## CSS 基础语法

目录

**第一章 CSS简介** 3

[1、CSS的选择器 3](#_Toc13811)

[2、字体系列 6](#_Toc1699)

[3、CSS文本属性 7](#_Toc5690)

**第二章 CSS的样式** 9

[1、内部样式 9](#_Toc3316)

[2、外部样式 10](#_Toc28503)

[3、行内样式表 10](#_Toc13366)

**[第三章 chrome调试工具以及快速书写代码的方法 10](#_Toc13426)**

[1、chrome调试工具 10](#_Toc20071)

[2、快速生成HTML结构的语法 11](#_Toc23295)

[3、快速格式化代码 12](#_Toc26238)

**第四章 多种选择器**

[1、复合选择器 14](#_Toc11804)

[2、后代选择器 14](#_Toc24351)

[3、子元素选择器 15](#_Toc8687)

[4、并集选择器 16](#_Toc20385)

[5、伪类选择器 17](#_Toc15638)

[6、链接伪类 17](#_Toc17989)

[7、focus伪类选择器 18](#_Toc22145)

[8、元素的显示模式 18](#_Toc293)

**第五章 块元素与行内元素、工具** 18

1、块元素 19

2、行内元素 19

[3、行内块元素 19](#_Toc4897)

[4、元素显示模式的转换 19](#_Toc31655)

[5、一个小工具的使用snipaste 20](#_Toc10773)

[6、单行文字垂直居中 20](#_Toc23672)

**[第六章 背景颜色与图片 21](#_Toc27452)**

[1、背景颜色 21](#_Toc28755)

[2、背景图片 21](#_Toc9045)

[3、背景平铺 21](#_Toc41)

[4、背景位置--方位名词 22](#_Toc1381)

[5、背景固定 22](#_Toc19850)

[6、背景的复合属性 22](#_Toc25885)

**[第七章 CSS三大特性 23](#_Toc10844)**

[1、层叠性：解决样式冲突问题 23](#_Toc4460)

[2、继承性 23](#_Toc26570)

[3、 优先级 24](#_Toc29237)

**[第八章 盒子 25](#_Toc22996)**

[1、盒子模型 25](#_Toc16388)

[2、盒子模型的组成 26](#_Toc24892)

[3、盒子边框 26](#_Toc16753)

[4、表格细线边框 26](#_Toc22440)

[5、边框会影响盒子的实际大小 27](#_Toc10543)

[6、内边距：盒子内容与边框的距离 27](#_Toc3588)

[7、外边距 27](#_Toc20855)

[8、行内元素和行内块元素水平居中 27](#_Toc30391)

[9、外边距合并 27](#_Toc10207)

[10、清除内外边距 28](#_Toc8864)

[11、PS的基本操作 28](#_Toc29921)

[12、 新知识 28](#_Toc12035)

[13、圆角边框原理：盒子变圆角 29](#_Toc17563)

[13、 盒子阴影 29](#_Toc29640)

[14、 文字阴影 30](#_Toc8849)

**[第九章 浮动 30](#_Toc31927)**

[1、浮动简介 30](#_Toc10357)

[2、浮动特性 31](#_Toc28057)

[3、清除浮动 33](#_Toc8260)

[4、常见图片格式 35](#_Toc12393)

[5、PS切图 35](#_Toc15420)

**第十章 CSS书写顺序** 35

**第十一章 定位** 35

[1、定位简介 36](#_Toc1523)

[2、定位类型 36](#_Toc15210)

[3、CSS精灵技术（CSS Spirtes/CSS 雪碧） 41](#_Toc5276)

[4、字体图标 41](#_Toc18418)

**[第十二章 CSS三角、鼠标样式与解决文字图片出现的问题 43](#_Toc11808)**

[1、CSS三角制作：将宽度高度设置为0。 43](#_Toc1078)

[2、用户的鼠标样式cursor 43](#_Toc3209)

[3、取消表单轮廓和防止拖拽文本域resize 44](#_Toc9856)

[5、 解决图片底侧默认空白缝隙问题 45](#_Toc1974)

