



# 老马经典前端CSS3教程

@马伦-flydragon

# 课程内容简介

- 第一章：开发环境搭建
- 第二章：HTML5各种标签及语义化
- 第三章：CSS3及整站开发
- 第四章：JavaScript基础
- 第五章：JavaScript高级
- 第六章：移动端开发
- 第七章：前端高级框架
- 第八章：NodeJs
- 第九章：前端工作流及工程化

# 老马自我介绍

## 老马联系方式

QQ : 515154084

邮箱 : [malun666@126.com](mailto:malun666@126.com)

微博 : <http://weibo.com/flydragon2010>

百度传课 :

<https://chuanke.baidu.com/s5508922.html>



# 第一节：CSS基础入门

## CSS语法与选择器

1

CSS定义与编写CSS

2

CSS语法与选择器

3

CSS选择器优先级

4

CSS层叠和继承特性

# CSS的定义

- 什么是CSS？

CSS 指层叠样式表 (Cascading Style Sheets)

CSS通常称为CSS样式表或层叠样式表（级联样式表），主要用于设置HTML页面中的文本内容（字体、大小、对齐方式等）、图片的外形（宽高、边框样式、边距等）以及版面的布局等外观显示样式。

在HTML4标准之前，都是通过标签的属性或者样式标签控制样式。

- CSS就是控制页面布局和样式

- CSS的版本：2.1→3.0

- 类比例子：奶油蛋糕、画画

- HTML和CSS的关系：

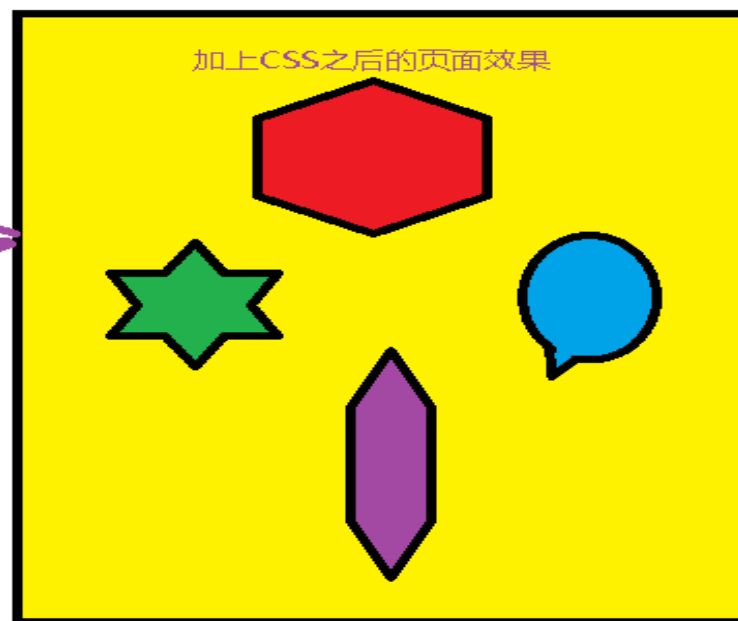
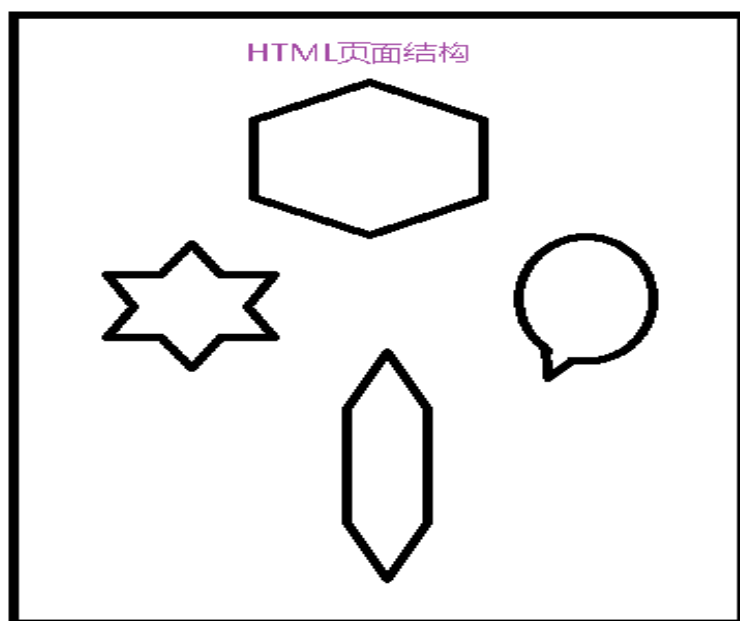
HTML 结构层      负责从 语义的角度搭建页面结构

CSS 样式层   负责从 审美的角度美化页面

JavaScript 行为层   负责从 交互的角度提升用户体验

# HTML和CSS的关系

- HTML：页面结构
- CSS：页面样式表现
- JavaScript：交互行为



# 如何编写CSS

- 内联样式 style属性（优先级最高）

```
<span style="color:red;"></span>
```

- 内嵌（嵌入，内部）样式（优先级次于内联样式）  
在head标签中添加style标签。

```
<head>  
<style> p {color:red;}</style>  
</head>
```

- 外部样式，外联样式（link）（优先级次于内嵌样式）  
通过link标签引入css样式文件

```
<link rel="stylesheet" href="a.css" />
```

type属性：只有一个选项，“text/css”，指定当前为css文本文件

rel：指定当前HTML文件与CSS文件的关系是样式表。href：指定外联样式表的路径

- 导入样式（优先级低于外联样式）

@import，导入样式会导致，css文件不能并行下载。不推荐使用。

导入样式的书写必须在所有的css规则书写之前

- 缺省样式（浏览器样式）（优先级最低）

可以通过设置修改浏览器的样式。

所有的标签都有默认的风格：h1,h2,p,div,span,ul等

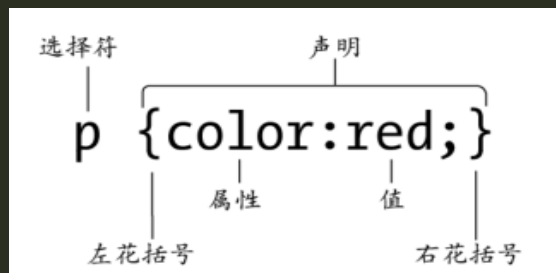
优先级依次降低



# CSS语法

- 选择符
- 属性声明
- 括号
- 换行不敏感、空格不敏感
- 案例：

```
div { color:red;}  
p{font-size:14px;color:blue;}  
h1 {  
    width: 20px;  
    height:200px;  
}
```





# CSS简单属性

- width : 设置宽度 , 单位px像素
- height : 高度
- color : 前景色 , 也就是文字的颜色
- background-color : 背景色。
- font-size:字体的大小。

# 案例：CSS设置样式

- 设置p标签的背景色为红色
- 设置H1标签的字体颜色为绿色
- 设置span标签的文本为14像素

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

  h1 {color: green;}

  span { font-size: 14px;}
</style>
```

# CSS的注释

- CSS的注释格式： `/* 注释内容 */`
- 可以同时注释多行语句比如：

```
/*单行注释*/
```

单行注释

```
/*多行注释  
多行注释  
多行注释*/
```

多行注释

```
/*  
* 文件声明注释  
* 作者：传智播客  
* 目的：演示注释的用法  
*/
```

常用用法 不是语法

# CSS选择器综述

- 所有标签选择器 `* {}`
- 标签选择器 `p {}` `div {}`
- ID选择器 `#head {}`
- 类选择器 `.head {}`
- 层级选择器
- 分组选择器
- 属性选择器
- 子元素选择器
- 相邻兄弟选择器
- 伪类选择器
- 伪元素选择器

# 通配符选择器

通配符选择器用 “\*” 号表示，他是所有选择器中作用范围最广的，能匹配页面中所有的元素。其基本语法格式如下：

```
*{属性1:属性值1; 属性2:属性值2; 属性3:属性值3; }
```

例如下面的代码，使用通配符选择器定义CSS样式，清除所有HTML标记的默认边距。

```
* {  
    margin: 0;           /* 定义外边距*/  
    padding: 0;          /* 定义内边距*/  
}
```

通配符的穿透力很强，优先级高于继承的样式，会覆盖继承的样式。**一般不用。**

# 标签选择器

- 通过标签名来选择一类标签。
- `p { color: red; }`
- 选择所有的p标签都设置成 字体为红色。

# ID选择器



HTML标签都有公共ID属性，而且整个页面唯一



id一般用于css的选择器和js的钩子




id选择器使用“#”进行标识，后面紧跟id名



#id名{属性1:属性值1; 属性2:属性值2; 属性3:属性值3; }

# ID选择器命名规范

- 1) 只允许出现字母（大小写均可，严格区分）、下划线、数字。  
也就是说，`id="laoHe"`和`id="laohe"`不冲突
  - 2) 只允许以字母开头
  - 3) 命名没有长度限制，可以是 1 个字母，也可以是很多个。不过不建议太长。
  - 4) 不允许出现标签名（不是硬性规定，是有工作经验的表现）
- 

注意：

id选择器的优先级非常高，所以确定在整个页面内唯一出现时，才可以使用id选择器，不然因为优先级问题在后续维护中不能很好进行修改更新。



# 类选择器

## •什么是类？

- 物以类聚，人以群分。
- 黄种人、白种人、欧洲人、亚洲人
- 所有的鸭子、所有的猫、
- 所有的按钮、所有的图片、所有的侧边栏、所有的banner

## •类选择器，是对HTML标签中class属性进行选择。 CSS类选择器的选择符是“.”

