px;
    float: left;
    background: yellow;
}
#div5{
    height: 100px;
    width: 100px;
    background: black;
    float: left;
}
#div99{
    height: 40px;
    width: 100%;
    background: red;
    clear: both;
}
#div99>ul>li{
    float: left;
}
</style>
</head>
<body>
    <div id="div0">
        <div id="div1"></div>
        <div id="div2"></div>
        <div id="div3"></div>
        <div id="div4"></div>
        <div id="div5"></div>
    </div>
    <div id="div0">
        <div id="div99">
            <ul>
                <li>1</li>
                <li>2</li>
                <li>3</li>
                <li>4</li>
            </ul>
        </div>
        <div id="div1"></div>
        <div id="div2"></div>
        <div id="div3"></div>
        <div id="div4"></div>
        <div id="div5"></div>
    </div>
</body>
</html>

```

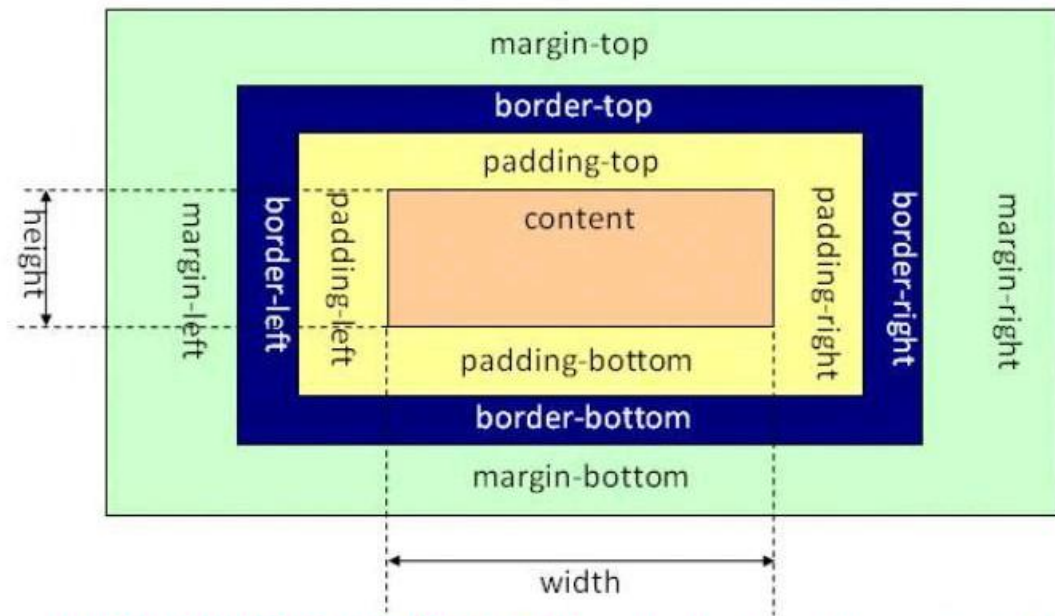
浮动做导航

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title></title>
    <style>
      /*初始化所有样式*/
      *{margin:0px;padding:0px;}
      /*给导航最外面加上高度*/
      #nav{
        width:1100px;
        height:40px;
        /*border:1px solid red;*/
      }
      /*让导航横着显示去掉点*/
      #nav ul li{
        float:left;
        list-style-type:none;
      }
      /*调整a标签的细节*/
      #nav ul li a{
        width:100px;
        height:40px;
        /*border:1px solid green;*/
        display:block;
        text-align:center;
        font:15px/40px 微软雅黑;
        text-decoration:none;
        color:#333333;
      }
      /*鼠标悬停的颜色*/
      #nav ul li a:hover{
        color:#c81623;
      }
    </style>
  </head>
  <body>
    <div id="nav">
      <ul>
        <li><a href="">秒杀</a></li>
        <li><a href="">优惠券</a></li>
        <li><a href="">PLUS会员</a></li>
        <li><a href="">闪购</a></li>
        <li><a href="">拍卖</a></li>
        <li><a href="">京东服饰</a></li>
        <li><a href="">京东超市</a></li>
        <li><a href="">生鲜</a></li>
        <li><a href="">全球购</a></li>
        <li><a href="">京东金融</a></li>
        <div style="clear:both"></div>
      </ul>
    </div>
  </body>
</html>
```