6、单行文字溢出省略号显示 45

[7、 多行文字溢出省略号显示 45](#_Toc16113)

[8、 布局技巧---margin负值运用 45](#_Toc4127)

[9、 文字围绕浮动元素 46](#_Toc25001)

[10、 行内块的巧妙运用 47](#_Toc24766)

[11、 CSS初始化 47](#_Toc32659)

**[第十三章 html新特性 47](#_Toc10568)**

[1、语义化标签 47](#_Toc19005)

[2、HTML视频标签 48](#_Toc15125)

[3、音频标签 49](#_Toc9108)

**[第十四章 CSS3新特性 51](#_Toc22126)**

[1、新增选择器 51](#_Toc24719)

[2、伪元素清除浮动 54](#_Toc25363)

**[第十五章 CSS3盒子与图片处理 55](#_Toc7680)**

[1、CSS3盒子模型 55](#_Toc30270)

[2、CSS3图片模糊处理 55](#_Toc8488)

[3、计算盒子宽度calc函数 56](#_Toc22891)

[4、 CSS3过渡 56](#_Toc18890)

**[第十六章 品优购项目 58](#_Toc8363)**

[1、logo优化 58](#_Toc2493)

2、搜索框 58

**[第十七章 服务器 58](#_Toc5202)**

**[第十八章 2D转换（在二维坐标系进行改变） 58](#_Toc13922)**

[1、移动位置的语法：定位、外边距、移动 58](#_Toc4406)

2、translate最大优点 ...............................................................................................................59

**[第十九章 CSS动画（animation） 60](#_Toc27231)**

[1、用 keyframes定义动画 60](#_Toc23524)

[2、使用动画 60](#_Toc8841)

3、动画常见属性 60

4、动画简写 61

[5、速度曲线细节 61](#_Toc1645)

**[第二十章 3D转换 62](#_Toc22258)**

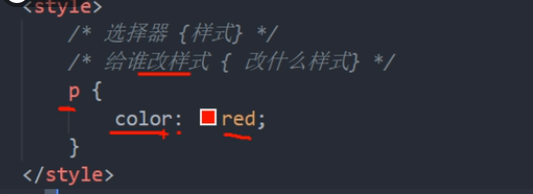
[1、translate3d(x,y,z) 62](#_Toc17859)

2、perspective透视 62

[3、3D旋转rotate3d 63](#_Toc6501)

### 第一章 CSS简介

# CSS 用于控制网页的样式和布局，CSS3 是最新的 CSS 标准。



<style>

P{

Color:red;

Font-size:12px;#文字像素

}

</style>

CSS选择器

标签名{

Color:\*\*;

Font-size:\*\*;

}

**1、CSS的选择器**

1. 标签选择器:使页面内某一类标签变成同一样式
2. 类选择器：若想要异化选择不同的标签，单独选一个或者几个标签，可以使用类选择器；font-size:\*\*px 字体大小

<style>

.suibian{

Color:red;

}

</style>

<li class=”suibian”>突然</li>

如需选择拥有特定 class 的元素，请写一个句点（.）字符，后面跟类名,不要以数字或者中文为类名。可以一个类名被多次使用，可以定义多个名。

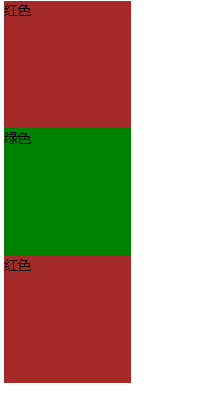
在此例中，所有带有 class="center" 的 HTML 元素将为红色且居中对齐：

.center {

text-align: center;

color: red;}

示例：



类选择器：多类名,用空格键分开





1. id选择器

可以为标有特定id的HTML元素指定特定的样式,只能调用一次。

HTML元素以id属性来设置id选择器，CSS中id选择器以”#”来定义,

语法：

#id名{

属性1：属性值1；

........

}

#nav{

Color:red;

}



（4）通配符选择器

通配符选择器使用“\*”定义，为所有标签变为定义的格式：

\*{

属性1：属性值1；

....