```
.demo { color:red;}
```

## •标签可以包含多个类选择器，在class标签中用空格隔开

例如：

html： `<p class="demo c2"> 类标签选择器</p>`

css： `.demo { color: red; }`

# 选择器综合案例

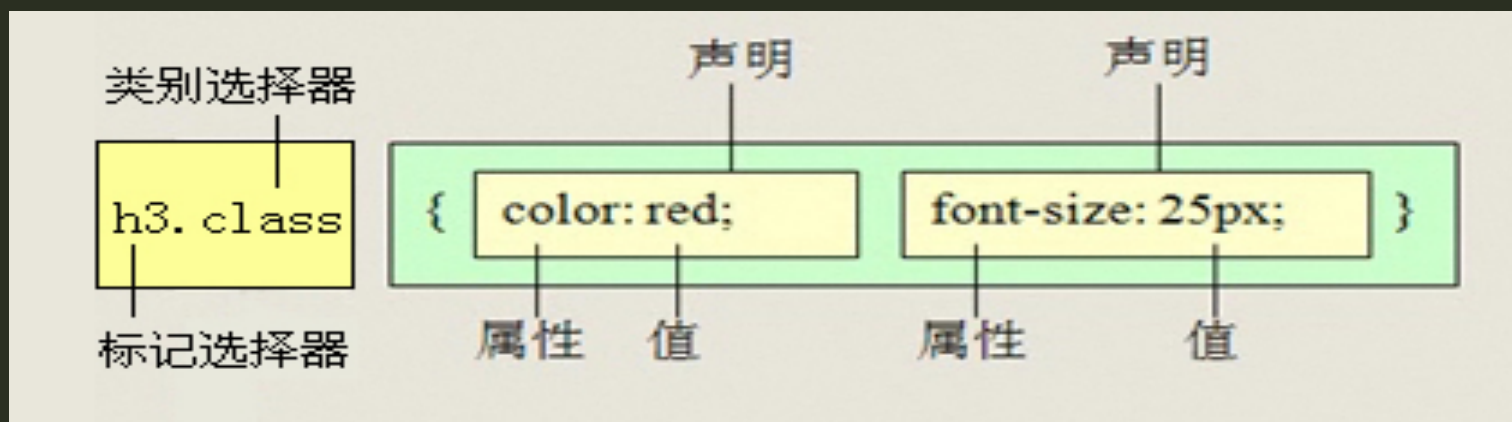
- 新闻案例：新闻内容见备注。
- 要求：
  - 标题：颜色为红色，字体30像素。
  - 评论数：颜色为黄色，字体18px
  - 日期：颜色为紫色，字体18px
  - 新闻内容，字体18px，字体颜色为银灰色。

# CSS复合选择器

复合选择器是由两个或多个基础选择器，通过不同的方式组合而成的，具体如下

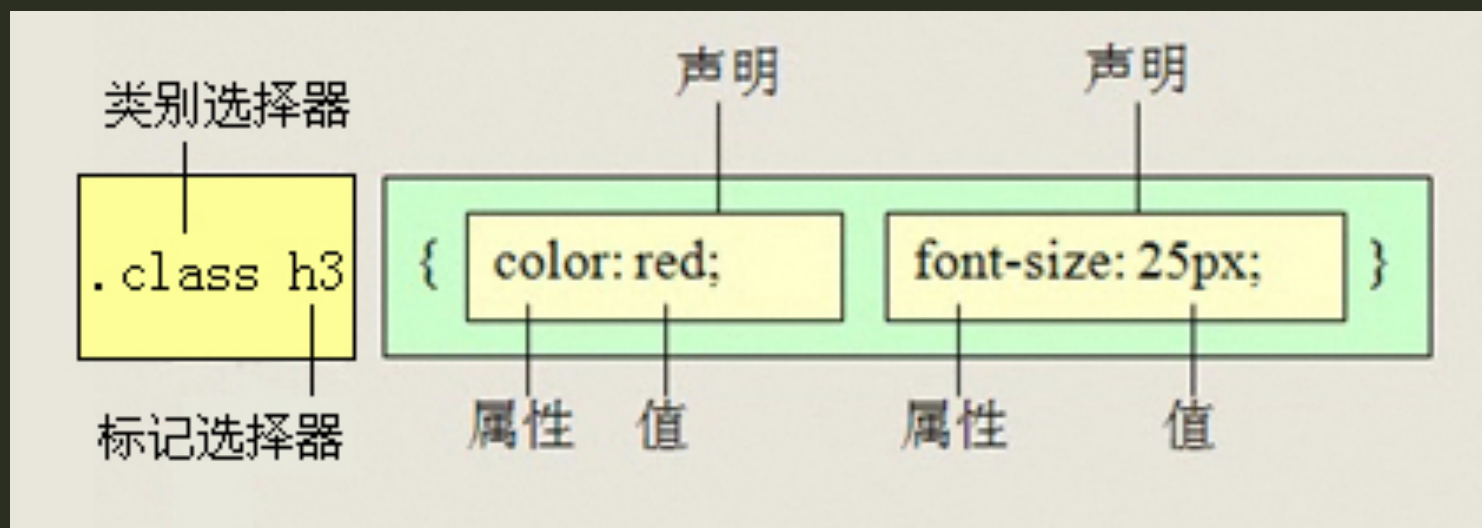
## 1、标签指定式选择器（即....又....）

标签指定式选择器又称交集选择器，由两个选择器构成，其中第一个为标记选择器，第二个为class选择器或id选择器，两个选择器之间不能有空格，如h3.special或p#one。



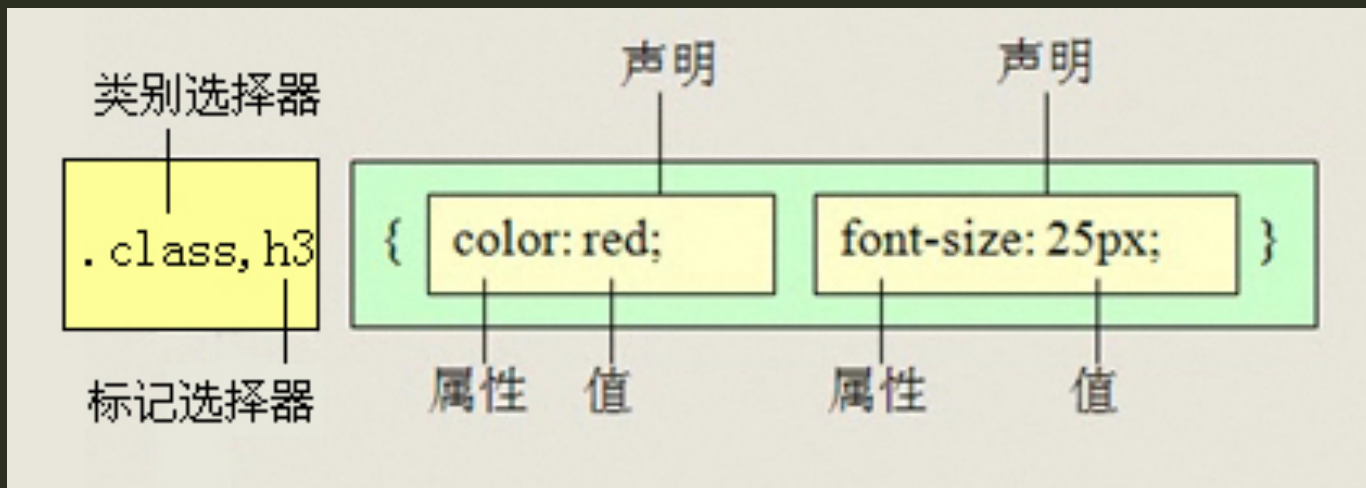
# 后代选择器(包含选择器)

后代选择器用来选择元素或元素组的后代，其写法就是把外层标记写在前面，内层标记写在后面，中间用空格分隔。当标记发生嵌套时，内层标记就成为外层标记的后代。



# 并集选择器

并集选择器是各个选择器通过逗号连接而成的，任何形式的选择器（包括标记选择器、class类选择器id选择器等），都可以作为并集选择器的一部分。如果某些选择器定义的样式完全相同，或部分相同，就可以利用并集选择器为它们定义相同的CSS样式。



# 子元素选择器

子代选择器，是让CSS选择器智能选择儿子辈的元素。

例如：`h1 > strong {color:red;}`

解读为：选择器 `h1 > strong` 可以解释为“选择作为 `h1` 元素子元素的所有 `strong` 元素”。

尖括号和选择器之间可以有空格也可以没有，没有限制。

建议:选择器和尖括号间有空格。

# 属性选择器

## ➞ 简单属性选择

```
h1[class]{ color:red;}
选择所有拥有class属性的
h1标签
h1[class][id]{color:red;}
```

## ➞ 根据属性值选择

```
p[id="aside">{color:red;}
根据属性名相等选择
```

## ➞ 属性名全包含

```
p[class~="a">{color:rd;}
只要包含属性，就被选
择。ie6不支持。
```

## ➞ 属性模糊选择

```
E[att^="val"] { sRules }
E[att$="val"] { sRules }
选择具有att属性且属性值
为以val开头（结尾）的字符串的E元素。
```

## ➞ 属性模糊匹配包含

```
E[att*="val"] { sRules }
选择具有att属性且属性值
为包含val的字符串的E元素。
```

# 属性选择器综合案例

- 选择所有的文本框标签
  - `input[type="textbox"]{}`
- 选择包含aside类的所有div标签
  - `div[class~="aside"]{}`
- 选择拥有id和class属性的p标签
  - `p[id][class]{}`



# CSS伪类和伪元素

- 状态是动态变化的，当一个元素达到一个特定状态时，它可能得到一个样式，当状态改变时，又失去这个样式。
- 它的功能和class有些类似，但它是基于文档之外的抽象，所以叫伪类

# 伪类

:link

伪类将应用于未被访问过的链接。IE6不兼容，解决此问题，直接使用a标签。

:hover

伪类将应用于有鼠标指针悬停于其上的元素。在IE6只能应用于a连接，IE7+所有元素都兼容。

:active

伪类将应用于被激活的元素，如被点击的链接、被按下的按钮等。

:visited

伪类将应用于已经被访问过的链接

:focus

伪类将应用于拥有键盘输入焦点的元素。

顺序问题：**LoVe** **HAte**原则。

综合案例，电商菜单。

菜单1

菜单2

菜单3

菜单4

菜单5

# CSS伪元素

`:first-child` 伪类，选择属于第一个子元素的元素。

例如：`span:first-child{ /*选择属于第一个子元素的所有span标签。 */`

伪元素 是控制内容。在CSS3的标准高中是用两个冒号。

`:first-line` 伪元素

`:first-letter` 伪元素

注释：以上两个伪元素只能用于块级元素

`:before`与`:after`伪元素，可以设置元素之前后之后的 内容，并且配合  
`content`设置相关内容。比如：`#demo:after,#demo:before {`  
`content:"--";display:block; }`

# CSS层叠性

所谓层叠性是指多种CSS样式的叠加。例如，当使用内嵌式CSS样式表定义<p>标记字号大小为12像素，链入式定义<p>标记颜色为红色，那么段落文本将显示为12像素红色，即这两种样式产生了叠加

# CSS继承性

所谓继承性是指书写CSS样式表时，子标记会继承父标记的某些样式，如文本颜色和字号。想要设置一个可继承的属性，只需将它应用于父元素即可。

- 恰当地使用继承可以简化代码，降低CSS样式的复杂性。但是，如果在网页中所有的元素都大量继承样式，那么判断样式的来源就会很困难，所以对于字体、文本属性等网页中通用的样式可以使用继承。例如，字体、字号、颜色、行距等可以在body元素中统一设置，然后通过继承影响文档中所有文本。
- a标签不继承父容器的字体颜色。
- 所有字相关的都可以继承，比如：color、text-系列、font-系列、line-系列、cursor
- 并不是所有的CSS属性都可以继承，例如，下面的属性就不具有继承性：边框、外边距、内边距、背景、定位、元素宽高属性。
- 所有盒子相关的属性都不能继承。

# 关于优先级

将军  
内联样式

师长  
#

旅长  
.

士兵  
p

0

1

0

30

0

0

1

0

备注：1个将军= 255个师长，1个师长=255旅长，1个旅长=255士兵，256个士兵 > 1个旅长

# CSS的优先级

内联样式  
0

ID选择器  
1

类选择器  
678

标签选择器

内联样式最大，内联样式的优先级最高。

ID选择器的优先级，仅次于内联样式。

类选择器优先级低于ID选择器

标签选择器低于类选择器。

256个标签选择器相加 大于 一个类选择器

256个类选择器相加 大于 一个ID选择器



## • CSS优先级

定义CSS样式时，经常出现两个或更多规则应用在同一元素上，这时就会出现优先级的问题。

在考虑权重时，初学者还需要注意一些特殊的情况，具体如下：

- **继承样式的权重为0**。即在嵌套结构中，不管父元素样式的权重多大，被子元素继承时，他的权重都为0，也就是说子元素定义的样式会覆盖继承来的样式。
- **行内样式优先**。应用style属性的元素，其行内样式的权重非常高，可以理解为远大于100。总之，他拥有比上面提高的选择器都大的优先级。

- 权重相同时，CSS遵循**就近原则**。也就是说靠近元素的样式具有最大的优先级，或者说排在最后的样式优先级最大。
- 所有都相同时，声明靠后的优先级大。
- CSS定义了一个!important命令，该命令被赋予最大的优先级。也就是说不管权重如何以及样式位置的远近，!important都具有最大优先级。
- 综述：
  - 行内样式 > 页内样式 > 外部引用样式 > 浏览器默认样式
  - **important > 内联 > ID > 伪类|类 | 属性选择 > 标签 > 伪对象 > 通配符 > 继承**
  - **综合案例。**

## 第二节：CSS单位与文字排版

标签的显示模式与Display属性

CSS的值和单位

CSS的文字排版

- **标签的类型(显示模式)**

- HTML标记一般分为块标记和行内标记两种类型，它们也称块元素和行内元素。具体如下：

- 块元素

- 每个块元素通常都会独自占据一整行或多整行，可以对其设置宽度、高度、对齐等属性，常用于网页布局和网页结构的搭建。
      - 块级元素容器可以容纳多个嵌套的块级标签或者行内标签。
      - 常见的块元素有<h1>~<h6>、<p>、<div>、<ul>、<ol>、<li>等，其中<div>标记是最典型的块元素。

- 标签的类型(显示模式)

- 行内元素

- 行内元素不占有独立的区域，仅仅靠自身的字体大小和图像尺寸来支撑结构，一般不可以设置宽度、高度、对齐等属性，常用于控制页面中文本的样式。
- 行内元素不能嵌套块级标签，只能嵌套其他的行内标签。
- 常见的行内元素有<strong>、<b>、<em>、<i>、<del>、<s>、<ins>、<u>、<a>、<span>等，其中<span>标记最典型的行内元素。

# Display改变标签显示模式

- display可以控制标签的显示模式。
- display : none | inline | block | inline-block
- 继承性：无
- display值的解释：
  - none：此元素不被显示，在文档中被移除。
  - block：此元素按块级元素显示：前后带换行符，自己占一行。内联元素 → 块元素
  - inline：此元素按内联元素显示：1个挨着1个。块元素 → 内联元素
  - inline-block：按行内标签进行排版，但是可以设置宽高，而且高度可以影响行高（以后再详细讲）。
  - 关于display的其他值，我们后续再解

# Display属性的兼容处理\*\*

- IE6,7支持inline元素转换成inline-block，但不支持block元素转换成inline-block，所以非inline元素在IE6,7下要转换成inline-block，需先转换成inline，然后触发hasLayout，以此来获得和inline-block类似的效果。