6.5 盒子模型

6.5.1 盒子模型图

■ 标准盒子模型



6.5.2 margin外边距

1) `margin` 外边距，设置盒子模型的外部距离。

`margin-top`: 设置当前元素和顶部元素之间的距离

`margin-left`: 设置当前元素和左部元素之间的距离

`margin-right`: 设置当前元素和右部元素之间的距离

`margin-bottom`: 设置当前元素和底部之间元素的距离

简写属性：

1. 分别设置4个边的间距(1个参数)

`margin`: 同时设置上下左右

2. 分别设置4个边的间距(2个参数)

`margin`: 上下/左右

3. 分别设置4个边的间距(3个参数)

`margin`: 上边/左右/下边

4. 分别设置4个边的间距(4个参数)

`margin`: 上/右/下/左

记住: `margin:0px auto` 可以让块状元素水平居中

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    /*
    #div0{
      border:1px solid green;
      width:200px;
      height:100px;
      margin-top:10px;
      margin-right:20px;
      margin-bottom:30px;
      margin-left:50px;
      margin:10px 20px 30px 50px;

    }

    */
    #div1{
      width:800px;
      height:400px;
      border:1px solid red;
      margin:0px auto;
    }
    #div2{
      width:800px;
      height:400px;
      border:1px solid red;
      margin: 30px;
    }

  </style>
</head>
<body>
  <div id="div0"></div>
  <div id="div1"></div>
  <div id="div2"></div>
</body>
</html>
```

6.5.3 padding 内边距

2) padding: 设置元素边框与内容之间的距离

`padding-top`: 设定当前元素的内容和顶部的边框之间距离

`padding-left`: 设定当前元素的内容和左边的边框之间距离

`padding-bottom`: 设定当前元素的内容和底部的边框之间距离

`padding-right`: 设定当前元素的内容和右边的边框之间距离

简写属性：

1. 分别设置4个边的间距(1个参数)

`padding`: 同时设置上下左右

2. 分别设置4个边的间距(2个参数)

`paddingt`: 上下/左右

3. 分别设置4个边的间距(3个参数)

`paddingt`: 上边/左右/下边

4. 分别设置4个边的间距(4个参数)

`paddingt`: 上/右/下/左

一般设置padding和margin，都不要同时设置4个参数，会破坏盒子模型理想的效果。

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style type="text/css">
    #div0{
      width: 200px;
      height: 200px;
      border: 1px solid red;
    }
    #div0_1{
      width: 100px;
      height: 100px;
      border: 1px solid blue;
      margin: 50px auto;
      padding-top: 20px;
      padding-right: 20px;
      padding-bottom: 20px;
      padding-left: 20px;
      padding: 30px;
    }
  </style>
</head>
<body>
  <div id="div0">
    <div id="div0_1">盒子模型</div>
  </div>
</body>
</html>
```

6.5.4 border边框

3) `border` 设置盒子模型边框的属性

`border-color`: 同时设定4个边的边框颜色

`border-width`: 同时设定4个边的边框宽度

`border-style`: 同时设定4个边的边框样式

`border-top` 单独设定顶部边框的边框颜色样式宽度

`border-top-width` 单独设定顶部边框的边框宽度

`border-top-color` 单独设定顶部边框的边框颜色

`border-top-style` 单独设定顶部边框的边框样式

`border-left` 单独设定左部边框的边框颜色样式宽度

`border-left-width` 单独设定左部边框的边框宽度

`border-left-color` 单独设定左部边框的边框颜色

`border-left-style` 单独设定左部边框的边框样式

`border-bottom` 单独设定底部边框的边框颜色样式

`border-bottom-width` 单独设定底部边框的边框宽度

`border-bottom-color` 单独设定底部边框的边框颜色

`border-bottom-style` 单独设定底部边框的边框样式

`border-right` 单独设定右部边框的边框颜色样式宽度

`border-right-width` 单独设定右部边框的边框宽度

`border-right-color` 单独设定右部边框的边框颜色

`border-right-style` 单独设定右部边框的边框样式

边框样式的取值:

`dotted` 点线

`dashed` 虚线

`solid` 实线

`double` 双实线

`groove` 槽状线

`ridge` 脊线

`inset` 内嵌效果

`outset` 外凸效果

border-image 边框图片

border-radius 圆弧

1. 一个值代表四个角
2. 二个值代表 左上右下 右上左下
3. 三个值 左上 右上左下 右下
4. 四个值 左上 右上 右下 左下

border 简写： `border:1px solid red` 同时设置4个边线的厚度，样式，颜色。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title></title>
    <style>
      div{
        width:500px;
        height:500px;
        /*
          border-color:gold;
          border-width:10px;
          border-style:dotted;
          border-top-color:yellow;
          border-top-style:solid;
          border-top-width:20px;
          */