}

通配符选择器不需要被调用，自动给所有元素使用样式；

**2、字体系列**

（1）Font-family定义字体系列，是微体软黑或者宋体等，可以用中文或者英文。



（2）Font-size字体大小：px大小是我们网页的最常用的单位

谷歌浏览器默认大小为16px。

（3）Font-weight:设置文本粗细，可以用数字或者字母表示粗细，一般开发使用数字。

|  |  |
| --- | --- |
| **值** | **描述** |
| normal | 默认值。定义标准的字符。 |
| bold | 定义粗体字符。 |
| bolder | 定义更粗的字符。 |
| lighter | 定义更细的字符。 |

Eg:

font-weight=bold; #加粗

font-weight=700 #加粗

1. font-style字体样式

|  |  |
| --- | --- |
| **值** | **描述** |
| normal | 默认值。浏览器显示一个标准的字体样式。 |
| italic | 浏览器会显示一个斜体的字体样式。 |
| oblique | 浏览器会显示一个倾斜的字体样式。 |
| inherit | 规定应该从父元素继承字体样式。 |

（5）字体的复合属性

将文字样式综合来写



使用font属性时，必须按照顺序来写，不能更换顺序，以空格隔开，不需要设置的属性可以省略，但是必须保留font-size和font-family属性，否则font属性不起作用。

**3、CSS文本属性**

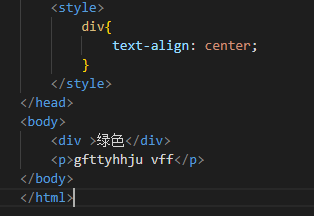
（1）文本颜色：color定义

颜色的表示：①十六进制：#ff0000/#cc00ff，实际操作只需要吸即可

②字母表示：green/pink/red/blue

③rgb代码：rgb(255,0,0)/rgb(255,0,255)

（2）对齐文本:text-align（设置水平对齐方式）让标签盒子里面的文字水平对齐。





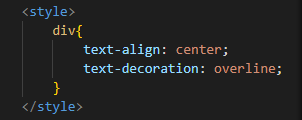
## 对齐文本可能的值

|  |  |
| --- | --- |
| **值** | **描述** |
| left | 把文本排列到左边。默认值：由浏览器决定。 |
| right | 把文本排列到右边。 |
| center | 把文本排列到中间。 |
| justify | 实现两端对齐文本效果。 |
| inherit | 规定应该从父元素继承 text-align 属性的值。 |

（3）装饰文本

Text-decoration规定添加文本装饰，如下划线、删除线、上划线

|  |  |
| --- | --- |
| **值** | **描述** |
| none | 默认。定义标准的文本。 |
| underline | 定义文本下的一条线。 |
| overline | 定义文本上的一条线。 |
| line-through | 定义穿过文本下的一条线。 |
| blink | 定义闪烁的文本。 |
| inherit | 规定应该从父元素继承 text-decoration 属性的值。 |



（4）文本缩进

Text-indent：属性用于指定文本的第一行缩进，通常是将段落的首行缩进。

PX：缩进px个长度

Em：是一个相对单位，就是当前元素1个文字的大小，若当前元素没有设置大小，就按照父元素一个文字大小。

（5）行间距

Line-height:属性用于设置行间的距离，可以控制文字行与行之间的距离。\*\*px

|  |  |
| --- | --- |
| **值** | **描述** |
| normal | 默认。设置合理的行间距。 |
| *number* | 设置数字，此数字会与当前的字体尺寸相乘来设置行间距。 |
| *length* | 设置固定的行间距。 |
| *%* | 基于当前字体尺寸的百分比行间距。 |
| inherit | 规定应该从父元素继承 line-height 属性的值。 |

### CSS的样式

CSS的三种样式表：（1）行内样式表（2）内部样式表（3）外部样式表

**1、内部样式**

Html页面标签



一般将style放在head里面，但是置于body同样可以运行，可以控制整个html页面的元素样式设置

**2、外部样式**

核心是样式单独写到CSS文件中，之后把CSS文件写入HTML页面中使用。

步骤：

新建一个后缀名为.css的样式文件，把所有CSS代码都放入此文件中，使用link标签引入这个文件<link rel=”stylesheet” href=”css文件名”>。