```
div {  
    display: inline-block;  
    *display: inline;  
    *zoom: 1;  
}
```

# CSS的颜色表示

- 命名颜色

red、black、white、purple、green、yellow、silver、gray、teal、blue、navy....[备注]

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

- RGB颜色表示法

RGB模式表示色彩，R：red，G：green，B：blue，三种颜色组合叠加可以产生自然界中所有的颜色，也是计算机常用模式。

RGB 颜色值是这样规定的：rgb(red, green, blue).

每个参数 (red, green 以及 blue) 定义颜色的强度，可以是介于 0 与 255 之间的整数，或者是百分比值（从 0% 到 100%）。

```
p { color: rgb(255,255,0);};
```

```
p { color: rgb(100%,100%,0%);};
```

- 十六进制颜色值

由于RGB模式书写非常不方便，三个255值直接用3个十六进制数表示。

十六进制颜色是这样规定的：#RRGGBB，其中的 RR（红色）、GG（绿色）、BB（蓝色）十六进制整数规定了颜色的成分。所有值必须介于 0 与 FF 之间。

取值范围 #000000 ~ #FFFFFF（黑色到白色）

其实就是RGB模式的另外一种写法，所有浏览器都兼容。

```
p { color: #c0c0c0; } p { color: #c09; } 三个数字的复制扩展成六个  
#c09=>#cc0099。
```

拓展作业：什么是Web安全色？



# CSS的长度单位

- 绝对长度单位：

cm：厘米，mm：毫米，in：英寸，pc：派卡（Picas），相当于我国新四号铅字的尺寸。都不常用，了解即可

- 相对长度单位：

px：像素点，像素就是显示器显示的一个点。

em：1em 等于当前的字体大小，例如：当前元素的字体大小为16px，那么1em = 16px。

- 单位之间的关系：1in = 2.54cm = 25.4

p { width: 1in; height: 20px; }

# CSS字体样式：字体大小

## 1. font字体大小

font-size属性用于设置字号，该属性的值可以使用相对长度单位，也可以使用绝对长度单位。

其中，相对长度单位比较常用，推荐使用像素单位px，绝对长度单位使用较少。

可选参数值：xx-small | x-small | small | **medium** | large | x-large | xx-large | smaller | larger

一般页面中：12px    14px    12cm

1em

例如：

```
p { font-size: 32px; }
```

## 2、font-family:字体

font-family属性用于设置字体。网页中常用的字体有宋体、微软雅黑、黑体等，例如将网页中所有段落文本的字体设置为微软雅黑，可以使用如下CSS样式代码：

```
p{ font-family:"微软雅黑";}
```

可以同时指定多个字体，中间以逗号隔开，表示如果浏览器不支持第一个字体，则会尝试下一个，直到找到合适的字体。如果字体名称包含空格或中文，则应使用引号括起

例如：font-family:Verdana, Arial, '宋体';

- 使用font-family设置字体时，需要注意以下几点：
- 各种字体之间必须使用英文状态下的逗号隔开。
- 中文字体需要加英文状态下的引号，英文字体一般不需要加引号。当需要设置英文字体时，英文字体名必须位于中文字体名之前。
- 如果字体名中包含空格、#、\$等符号，则该字体必须加英文状态下的单引号或双引号，例如font-family: "Times New Roman";。
- 尽量使用系统默认字体，保证在任何用户的浏览器中都能正确显示。

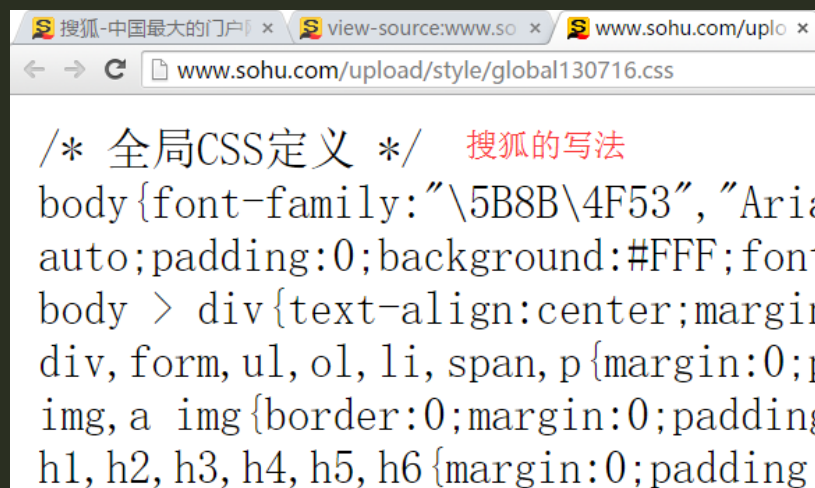
**使用font-family设置字体时，需要注意以下几点：**

在 CSS 中设置字体名称，直接写中文是可以的。但是在文件编码（GB2312、UTF-8 等）不匹配时会产生乱码的错误。

为此，在 CSS 直接使用 Unicode 编码来写字体名称可以避免这些错误。使用 Unicode 写中文字体名称，浏览器是可以正确的解析的。

–font-family: “\5FAE\8F6F\96C5\9ED1”，表示设置字体为“微软雅黑”。

–可以通过`escape()` 来得到



```
/* 全局CSS定义 */  搜狐的写法
body{font-family:“\5B8B\4F53”, "Aria
auto;padding:0;background:#FFF;font
body > div{text-align:center;margin
div, form, ul, ol, li, span, p{margin:0;p
img, a img{border:0;margin:0;padding
h1, h2, h3, h4, h5, h6 {margin:0;padding:
```

使用font-family设置字体时，需要注意以下几点：

字体名称	英文名称	Unicode 编码
宋体	SimSun	\5B8B\4F53
新宋体	NSimSun	\65B0\5B8B\4F53
黑体	SimHei	\9ED1\4F53
微软雅黑	Microsoft YaHei	\5FAE\8F6F\96C5\9ED1
楷体_GB2312	KaiTi_GB2312	\6977\4F53_GB2312
隶书	LiSu	\96B6\4E66
幼儿园	YouYuan	\5E7C\5706
华文细黑	STXihei	\534E\6587\7EC6\9ED1
细明体	MingLiU	\7EC6\660E\4F53
新细明体	PMingLiU	\65B0\7EC6\660E\4F53

# 字体系列

- 西方国家字母体系分为两类：serif以及sans serif。
- serif是有衬线字体，意思是在字的笔画开始、结束的地方有额外的装饰，而且笔画的粗细会有所不同。相反的，sans serif就没有这些额外的装饰，而且笔画的粗细差不多。
- 为了更安全的设置：  
`font-family: tahoma,arial,'Hiragino Sans GB',\5b8b\4f53,sans-serif;`
- 前面的字体都查找失败后，在系统中找一种sans-serif字体作为默认字体。
- 注意顺序，如果把sans-serif放前面去，后面的都失效了。

无衬线字体	AaBbCc
衬线字体	AaBbCc
衬线字体的衬线 (红色部分)	AaBbCc

### 3、font-weight:字体粗细

font-weight属性用于定义字体的粗细，其可用属性值：normal、bold、bolder、lighter、100~900（100的整数倍），有继承性。

400 : Normal

700 : bold

字体的加粗跟字体有关，比如：一种字体只有两种粗细程度的变化，那么normal 到bold 和到bolder都是一样的效果。



## 4、font-style:字体风格

font-style属性用于定义字体风格，如设置斜体、倾斜或正常字体，其可用属性值如下：

- normal：默认值，浏览器会显示标准的字体样式。
- italic：浏览器会使用斜体的字体样式显示，如果字体没有斜体，那么正常显示字体。
- oblique：浏览器会让文字倾斜显示。。

## 5、font:综合设置字体样式

font属性用于对字体样式进行综合设置，其基本语法格式如下：