        /*
          border-top:100px groove red;
          border-right: 50px groove red;
          border-bottom:30px groove red;
          border-left:20px groove red;
          */

        /*border:50px ridge red;*/
        /*border:50px inset red;*/
        /*border:50px outset red;*/
        border:50px solid red;
        border-image:url(/1.jpg);
      }
    </style>
  </head>
  <body>
    <div></div>
  </body>
</html>
```

弧度实例：


```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title></title>
    <style>
      div img{
        width:200px;
        height:200px;
        border:1px solid red;
        border-radius:0px 0px 200px 200px;
      }

      #p1{
        width:200px;
        height:100px;
        background:red;
        border-radius:0px 0px 100px 100px;
      }

      #p2{
        width:200px;
        height:100px;
        background:red;
        border-radius:100px 100px 0px 0px;
      }

      #p3{
        width:100px;
        height:200px;
        background:red;
        border-radius:0px 100px 100px 0px;
      }
      #div1{
        width:0px;
        height:0px;
        border-width:40px;
        border-style:solid;
        border-color:transparent transparent blue blue;
      }
    </style>
  </head>
  <body>
    <div></div>
    <p id="p1"></p>
    <p id="p2"></p>
    <p id="p3"></p>
    <div id="div1"></div>
  </body>
</html>

```

6.6 鼠标光标样式

<p>请把鼠标移动到单词上，可以看到鼠标指针发生变化：</p>