**3、行内样式表**

<div style=”color:red;font-size:12px;”>在某一行标签里面命名。

动作独家冠名等创新形式，在世界杯赛事直播中长时间、多频次呈现合作品牌元素，强化企业品牌传播效果，同时在下半场比赛开始前播出合作品牌的15秒广告片。传奇系列产品还为合作品牌定制了《我爱世界杯》融媒体独家冠名、独家特约项目，以及总台新媒体节。

1. **chrome调试工具以及快速书写代码的方法**

**1、chrome调试工具**

1、Ctrl+滚轮：放大缩小代码或者界面

2、左边是html，右边是CSS

3、右边CSS可以改动数值或者查看颜色

4、Ctrl+0复原浏览器大小

Emmet语法：提高HTML与CSS书写速度

**2、快速生成HTML结构的语法**

（1）生成标签，直接输入标签名，按tab键

（2）生成多个相同标签，加上\*，按tab键。eg:div\*3，生成3个div标签

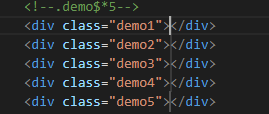
（3）若有父子级关系的标签，可以用>,tab键。比如ul>li就可以了

（4）若有兄弟关系的标签，用+就可以，比如div+p

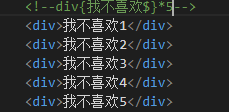
（5）若生成带有类名或者id名的标签，直接写.demo或者#two 按下tab键就可以。Eg:ul>li#sdfg



（6）如果生成的div类名有顺序，可以用自增符号$

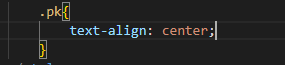


（7）如果想要在生成的标签内部写内容可以用{}表示

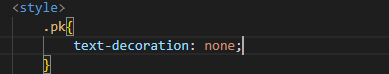


（8）采取首字母简写按tab键即可

Eg:tac



Tdnone



W100



**3、快速格式化代码**

快捷键：shift+alt+f

也可以设置当我们保存页面的时候自动格式化代码

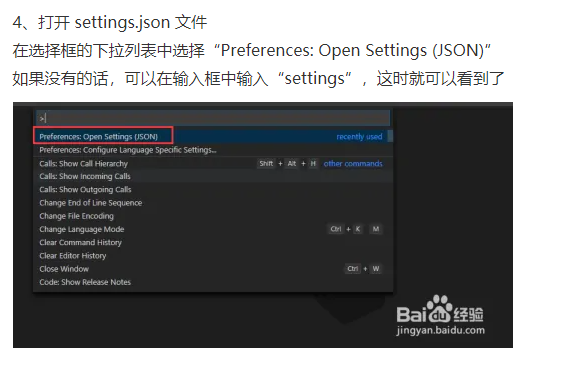
1. 文件--->【首选项】--->【设置】
2. 搜索emmet.include
3. 在settings.json下的用户添加以下语句
4. “editor.formatOnType”:true