```
选择器{font: font-style font-weight font-size/line-height font-family;}
```

使用font属性时，必须按上面语法格式中的顺序书写，各个属性以空格隔开。

其中不需要设置的属性可以省略（取默认值），但必须保留font-size和font-family属性，否则font属性将不起作用。

淘宝的字体设置：font: 12px/1.5 tahoma,arial,'Hiragino Sans GB',\5b8b\4f53,sans-serif;

- **CSS文本外观属性**

- 1、**color**:文本颜色

**color**属性用于定义文本的颜色，其取值方式有如下3种：

- 预定义的颜色值，如red, green, blue等。
- 十六进制，如#FF0000, #FF6600, #29D794等。实际工作中，十六进制是最常用的定义颜色的方式。
- RGB代码，如红色可以表示为rgb(255,0,0)或rgb(100%,0%,0%)。

需要注意的是，如果使用**RGB**代码的百分比颜色值，取值为0时也不能省略百分号，必须写为0%。

## 2、**letter-spacing**:字间距

**letter-spacing**属性用于定义字间距，所谓字间距就是字符与字符之间的空白。其属性值可为不同单位的数值，允许使用负值，默认为**normal**。

### 3、word-spacing:单词间距

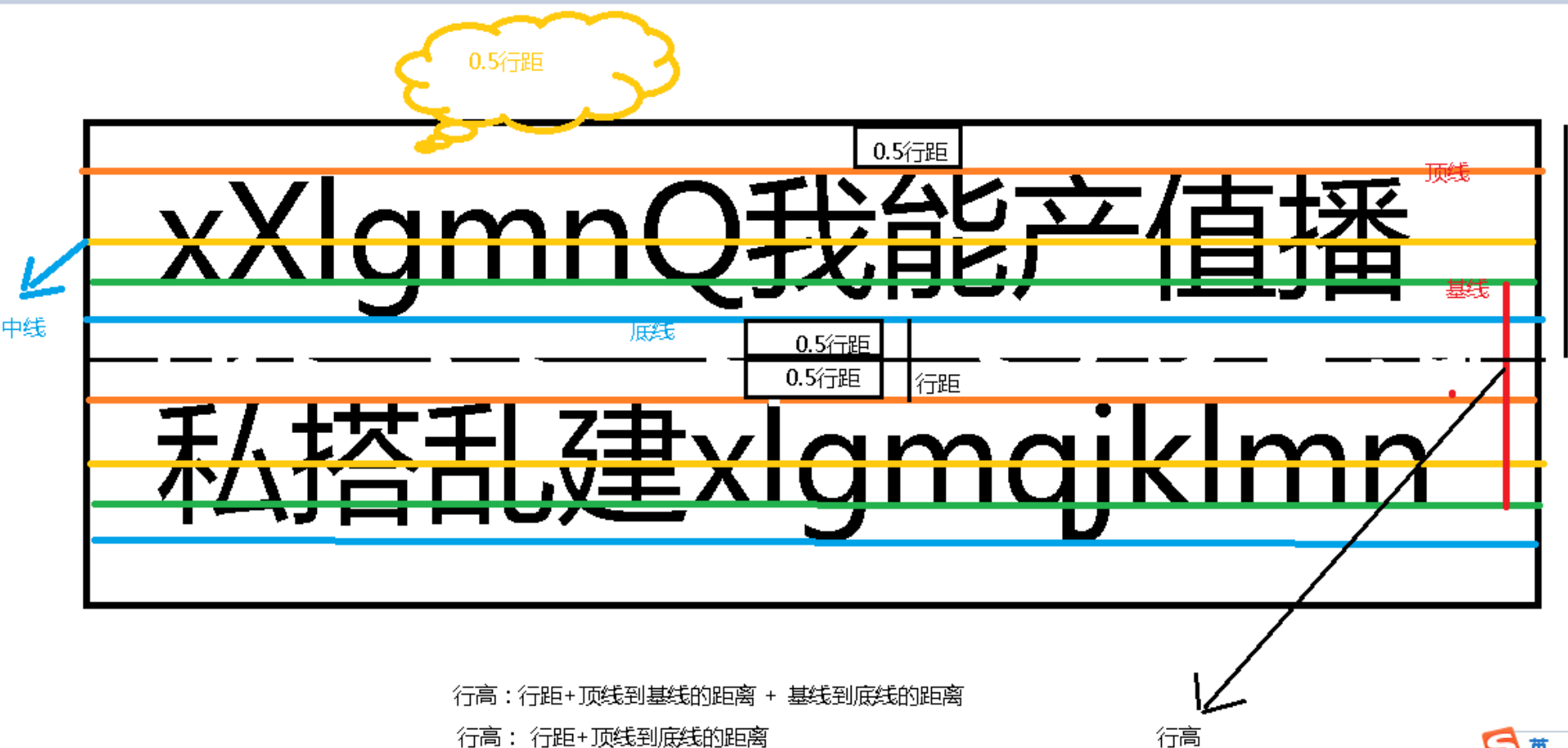
–word-spacing属性用于定义英文单词之间的间距，对中文字符无效。和letter-spacing一样，其属性值可为不同单位的数值，允许使用负值，默认为normal。

–word-spacing和letter-spacing均可对英文进行设置。不同的是letter-spacing定义的为字母之间的间距，而word-spacing定义的为英文单词之间的间距。

## 4、line-height:行间距

line-height属性用于设置行间距，就是行与行之间的距离，即字符的垂直间距，一般称为行高。line-height常用的属性值单位有三种，分别为像素px，相对值em和百分比%，实际工作中使用最多的是像素px。

# 行高原理



## 玩转行高

行高可以跟单位，也可以不跟单位。但是意义不一样。

font: 12px/1.5em '宋体';

font:12px/150% '宋体';

font:12px/1.5 '宋体';

font: 12px/20px '宋体';

继承特性

加上单位 先计算 后继承     是以父亲为基准

不加单位 先继承 后计算     是孩子根据自己的文字大小为基准

line-height : 可以让单行文本垂直居中。

行高的详细案例参考：

<http://www.cnblogs.com/fengzheng126/archive/2012/05/18/2507632.html>



## 5、text-decoration:文本装饰

text-decoration属性用于设置文本的下划线，上划线，删除线等装饰效果，其可用属性值如下：

`<s>` 删除线 `</s>`

–none：没有装饰（正常文本默认值）。

–underline：下划线。

–overline：上划线。

–line-through：删除线。

另外，text-decoration后可以赋多个值，用于给文本添加多种显示效果，例如希望文字同时有下划线和删除线效果，就可以将underline和line-through同时赋给text-decoration。

## 6、text-align:水平对齐方式

text-align属性用于设置文本内容的水平对齐，相当于html中的align对齐属性。其可用属性值如下：

–left：左对齐（默认值）

–right：右对齐

–center：居中对齐

## 7、text-indent:首行缩进

text-indent属性用于设置段落首行文本的缩进，**只能设置于块级标签**。其属性值可为不同单位的数值、em字符宽度的倍数、或相对于浏览器窗口宽度的百分比%，允许使用**负值**，建议使用em作为设置单位。

## 8、white-space:空白符处理

使用HTML制作网页时，不论源代码中有多少空格，在浏览器中只会显示一个字符的空白。在CSS中，使用white-space属性可设置空白符的处理方式，其属性值如下：

- normal：常规（默认值），文本中的空格、空行无效，满行（到达区域边界）后自动换行。
- pre：预格式化，按文档的书写格式保留空格、空行原样显示。
- nowrap：空格空行无效，强制文本不能换行，除非遇到换行标记<br />。内容超出元素的边界也不换行，若超出浏览器页面则会自动增加滚动条。

## 8、word-break:自动换行

normal      使用浏览器默认的换行规则，不中断单词显示。

break-all   允许在单词内换行。

keep-all   只能在半角空格或连字符处换行。

## 8、word-wrap

属性允许长单词或 URL 地址换行到下一行normal

normal 只在允许的断字点换行（浏览器保持默认处理）。

break-word 在长单词或 URL 地址内部进行换行。

几乎得到了浏览器的支持

### 中乙队赛前突然换帅仍胜毅腾 高原黑马欲阻击舜天

2014年07月16日20:11 新浪体育 评论中大奖(11人参与) [收藏本文](#)

新浪体育讯 7月16日是燕京啤酒[\[微博\]](#)2014中国足协杯第三轮比赛，丽江嘉云昊队主场迎战哈尔滨毅腾队的比赛日。然而就在比赛日中午，丽江嘉云昊队主帅李虎和另外两名成员悄然向俱乐部提出了辞呈，并且收拾行囊准备离开。在这样的情况下，丽江嘉云昊队不得不由此前的教练员杨贵东代理指挥了本场比赛。

在昨日丽江嘉云昊队主帅李虎就缺席了赛前的新闻发布会，当时俱乐部给出的解释是李虎由于身体欠佳，去医院接受治疗。然而今日李虎出现在俱乐部时，向记者否认了这一说法，并且坦言已经向俱乐部提出了辞呈。

据记者多方了解的情况，李虎[\[微博\]](#)极其教练组近来在执教成绩上承受了不小的压力，在联赛间歇期期间，教练组曾向俱乐部提出能够多引进有实力的球员补强球队，然而由于和俱乐部在投入以及成绩指标上的分歧，李虎最终和教练组一起在比赛日辞职。

这样的情况并没有影响到丽江嘉云昊队[\[微博\]](#)的队员，在比赛中丽江队在主场拼的非常凶，在暴雨之中仍然发挥出了体能充沛的优势，最终凭借点球击败了中超球队哈尔滨毅腾，顺利晋级下一轮比赛。根据中国足协杯的赛程，丽江嘉云昊队将在本月23日迎战江苏舜天队。

# 中乙队赛前突然换帅仍胜毅腾 高原黑马欲阻击舜天

2014年07月16日20:11 新浪体育 评论中大奖 (11人参与) [收藏本文](#)

801

新浪体育讯 7月16日是燕京啤酒[微博]2014中国足协杯第三轮比赛，丽江嘉云昊队主场迎战哈尔滨毅腾队的比赛日。然而就在比赛日中午，丽江嘉云昊队主帅李虎和另外两名成员悄然向俱乐部提出了辞呈，并且收拾行囊准备离开。在这样的情况下，丽江嘉云昊队不得不由此前的教练员杨贵东代理指挥了本场比赛。

在昨日丽江嘉云昊队主帅李虎就缺席了赛前的新闻发布会，当时俱乐部给出的解释是李虎由于身体欠佳，去医院接受治疗。然而今日李虎出现在俱乐部时，向记者否认了这一说法，并且坦言已经向俱乐部提出了辞呈。

据记者多方了解的情况，李虎[微博]极其教练组近来在执教成绩上承受了不小的压力，在联赛间歇期期间，教练组曾向俱乐部提出能够多引进有实力的球员补强球队，然而由于和俱乐部在投入以及成绩指标上的分歧，李虎最终和教练组一起在比赛日辞职。

这样的情况并没有影响到丽江嘉云昊队[微博]的球员，在比赛中丽江队在主场拼的非常凶，在暴雨之中仍然发挥出了体能充沛的优势，最终凭借点球击败了中超球队哈尔滨毅腾，顺利晋级下一轮比赛。根据中国足协杯的赛程，丽江嘉云昊队将在本月23日迎战江苏舜天队。

# 第三节

## PART THREE

### 盒模型

认识盒模型  
盒模型相关属性  
盒模型边距合并

老马自我介绍

老马联系方式

QQ : 515154084

邮箱 : [malun666@126.com](mailto:malun666@126.com)

微博 : <http://weibo.com/flydragon2010>

百度传课 :

<https://chuanke.baidu.com/s5508922.html>





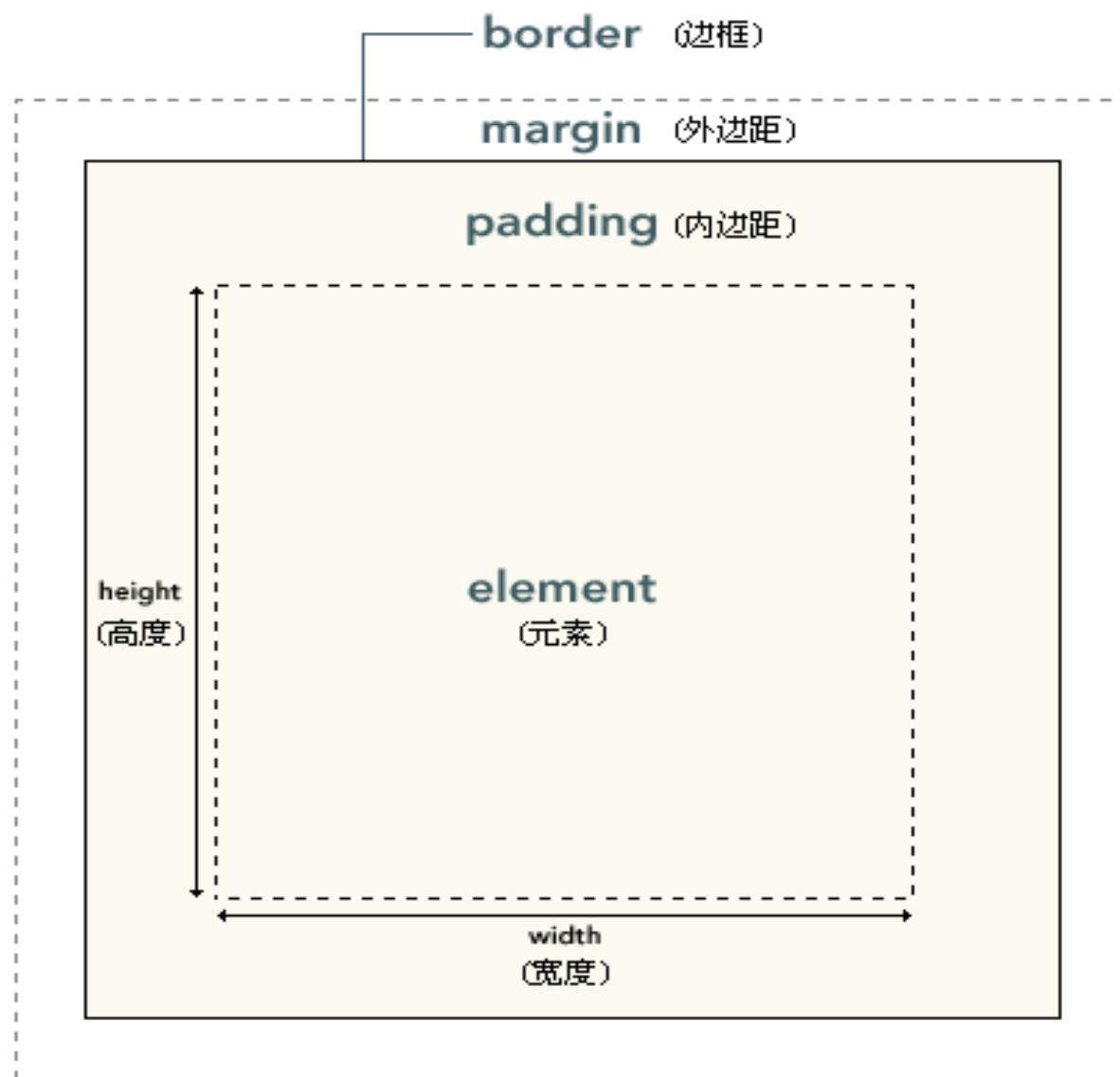
## • 盒子模型的概念

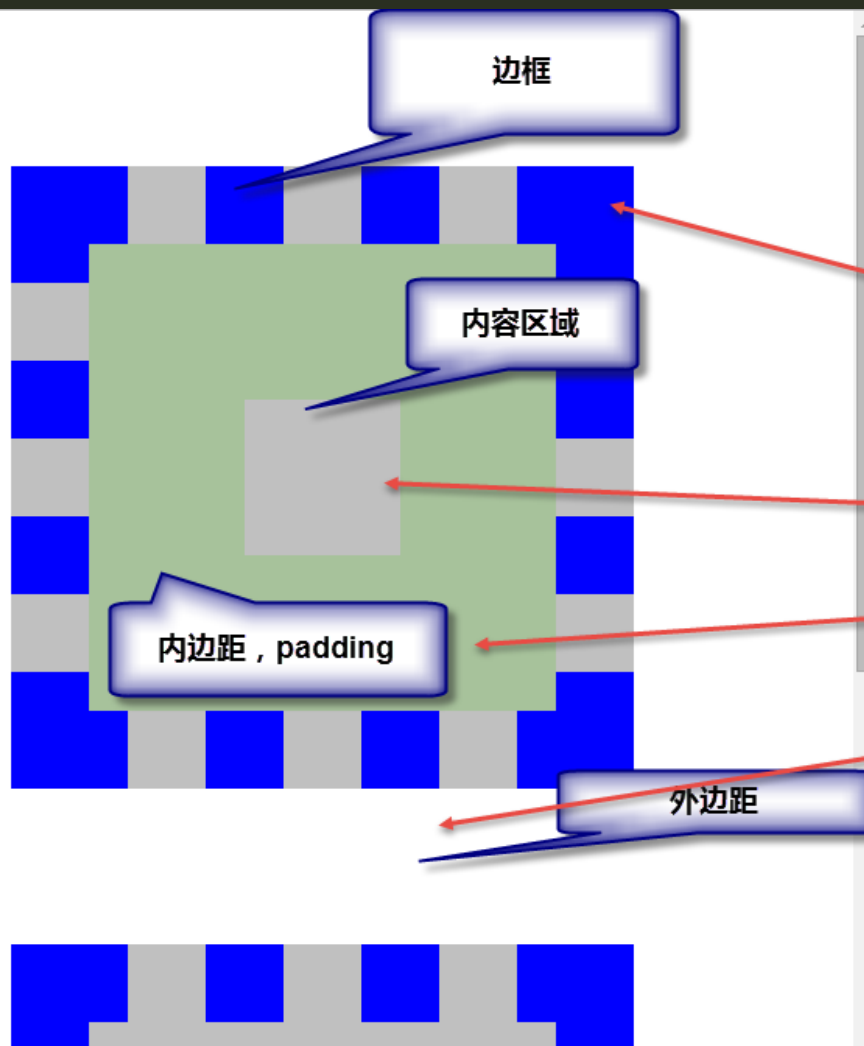
所谓**盒子模型**就是把HTML页面中的元素看作是一个矩形的盒子，也就是一个盛装内容的容器。

每个矩形都由元素的**内容**、**内边距**（padding）、**边框**（border）和**外边距**（margin）组成。

# 盒模型

- DT
- 内容
- 内边距
- 边框
- 外边距
- do
- 盒模





Elements | Network | Sources | Timeline | Profiles | Resources | Audits | Console