```
<span style="cursor:crosshair">
十字架</span><br />
<span style="cursor:default">
默认</span><br />
<span style="cursor:pointer">
小手</span><br />
<span style="cursor:move">
移动</span><br />
<span style="cursor:e-resize">
左右扩大</span><br />
<span style="cursor:ne-resize">
右斜着扩大</span><br />
<span style="cursor:nw-resize">
左斜着扩大</span><br />
<span style="cursor:n-resize">
上下扩大</span><br />
<span style="cursor:text">
文本</span><br />
<span style="cursor:wait">
等待</span><br />
<span style="cursor:help">
帮助</span>
```

第7章 实战布局

7.1 常见布局分类

div+css 布局的分类：

一列浮动布局

二列浮动布局

三列浮动布局

混合浮动布局

其实不管什么布局，都是人为的起的名字，就看布局者如何归类。

比如京东的首页：



比如淘宝的首页布局：

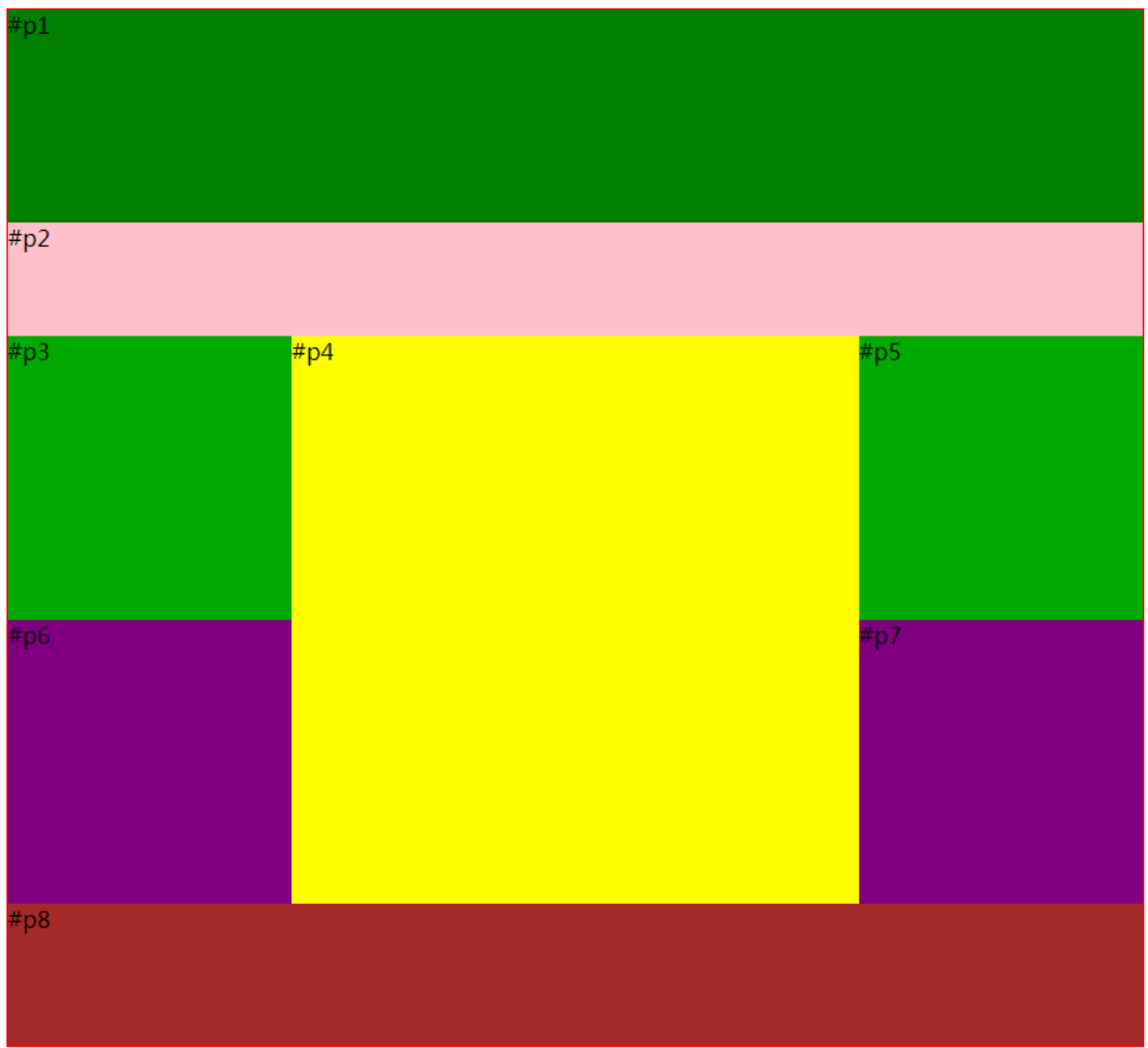


比如魅族的首页布局：

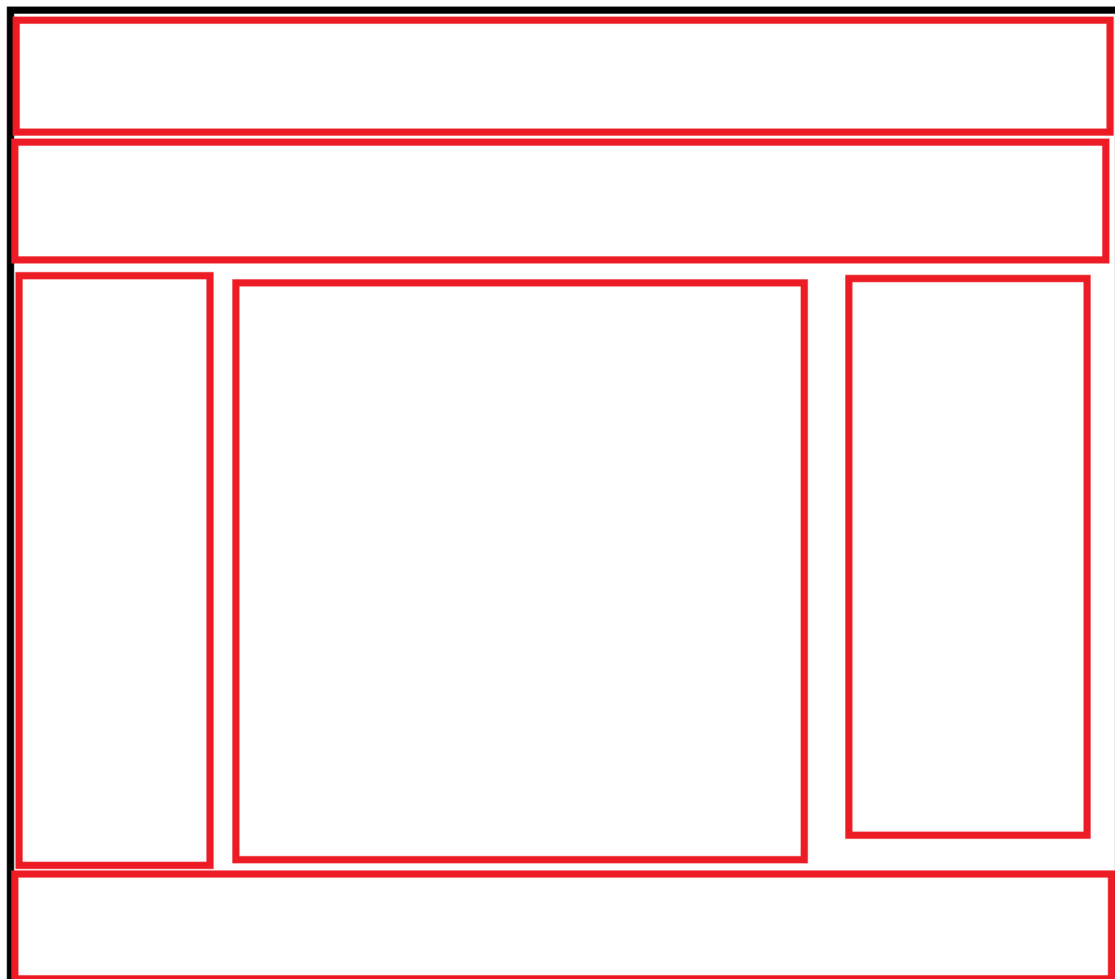


7.2 布局练习

完成如图样式：



布局分析：



大家记住了，做任何项目，第一步都是分析，有了确切的可执行方案以后，再进行代码操作。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title> </title>
    <style>
      .clear{
        clear:both;
      }
      #ddiv{
        width:800px;
        margin:0px auto;
        border:1px solid red;
      }
      #ddiv>.p1{
        width:800px;
        background:green;
        height:150px;
      }
      #ddiv>.p2{
        width:800px;
        height:80px;
        background:pink;
      }
    </style>
  </head>
  <body>
    <div class="clear">
      <div class="p1"></div>
      <div class="p2"></div>
    </div>
  </body>
</html>
```



```

        </div>
        <div class="center">#p4</div>
        <div class="right">
            <div class="p5">#p5</div>
            <div class="p7">#p7</div>
        </div>
        <div class="clear"></div>
    </div>
    <div id="p8">#p8</div>
</div>
</body>
</html>