```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body>
    <div class="box"></div>
    <div class="box"></div>
  </body>
</html>
```

html body div.box

Styles | Event Listeners | DOM Breakpoints | Properties

element.style { }

.box {

- width: 100px;
- height: 100px;
- background-color: silver;
- border: 50px dotted blue;
- margin: 100px;
- padding: 100px;

}

\* {

- padding: 0;
- margin: 0;

}

div {

- display: block;

}

05.htm:8

05.htm:7

user agent stylesheet

Find in Styles

Console | Search | Emulation | Rendering

<top frame> Preserve log

Navigated to file:///D:/webpro/05.htm

margin 100

border 50

padding 100

100 x 100

100 50

100 50

100

50

100

Show inherited properties

- background-color: rgb(192, 192, 192);
- border-bottom-color: rgb(0, 0, 255);

Filter

## 结论：

- 网页就是多个盒子嵌套排列的结果。
- 内边距出现在内容区域的周围，当给元素添加背景色或背景图像时，该元素的背景色或背景图像也将出现在内边距中。
- 外边距是该元素与相邻元素之间的距离。
- 如果给元素定义边框属性，边框将出现在内边距和外边距之间。

**注意：**虽然盒子模型拥有内边距、边框、外边距、宽和高这些基本属性，但是并不要求每个元素都必须定义这些属性

## • 边框属性

设置内容	样式属性	常用属性值
上边框	border-top-style:样式;	
	border-top-width:宽度;	
	border-top-color:颜色;	
	border-top:宽度 样式 颜色;	
下边框	border-bottom-style:样式;	
	border-bottom-width:宽度;	
	border-bottom-color:颜色;	
	border-bottom:宽度 样式 颜色;	
左边框	border-left-style:样式;	
	border-left-width:宽度;	
	border-left-color:颜色;	
	border-left:宽度 样式 颜色;	
右边框	border-right-style:样式;	
	border-right-width:宽度;	
	border-right-color:颜色;	
	border-right:宽度 样式 颜色;	
样式综合设置	border-style:上边 [右边 下边 左边];	none无（默认）、solid单实线、dashed虚线、dotted点线、double双实线
宽度综合设置	border-width:上边 [右边 下边 左边];	像素值
颜色综合设置	border-color:上边 [右边 下边 左边];	颜色值、#十六进制、rgb(r,g,b)、rgb(r%,g%,b%)
边框综合设置	border:四边宽度 四边样式 四边颜色;	

- **边框属性—设置边框样式 ( border-style )**
  - **边框样式**用于定义页面中边框的风格，**常用属性值**如下：
    - none：没有边框即忽略所有边框的宽度（默认值）
    - solid：边框为单实线
    - dashed：边框为虚线
    - dotted：边框为点线
    - double：边框为双实线

- 边框属性—设置边框样式 ( border-style )

- 既可以对盒子的单边进行设置，也可以综合设置四条边的样式：

- border-top-style:上边框样式
- border-right-style:右边框样式
- border-bottom-style:下边框样式
- border-left-style:左边框样式
- border-style:上边框样式 右边框样式
- border-style:上边框样式 左右边框样式
- border-style:上下边框样式 左右边框样式
- border-style:上下左右边框样式

- 使用border-style属性综合设置四边样式时，必须按上右下左的顺时针顺序。
- 省略时采用值复制的原则，即一个值为四边，两个值为上下/左右，三个值为上/左右/下。

- **边框属性—设置边框样式 ( border-style )**

- 例如<p>只有上边为虚线dashed，其他三边为单实线solid，可以使用border-style综合属性分别设置各边样式：

```
p{ border-style:dashed solid solid solid;}
```

- 或综合设置四条边，然后采用上边覆盖：

```
p{ border-style:solid;}      /*综合设置四边样式*/  
p{ border-top-style:dashed;} /*上边样式覆盖*/
```



- 边框属性—设置边框宽度 ( border-width )

- ●border-top-width : 上边框宽度  
边框宽度也可以针对四条边分别设置，或综合设置四条边的宽度：

- border-right-width : 右边框宽度

- border-bottom-width : 下边框宽度

- border-left-width : 左边框宽度

- border-width : 上边框宽度 [ 下边框宽度 [ 左边框宽度 [ 右边框宽度 ] ] ] ]

综合设置四边宽度必须按上右下左的顺时针顺序采用值复制，即一个值为四边，两个值为上下/左右，三个值为上/左右/下。

## • 4.2.1 边框属性—设置边框颜色 ( border-color )

— 边框颜色的单边与综合设置如下：

- border-top-color: 上边框颜色
  - border-right-color: 右边框颜色
  - border-bottom-color: 下边框颜色
  - border-left-color: 左边框颜色
  - border-color: 上边框颜色 [右边框颜色] [下边框颜色] [左边框颜色]
- 顺时针顺序，即一个值为四边，两个值为上下/左右，三个值为上/左右/下。
  - 其取值可为预定义的颜色值、十六进制 #RRGGBB 或 RGB 代码 rgb(r,g,b)，最常用的是十六进制 #RRGGBB。

- 边框属性—设置边框颜色 ( border-color )

**注意：**

设置边框颜色时同样**必须设置边框样式**，如果未设置样式或设置为none，则其他的边框属性无效。

- **边框属性—综合设置边框**

- 像border、border-top等这样，能够一个属性定义元素的多种样式，在CSS中称之为**复合属性**。
- 常用的**复合属性**有font、border、margin、padding和background等。
- 复合属性可以**简化代码**，**提高**页面的**运行速度**，但是如果只有一项值，最好不要应用复合属性，以免样式不被兼容。

## 多学几招

### 1. 表单的边框通常改为 0 ;

`border:0 none;//兼容所有浏览器`

### 2. 表格的细线边框

`table { border-collapse:collapse; }`

### 3. 轮廓（ 链接有虚线 和 文本框 有 蓝色边框 ）

`outline-style:none;`

- **内边距属性**

- padding属性用于设置内边距，也是复合属性，相关设置如下：

- padding-top:上内边距
    - padding-right:右内边距
    - padding-bottom:下内边距
    - padding-left:左内边距
    - padding:上内边距[右内边距 下内边距 左内边距]
    - 内边距是享有 背景色的。

- 外边距属性

- margin-top:上外边距
- margin-right:右外边距
- margin-bottom:下外边距
- margin-left:左外边距
- margin:上外边距 [右外边距 下外边距 左外边距]

- margin相关属性的值，以及复合属性margin取1~4个值的情况与padding相同。
- 外边距可以使用负值，使相邻元素重叠。

- **外边距属性**

- 对块元素应用宽度属性width，并将左右的外边距都设置为auto，可使块级元素水平居中，实际工作中常用这种方式进行网页布局，示例代码如下：

```
.header{ width:960px; margin:0 auto;}
```

- 为了更方便地控制网页中的元素，制作网页时，可使用如下代码清除元素的默认内外边距：

```
*{  
    padding:0;      /*清除内边距*/  
    margin:0;       /*清除外边距*/  
}
```



- 外边距属性

注意：

使用margin定义块元素的垂直外边距时，可能会出现外边距的合并。

## 行内元素 关于padding 和margin 问题

注意：

- | 行内元素不要给上下的margin 和padding
- | 上下margin和padding会被忽略。
- | 左右margin和padding会起作用。

- **盒子的宽与高**

- 使用宽度属性width和高度属性height可以对盒子的大小进行控制。
- width和height的属性值可以为不同单位的数值或相对于父元素的百分比%，实际工作中最常用的是像素值。
- 大多数浏览器，如Firefox、IE6及以上版本都采用了W3C规范，符合CSS规范的盒子模型的总宽度和总高度的计算原则是：
  - 盒子的总宽度= width+左右内边距之和+左右边框宽度之和+左右外边距之和
  - 盒子的总高度= height+上下内边距之和+上下边框宽度之和+上下外边距之和

- 盒子的宽与高

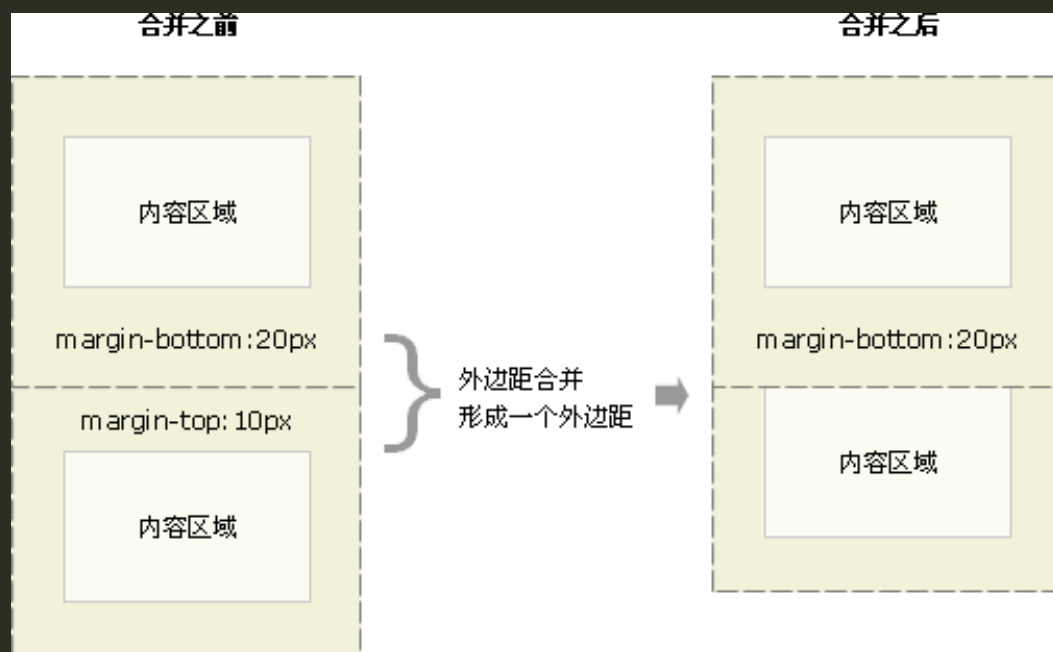
注意：

- 1、宽度属性width和高度属性height仅适用于块级元素，对行内元素无效（<img />标记和<input />除外）。
- 2、计算盒子模型的总高度时，还应考虑上下两个盒子垂直外边距合并的情况。

- 相邻块元素垂直外边距的合并

当上下相邻的两个块元素相遇时，如果上面的元素有下外边距margin-bottom，下面的元素有上外边距margin-top，则他们之间的垂直间距不是margin-bottom与margin-top之和，而是两者中的较大者。这种现象被称为相邻块元素垂直外边距的合并（也称外边距塌陷）。

- 相邻块元素垂直外边距的合并



- **嵌套块元素垂直外边距的合并**

- 对于两个嵌套关系的块元素，如果父元素没有上**内边距**及**边框**，则父元素的上外边距会与子元素的上外边距发生合并，合并后的外边距为两者中的较大者，即使父元素的上外边距为0，也会发生合并。
- 如果希望外边距不合并，可以为父元素定义1像素的上边框或上内边距。

## 第四节：浮动高级



浮动原理  
浮动处理  
浮动与布局



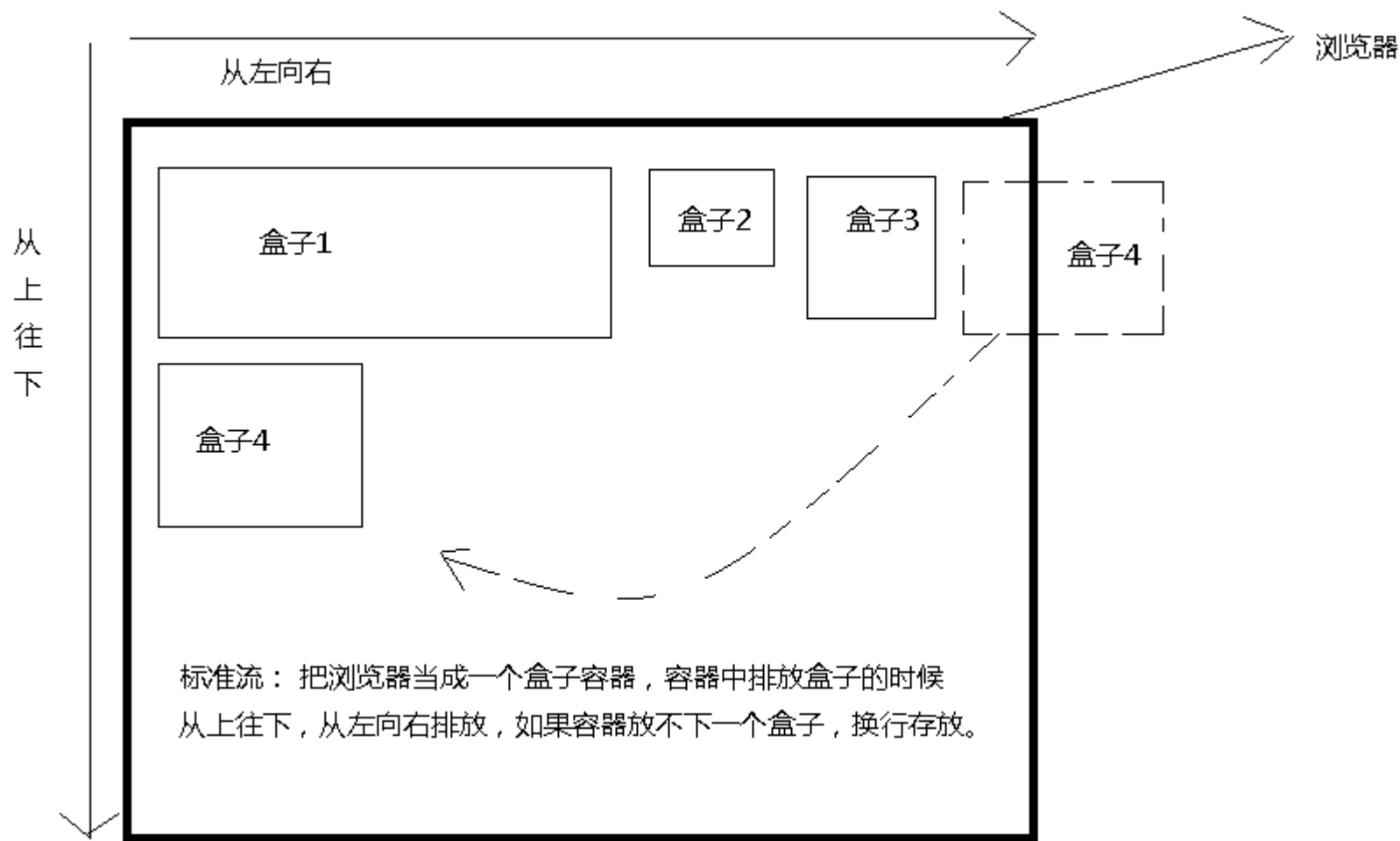
内容溢出处理



- 标准流(normal flow)

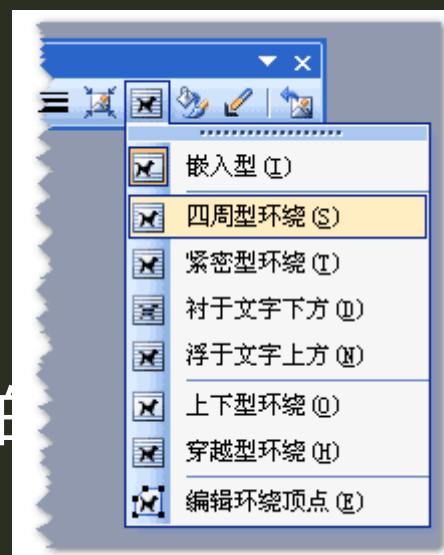
html语言当中另外一个相当重要的概念-----标准流！或者普通流。标准流实际上就是一个网页内标签元素正常排列的顺序的意思；比如块级元素会独占一行，行内元素会按顺序依次从左向右，从上向下排列；按照这种大前提的布局排列之下绝对不会出现列外的情况叫做标准流布局，也称做流式布局。

# 标准流，流式布局



# 浮动的本质

- word中图片和文字混排
- 浮动（float），浮动的盒子可以向左或向右移动，直到它的外边缘碰到包含盒子或另一个浮动盒子的边框为止
- 浮动做布局，仅仅是附带来的，是滥用的用于图文排版。



- 元素的浮动属性**float**

### 什么是浮动？

元素的浮动是指设置了浮动属性的元素会脱离标准文档流的控制，移动到其父元素中指定位置的过程。

### 如何定义浮动？

在CSS中，通过**float**属性来定义浮动，其基本语法格式如下：

```
选择器{float:属性值;}  none ,left, right
```

- 元素的浮动属性**float**

**float**属性值:

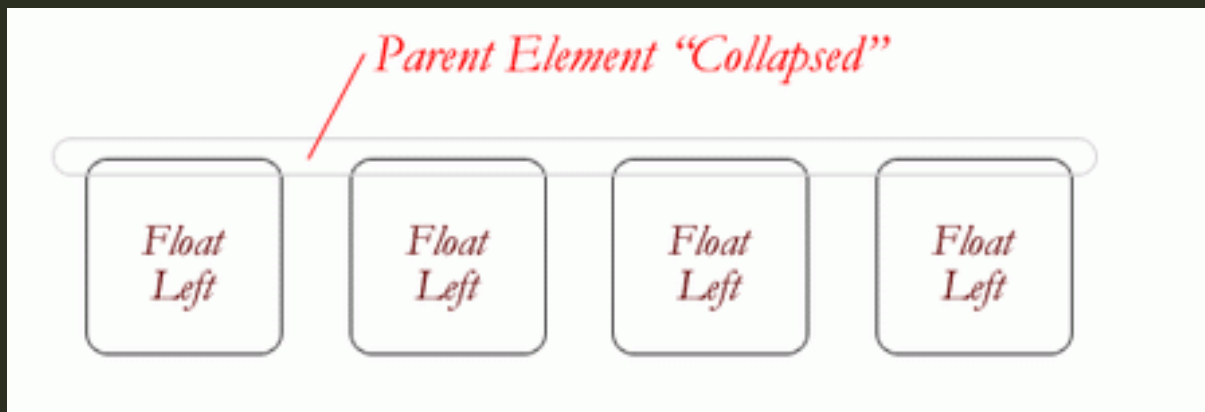
属性值	描述
left	元素向左浮动
right	元素向右浮动
none	元素不浮动（默认值）

## • 浮动特性★★★★★

1. 浮动脱离标准流，不占位置，但会影响标准流。浮动只有左右浮动。
2. 浮动的元素A排列位置，跟上一个元素（块级）有关系。如果上一个元素有浮动，则A元素顶部会和上一个元素的顶部对齐；如果上一个元素是标准流，则A元素的顶部会和上一个元素的底部对齐。
3. 一个父盒子里面的子盒子，如果其中一个子级有浮动的，则其他子级都需要浮动。这样才能一行对齐显示。
4. 浮动根据元素书写的位置来显示相应的浮动。
5. 元素添加浮动后，如果没有设置宽高的话，元素会具有行内块元素的特性。元素的大小完全取决于定义的大小或者默认的内容多少。也就是具有了包裹性。
6. 浮动具有破坏性，元素浮动后，破坏来原来的正常流布局，造成内容

# 父容器高度塌陷

- 如果一个标准流中的盒子所有的子元素都进行了浮动，而且盒子没有设置高度，那么父容器整个高度会塌陷。
- 如何实现自适应包裹浮动子元素呢？clearfix



## overflow属性的应用

当盒子内的元素超出盒子自身的大小时，内容就会溢出（IE6除外），这时如果想要规范溢出内容的显示方式，就需要使用CSS的overflow属性，其基本语法格式如下：

```
选择器{overflow:属性值;}
```



## overflow属性的常用值

属性值	描述
visible	内容不会被修剪，会呈现在元素框之外（默认值）
hidden	溢出内容会被修剪，并且被修剪的内容是不可见的
auto	在需要时产生滚动条，即自适应所要显示的内容
scroll	溢出内容会被修剪，且浏览器会始终显示滚动条

## 清除浮动

语法：`clear : left|right|both;`

clear是指让当前元素放置的时候，会自动判断左侧或右侧或者双侧是否有浮动的元素，如果有就往下放，如果没有的就放下。

# 清除浮动

## 清除浮动的基本语法格式

在CSS中，`clear`属性用于清除浮动，其基本语法格式如下：

```
选择器{float:属性值;}
```

- 清除浮动

clear属性的常用值

属性值	描述
left	不允许左侧有浮动元素（清除左侧浮动的影响）
right	不允许右侧有浮动元素（清除右侧浮动的影响）
both	同时清除左右两侧浮动的影响

- 清除浮动

常用四种清除浮动的方法：

- 方法一：使用空标记清除浮动，隔墙放。增加标签。
- 方法二：使用overflow属性清除浮动。会误伤。
- 方法三：使用after伪对象清除浮动。
- 方法四：使用before after伪对象清除浮动
- Clearfix

- 清除浮动

**after伪对象清除浮动时的注意事项:**

1. 必须为需要清除浮动的元素伪对象设置 “height:0;” 样式，否则该元素会比其实际高度高出若干像素。
2. 必须在伪对象中设置content属性，属性值可以为空，如 “content: "";” 。