```

7.3 企业站布局

布局效果如下：



```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
    <style type="text/css">
        /* reset 样式重置, 百度关键词: css reset */
        *{margin: 0px;padding: 0px;}
        img{border:0px;}
        ul li{list-style:none;}
        a{text-decoration:none;}
        /* 实际的样式 */
        .clear{
            clear: both;
        }
    </style>

```

```
.fl{
    float: left;
}
.fr{
    float: right;
}

.w{
    width: 1000px;
    /* border: 1px solid red; */
}

#header{
    width: 1000px;
    height: 80px;
    background: #eee;
}
#header .logo{
    margin-left: 20px;
    width: 150px;
    height: 80px;
    background: yellow;
}
#nav{
    width: 1000px;
    background: black;
    margin-top: 20px;
}
#nav li{
    float: left;
    padding: 0px 20px;
}
#nav li a{
    color: white;
}

#main{
    height: 400px;
    background: blue;
    margin-top: 20px;
}

#main .left{
    width: 700px;
    height: 400px;
    background: green;
}

#main .left .article{
    width: 650px;
    height: 180px;
    background: orange;
    margin: 10px 20px;
}

#main .right{
```



```

        width: 250px;
        height: 400px;
        background: orange;
    }

    #footer{
        height: 100px;
        width: 1000px;
        background: teal;
        margin-top: 20px;
    }
</style>
</head>
<body>
    <div class="w">
        <div id="header">
            <div class="logo"><img src="" /></div>
        </div>
        <ul id="nav">
            <li><a href="" >首页</a></li>
            <li><a href="" >企业产品</a></li>
            <li><a href="" >企业文化</a></li>
            <li><a href="" >产品功能</a></li>
            <li><a href="" >合作单位</a></li>
            <li><a href="" >联系我们</a></li>
            <div class="clear"></div>
        </ul>
        <div id="main">
            <div class="left fl">
                <div class="article">
                    <h3>爆炸性新闻</h3>
                    <p>昨夜发生了一件离奇案件,某程序员与项目经理深夜交谈.....<a href=""
title="">详细查看</a></p>
                </div>
                <div class="article">
                    <h3>爆炸性新闻</h3>
                    <p>昨夜发生了一件离奇案件,某程序员与项目经理深夜交谈.....<a href=""
title="">详细查看</a></p>
                </div>
            </div>
            <div class="right fr"></div>
        </div>
        <div id="footer"></div>
    </div>
</body>
</html>