```
.clearfix:after{ /*对类名为father父元素应用after伪对象样式*/  
    display:block;  
    clear:both;  
    content:"";  
    visibility:hidden;  
    height:0;  
}
```



- 版心和布局流程

### 为什么要应用布局？

阅读报纸时容易发现，虽然报纸中的内容很多，但是经过合理地排版，版面依然清晰、易读。同样，在制作网页时，要想使页面结构清晰、有条理，也需要对网页进行“排版”。

### 版心

“版心”是指网页中主体内容所在的区域。一般在浏览器窗口中水平居中显示，常见的宽度值为960px、980px、1000px等。



- 版心和布局流程

## 布局流程

为了提高网页制作的效率，布局时通常需要遵守一定的布局流程，具体如下：

- 1、确定页面的版心（可视区）。
- 2、分析页面中的行模块，以及每个行模块中的列模块。
- 3、运用盒子模型的原理，通过DIV+CSS布局来控制网页的各个模块。

# 第五节：CSS布局高级



# 元素的定位属性

元素的定位属性主要包括定位模式和边偏移两部分。

## 1、定位模式

在CSS中，`position`属性用于定义元素的定位模式，其基本语法格式如下：

选择器{`position`:属性值;}

### `position`属性的常用值

值	描述
<code>static</code>	自动定位（默认定位方式）
<code>relative</code>	相对定位，相对于其原文档流的位置进行定位
<code>absolute</code>	绝对定位，相对于其上一个已经定位的父元素进行定位
<code>fixed</code>	固定定位，相对于浏览器窗口进行定位

- 元素的定位属性

元素的定位属性主要包括定位模式和边偏移两部分。

- 2、边偏移

在CSS中，通过边偏移属性**top**、**bottom**、**left**或**right**，来精确定义定位元素的位置，其取值为不同单位的数值或百分比。具体解释如下表所示：

边偏移属性	描述
top	顶端偏移量，定义元素相对于其父元素上边线的距离
bottom	底部偏移量，定义元素相对于其父元素下边线的距离
left	左侧偏移量，定义元素相对于其父元素左边线的距离
right	右侧偏移量，定义元素相对于其父元素右边线的距离

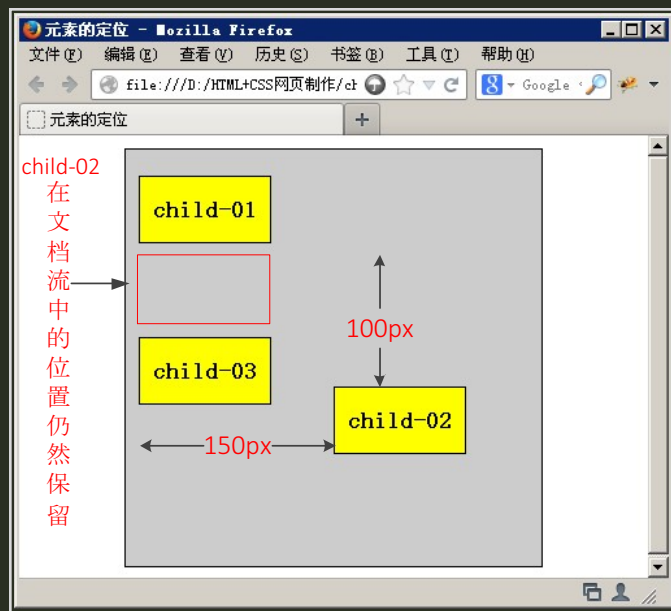
- 静态定位static

静态定位是元素的默认定位方式，当position属性的取值为static时，可以将元素定位于静态位置。所谓静态位置就是各个元素在HTML文档流中默认的位置。

在静态定位状态下，无法通过边偏移属性（top、bottom、left或right）来改变元素的位置。

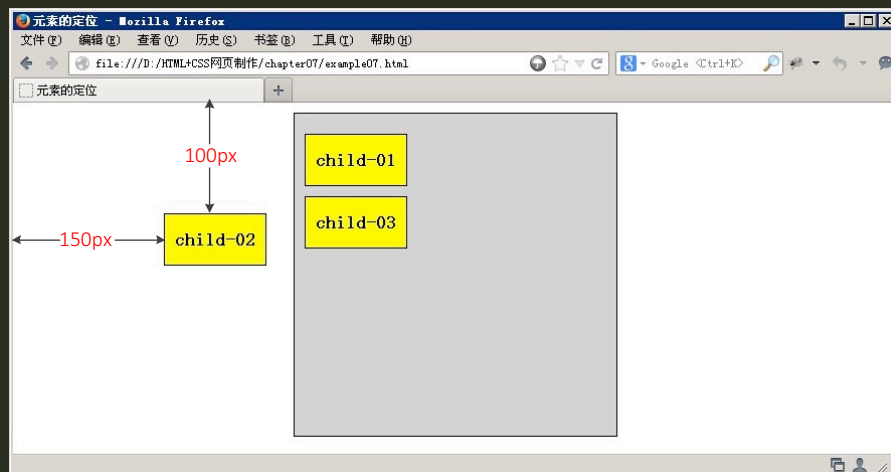
- 相对定位relative（自恋型）

相对定位是将元素相对于它在标准流中的位置进行定位，当position属性的取值为relative时，可以将元素定位于相对位置。对元素设置相对定位后，可以通过边偏移属性改变元素的位置，但是它在文档流中的位置仍然保留。如下图所示，即是一个相对定位的效果展示：



- 绝对定位absolute(看脸型)

绝对定位是将元素依据最近的已经定位（绝对、固定或相对定位）的父元素进行定位，若所有父元素都没有定位，则依据body根元素进行定位。当position属性的取值为absolute时，可以将元素的定位模式设置为绝对定位。如下图所示，即是一个相对定位的效果展示：



# 绝对定位的特性

- 包裹性
- 根据实际内容来撑开绝对定位元素的大小。
- 破坏性，导致父容器塌陷
- 由于脱离了文档流，导致父容器塌陷。
- 悬浮性
- 会漂浮起来，在原始流的元素之上。



## 固定定位fixed（说一不二型）

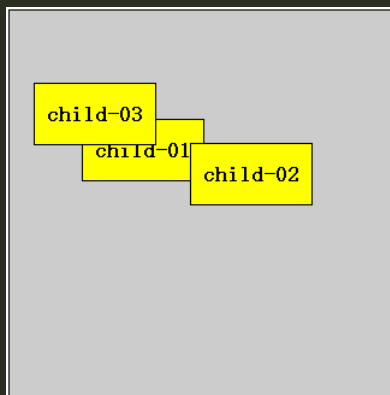
固定定位是绝对定位的一种特殊形式，它以浏览器窗口作为参照物来定义网页元素。当`position`属性的取值为`fixed`时，即可将元素的定位模式设置为固定定位。

当对元素设置固定定位后，它将脱离标准文档流的控制，始终依据浏览器窗口来定义自己的显示位置。不管浏览器滚动条如何滚动，也不管浏览器窗口的大小如何变化，该元素都会始终显示在浏览器窗口的固定位置。

**注意:**IE6浏览器不支持固定定位

- z-index层叠等级属性

当对多个元素同时设置定位时，定位元素之间有可能会发生重叠。



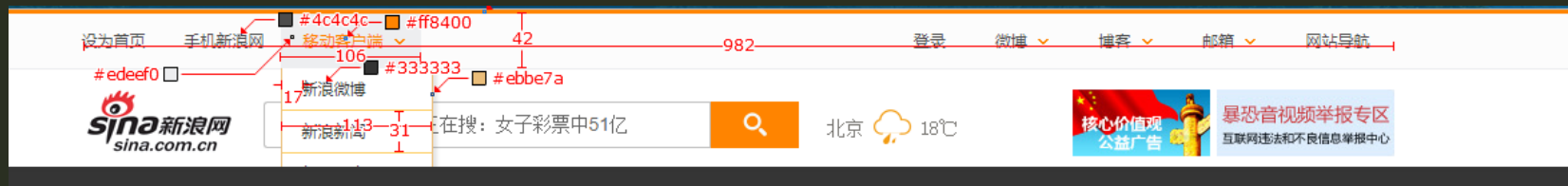
在CSS中，要想调整重叠定位元素的堆叠顺序，可以对定位元素应用z-index层叠等级属性，其取值可为正整数、负整数和0。z-index的默认属性值是0，取值越大，定位元素在层叠元素中越居上。

定义靠后的，默认在之前的元素之上。不要滥用z-index;  
父容器的z-index会影响子元素的层级级别。

# 绝对定位的应用

- 全屏
- 固定左侧边栏

# 综合案例



## 第六节： CSS实战技巧



CSS常用技巧



CSS精灵技术



CSS滑动门技术



Margin负值的应用

# CSS背景

- CSS的背景
- background-attachment      设置背景图像是否固定或者随着页面的其余部分滚动。
- background-color      设置元素的背景颜色。
- background-image      设置元素的背景图像。
- background-position      设置背景图像的开始位置。
- background-repeat      设置是否及如何重复背景图像。
- background 合写：在一个声明中设置所有的背景属性。

# background-image

- 可以设置盒子的背景图片：
- `background-image: url(bgimage.gif);`
- 注意url指向一个相对路径，url后面紧跟的是一对括号，括号内的是路径，路径可以用引号，也可以省略，建议省略。
- 背景图片会盖住背景颜色。
- 也就是说：背景图片的优先级要高于背景色

# background-attachment

- background-attachment属性设置背景图像是
- scroll：默认值。背景图像会随着页面其余部分的滚动而移动。否固定或者随着页面的其余部分滚动。
- fixed：固定显示，相对于body固定。一般只用于body的背景设置。
- background-attachment: fixed;



# background-position

- background-position 属性设置背景图像的起始位置。

值	描述
top left top center top right center left center center center right bottom left bottom center bottom right	如果您仅规定了一个关键词，那么第二个值将是"center"。  默认值：0% 0%。
x% y%	第一个值是水平位置，第二个值是垂直位置。  左上角是 0% 0%。右下角是 100% 100%。  如果您仅规定了一个值，另一个值将是 50%。
xpos ypos	第一个值是水平位置，第二个值是垂直位置。  左上角是 0 0。单位是像素 (0px 0px) 或任何其他 CSS 单位。  如果您仅规定了一个值，另一个值将是50%。  您可以混合使用 % 和 position 值。

# background-repeat

- background-repeat 属性定义了图像的平铺模式。

值	描述
repeat	默认。背景图像将在垂直方向和水平方向重复。
repeat-x	背景图像将在水平方向重复。
repeat-y	背景图像将在垂直方向重复。
no-repeat	背景图像将仅显示一次。
inherit	规定应该从父元素继承 background-repeat 属性的设置。

# background合写

- background合写的顺序：背景颜色、背景图地址、平铺设置、背景图滚动、背景图位置。
- background: #00FF00 url(bgimage.gif) no-repeat fixed top;
- 建议：背景都以合写的形式存在，CSS的字节更少。

# 背景综合案例



## CSS规避脱标

标准流（常规流）是及其稳定的写法，很多地方，我们尽量使用标准流

·  
优先选择标准，其次使用float，最后选择定位。

1. margin-left:auto; 左侧外边距自动

不用浮动块级元素也可以到右侧

2. vertical-align 和 inline-block 的使用

怎么让一个 盒子和文字对齐？

 主题市场

实现方式：可以使用浮动，也可以使用inline-block和 vertical-align

## CSS元素可见性

`overflow:hidden;`    `display:none;`    `visibility:hidden;`

单词	作用
<code>overflow: hidden</code>	隐藏超出的那部分
<code>display: none</code>	不显示元素 同时也不占位置
<code>visibility:hidden;</code>	不显示元素 但是占有位置

## CSS标签包含

### 常用标签嵌套

1.内联元素却不能包含块元素，它只能包含其他的内联元素

span> div

div> div>ul>li>div

2.有些块级元素不能放其他块级元素。

标题标记的<h1>、<h2>、<h3>、<h4>、<h5>、<h6>、<caption>；

段落标记的<p>； p>div

分隔线<hr>和<dt>

3、a标签不能嵌套 a和input标签，能嵌套的标签像,<b> <strong>等等。

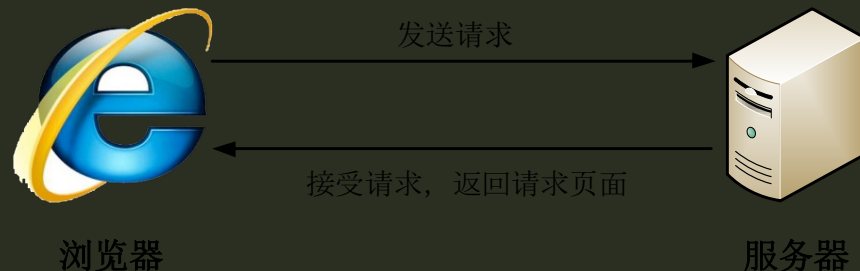
# CSS内容移除某个区域

常用于logo 优化

1. 利用text-indent:-2000em;
2. 利用padding 挤开盒子 并且overflow 切割。
- 3.利用margin拉动盒子配合overflow切割



## • CSS精灵技术



- 上图所示为网页的请求原理图，当用户访问一个网站时，需要向服务器发送请求，网页上的每张图像都要经过一次请求才能展现给用户。
- 然而，一个网页中往往会应用很多小的背景图像作为修饰，当网页中的图像过多时，服务器就会频繁地接受和发送请求，这将大大降低页面的加载速度。为了有效地减少服务器接受和发送请求的次数，提高页面的加载速度，出现了CSS精灵技术（也称CSS Sprites）。

- 简单地说，CSS精灵是一种处理网页背景图像的方式。它将一个页面涉及到的所有零星背景图像都集中到一张大图中去，然后将大图应用于网页，这样，当用户访问该页面时，只需向服务发送一次请求，网页中的背景图像即可全部展示出来。通常情况下，这个由很多小的背景图像合成的大图被称为精灵图，如下图所示为搜狗网站中的一个精灵图。

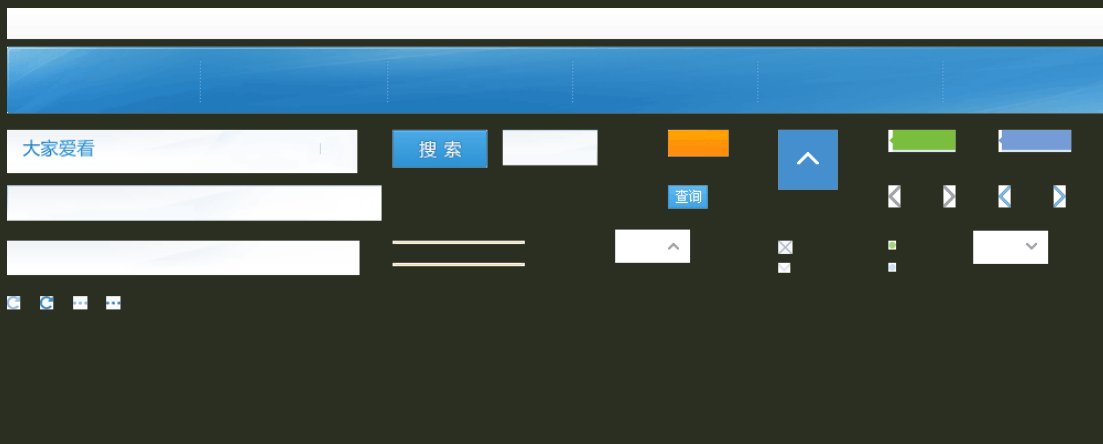


## • 工作原理

CSS 精灵其实是将网页中的一些背景图像整合到一张大图中（精灵图）。然而，各个网页元素通常只需要精灵图中不同位置的某个小图，要想精确定位到精灵图中的某个小图，就需要使用CSS的background-image、background-repeat和background-position属性进行背景定位，其中最关键的是使用background-position属性精确地定位。

## • 工作原理

CSS 精灵其实是将网页中的一些背景图像整合到一张大图中（精灵图）



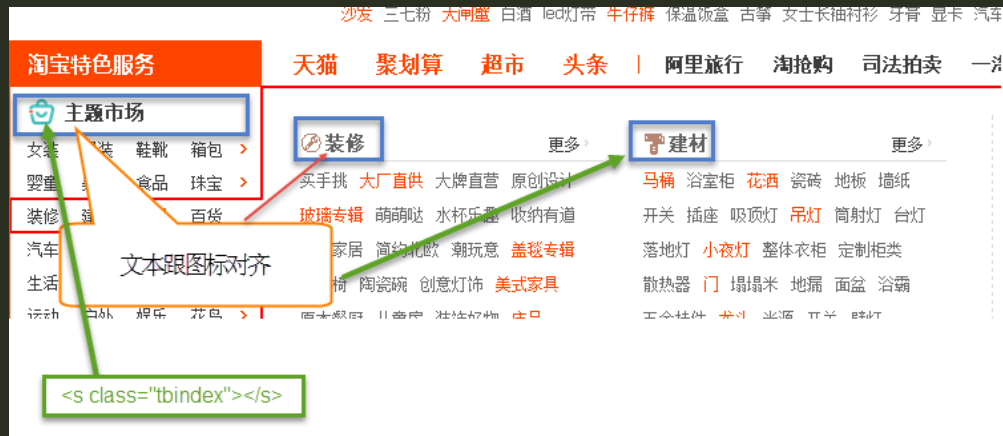
腾讯横向精灵图



新浪竖向精灵图

# 小图跟文字进行水平对齐技术

- 第一种方式：通过调整文字背景图的位置。  
通过padding+背景图的position配合产生对齐效果。  
注意精灵图（长条适合）
- 第二种方式：将背景图设置到一个s标签上，让s标签跟文本对齐。  
精灵图可以做的非常紧凑。  
关键点：文本的对齐到middle，可以通过设置图片的margin-top、left、right等控制图片的显示位置。
- 第三种方式：  
可以同浮动的形式实现

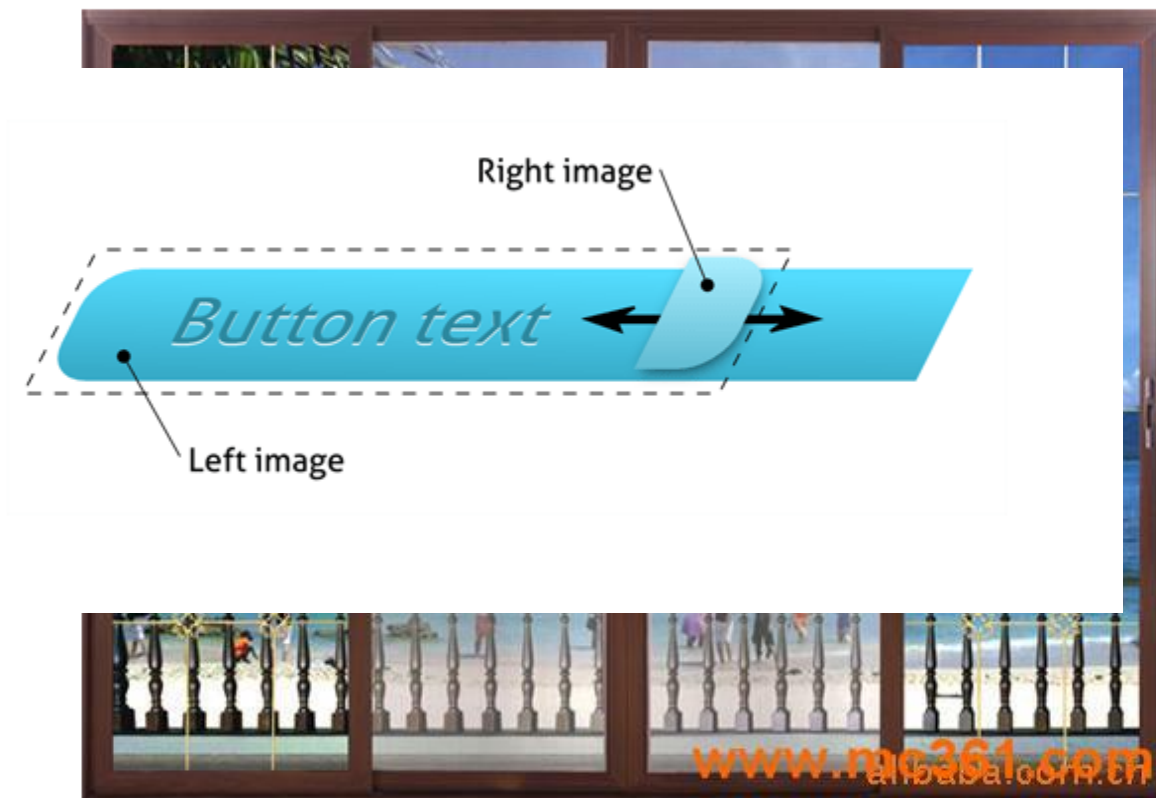


```
.arrow-d {  
    width: 9px;  
    height: 6px;  
    background:  
    url(../imgs/bg.fw.png) no-repeat  
    0 -80px;  
    display: inline-block;  
    vertical-align: middle;  
    margin-top: -2px;  
}
```

# 拓展学习

- 淘宝的小图，已经部分font-face在线字体化。
- 研究一下font-face？

- css滑动门技术



## • css滑动门技术

制作网页时，为了美观，常常需要为网页元素设置特殊形状的背景，如下图所示，即为设置了圆角背景的导航。



– 如果使用传统的CSS思想，要想为列表项“技术联盟”设置上图所示的圆角背景，可以对其所在的<li>标记应用大圆角背景。这种方式固然可以实现圆角背景的效果，但是，如果以后增加或减少列表项中的文字，就需要重新对<li>定义背景图像，以适应文本内容的多少。



- 需求分析

- 为了使各种特殊形状的背景能够自适应元素中文本内容的多少，出现了CSS滑动门技术。它从新的角度构建页面，使各种特殊形状的背景能够自由拉伸滑动，以适应元素内部的文本内容，可用性更强。

## • 阶段案例

通过前面两个小节的学习，相信初学者已经对滑动门技术有了一定的认识。为了加深初学者对滑动门技术的理解，下面将应用滑动门技术制作一个梯形导航，其效果如下图所示。



梯形导航

## • margin负值运用

制作网页时，为了拉开元素之间的距离，常常给元素设置大于0的外边距margin。其实，在实际工作中，为了实现一些特殊的效果，经常需要将元素的margin设置为负值，如下图所示的元素重叠效果。



元素的重叠效果

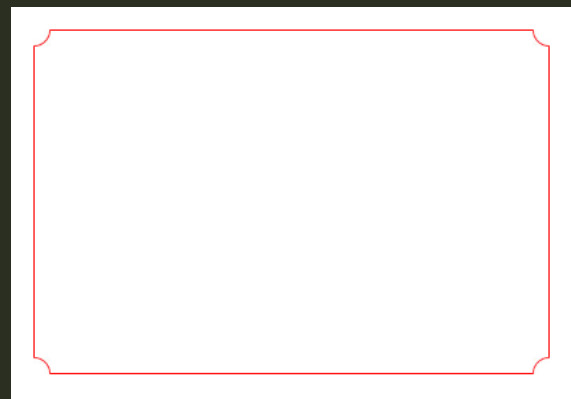
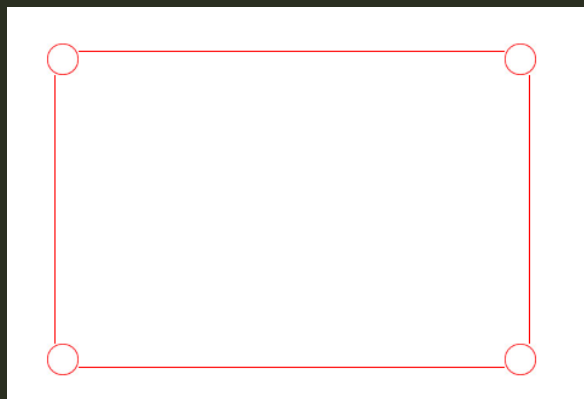
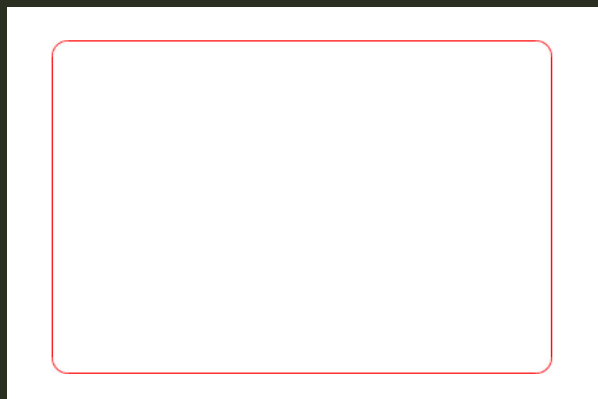
## 消除inline-block中的空隙

行内块之间会有缝隙，去掉的方法

1. 去除空格，把代码放在一行上。
2. 使用margin负值。
3. 给父级添加font-size:0 ;
4. 使用letter-spacing或者 word-spacing
- 5、使用float的方式

## CSS圆角

图片简单，但是写起来要复杂多。但是也有自己意外的一部分。



## CSS三角

用CSS实现 三角

背景图实现三角

在线字体三角

文字三角

## ico图标

1. [www.bitbug.net](http://www.bitbug.net) 进行转换 生成的是：favicon.ico 的小图标
2. `<link rel="shortcut icon" href="favicon.ico" />`



## 导入式补充:

导入式与链入式相同，都是针对外部样式表文件的。对HTML头部文档应用style标记，并在<style>标记内的开头处使用@import语句，即可导入外部样式表文件。其基本语法格式如下：

```
<style type="text/css" >  
@import url(css文件路径);或 @import "css文件路径";  
    /* 在此还可以存放其他CSS样式*/  
</style>
```

需要位于其他内嵌样式的上面。



## 样式表总结

样式表	优点	缺点	使用情况	控制范围
行内样式表	书写方便	没有实现样式和结构相分离	极少	控制一个标签
内嵌样式表	结构和样式相分离	没有彻底分离	较多	控制一个页面
外部样式表	完全实现结构和样式相分离	需要引入	最多	控制整个站点

# 关于表格

```
.my-table {  
  border: 1px solid #ccc;  
}  
.my-table td {  
  border: 1px solid blue;  
}
```



first row	first row
second row	second row

```
.my-table {  
  border: 1px solid #ccc;  
  border-collapse: collapse;  
}  
.my-table td {  
  border: 1px solid blue;  
}
```

first row	first row
second row	second row

**chrome & FF**  
**table: 2px**  
**td: 1px**

**border-collapse: collapse;**

/\* 可以让表格和 td盒子的边框进行合并，外显示表格的，td内部显示td的边框\*/

# CSS的选择器解析顺序

- CSS解析顺序：从最右选择器进行筛选，然后依次向左进行过滤。
- #aside div .tit { color: red; }
- 先解析：.tit进行过滤，然后过滤 div，然后最后过滤 #aside元素
- 选择器层级嵌套不要超过3层。

# CSS选择器优化效率

CSS选择器的性能排序：

- 1.id选择器 ( #myid )
- 2.类选择器 ( .myclassname )
- 3.标签选择器 ( div,h1,p )
- 4.相邻选择器 ( h1+p )
- 5.子选择器 ( ul < li )
- 6.后代选择器 ( li a )
- 7.通配符选择器 ( \* )
- 8.属性选择器 ( a[rel="external"] )
- 9.伪类选择器 ( a:hover,li:nth-child )

# CSS编写规范

- 文件规范
- 注释规范
- 命名规范
- 书写规范
- 测试规范等
- 参考地址：<http://www.w3cfuns.com/article-5596764-1-1.html>

# 有思想的CSS

- OOCSS
- ACSS
- BEM

# 综合案例

- 梅兰商贸案例
- PSD→HTML+CSS