```

7.4 HTML5 新标签补充

header 表示页面中的一个内容各区域或者整体的内容标题

nav 表示连接导航部分

footer 表示整个页面或者页面中的一个内容区域的脚注，一般用来，包含作者姓名文件创建时间或者作者联系方式

`section` 定义文档或者应用程序的一个内容区域，文章章节 页眉

`article` 文档中的一块独立内容，写文章，写新闻

`aside` 侧边栏，制作滚动广告

注意：以上这些标签虽然名字不一样，但其实都是div，只不过换了一个好听的名字，其定义也是人为的这么叫，主要方便搜索引擎根据标签搜索相应的内容时比较方便。

改造上面的html4.01写法为html5写法

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title> </title>
  <style>
    /*样式初始化*/
    *{margin:0px;padding:0px;}
    img{
      border:0px;
    }
    ul li{
      list-style:none;
    }
    a{
      text-decoration:none;
    }
    /*预定义样式*/
    .clear{
      clear:both;
    }
    .fl{
      float:left;
    }
    .fr{
      float:right;
    }
    .w{
      width:980px;
      margin:0px auto
    }
    header{
      width:980px;
      height:80px;
      background:#eeeeee;
    }
    header .logo{
      width:100px;
      height:80px;
      background:yellow;
      margin-left: 30px;
    }

    nav{
      width:980px;
```



```
<div id="main">
  <section class="f1">
    <article>
      <h3>炸裂新闻:昨夜某项目经理与某程序员大大出手</h3>
      <p>新闻简要:昨日, 项目经理提出手机开机可以根据主题自动变换后
壳颜色而大大出手, 专业人士老王友情提示, 请善待程序员。。。。<a href="">详情点击</a></p>
    </article>
    <article>
      <h3>伤感新闻:大爷健步如飞, 采访得知今年刚满25岁。</h3>
    </article>
  </section>
  <aside class="fr"></aside>
  <div class="clear"></div>
</div>
</body>
</html>
```

第8章 CSS3 新增属性

8.1 颜色属性

rgb (255,255,255) (100% , 100% , 100%)

rgba (255,255,255,0.5) a代表透明度取值 (0-1)

hsl

h:色调 (hue)

0-360 ---> 0和360表示红色, 120表示绿色, 240表示蓝色, 其他数值都是根据数值进行取值。

s:饱和度 (saturation)

0.0%---100% ---> 0% 黑色 100% 全彩色

l:亮度 (lightness)

0.0%-100% ---> 0.0 黑色 100% 白色

hsla a代表透明度 (0-1)



```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title></title>
    <style>
      p{
        color:rgb(100%,0%,100%);
```

```

    }
    p{
        color:rgba(255,0,0,0.2);
    }
    p{
        color:hs1(240,70%,20%);
    }
    P{
        color:hs1a(360,100%,50%,0.1);
    }
</style>
</head>
<body>
    <p>天王盖地虎,老王一米五</p>
</body>
</html>

```

8.2 背景图片

background-size 设置图片的大小的

100px 100px

参数1：宽，参数2：高

如果只设置一个值，后面值自动auto

100% 100%

参数1：宽，参数2：高

如果只设置一个值，后面值自动auto

background-size:cover 等比例缩放，超出部分不显示

background-size:contain 等比例缩放，缺少部分白色补全

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8"/>
        <title></title>
        <style>
            p{
                width:300px;
                height:300px;
                border:1px solid red;
                background:url(1.jpg) no-repeat;
                /*background-size:100px 100px;*/
                /*background-size:100% 100%;*/
                background-size:cover;
                /*background-size:contain;*/
            }
        </style>
    </head>
    <body>

```

```
<p></p>
</body>
</html>
```

8.3 文本属性

`text-overflow` 当文本溢出时显示省略号，但仅限于不换行的情况

`ellipsis` 显示省略号

但中文状态下就不会出现省略号，需要加另外一个参数

`white-space` 文本不进行换行显示。

`nowrap` 不换行

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    p{
      width:200px;
      height:20px;
      border:1px solid red;
      overflow:hidden;
      text-overflow:ellipsis;
      white-space:nowrap;
    }
  </style>
</head>
<body>
  <!-- <p>aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa</p> -->
  <p>同志们，胜利就在眼前了，要抓紧了</p>
</body>
</html>
```