“editor.formatOnSave”:true

1. 只需要设置一次即可，以后都可以自动保存格式化代码,每次按ctrl+s保存代码就会自动调整格式。

打开setting.json的方式：





### 第四章 多种选择器

**1、复合选择器**

由两个或多个基础选择器，通过不同的方式组合而成

常用的复合选择器包括：后代选择器、子选择器、并集选择器、伪类选择器等

**2、后代选择器**

又称包含选择器，可以选择父元素里面的子元素，其写法就是把外层标签写在前面，内层标签写在后面，中间空格分开。

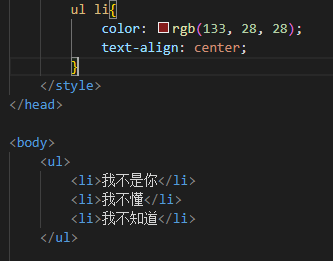
语法：元素1 元素2{样式声明}

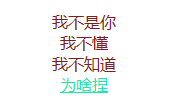
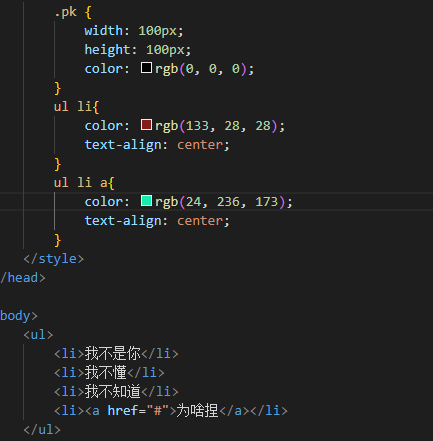
表示选择元素1里面的所有元素2（后代元素）

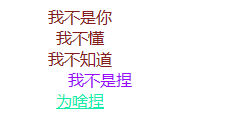
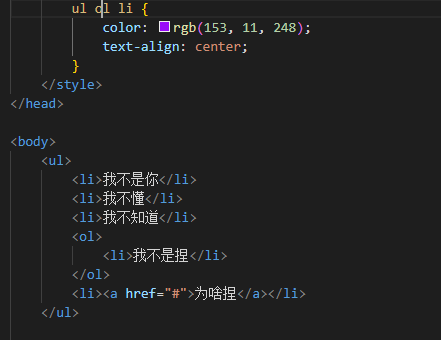
元素1与元素2需要空格隔开

最终改变的是元素2的样式

层层嵌套，可以一层一层往下写，eg:ul li a







**3、子元素选择器**

只能选择作为元素的最近一级子元素，简单理解为亲儿子元素

语法：元素1 >元素2{样式声明}

表示选择元素1里面的所有直接后代子元素2。

元素2必须是亲儿子元素，不可以是亲孙子或者重孙子元素



**4、并集选择器**

可以选择多组标签，为他们定义相同的样式，通常用于集体声明。

通过英文逗号链接而成，任何形式的选择器都可以作为并集选择器的一部分。

语法：元素1，元素2{样式声明}

可以使用后代选择器作为并集选择器的一部分，可以竖着写





**5、伪类选择器**

用于向某些选择器添加特殊效果，比如给链接添加特殊效果，或者选择第一个或者第n个元素。

伪类选择器最大特点是用冒号表示，eg: :hower、：first-child

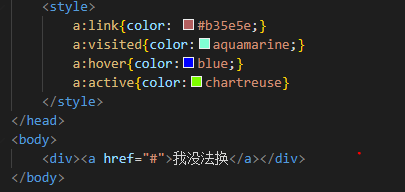
**6、链接伪类**

a:link 选择所有未被访问过的链接

a:visited 选择所有已经被访问过的链接

a:hover 选择鼠标指针位于其上的链接

a:active 选择活动链接（鼠标按下未弹起的链接）

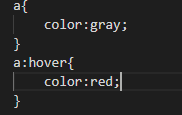


注意事项：

（1）为了确保生效，尽量按照LVHA顺序声明，link,visited,hover,active

（2）需要单独为a 指定样式

实际开发时的写法：



**7、focus伪类选择器**

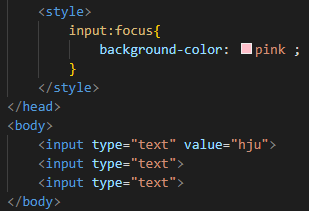
用于选取获得焦点的表单元素

焦点就是光标，一般情况<input>类表单元素才能获取，鼠标放在哪里就可以变颜色

Input:focus{

Background-color:yellow;

}





**8、元素的显示模式**

元素的显示模式就是元素以什么方式进行显示，比如div独占一行，一行可以放置多个span。

HTML元素一般分为块元素和行内元素两种类型。

### 第五章 块元素与行内元素、工具

### **块元素**

常见的块元素有<h1>~<h6>/<p>/<div>/<ol>/<li>/<ul>等

特点：（1）自己独占一行

（2）高度宽度以及外边距、内边距可以自己设置

（3）宽度默认是容器的100%

（4）是一个容器及盒子，里面可以放行内或者块元素

注意：（1）文字类的元素不能使用块级元素

（2）<p>标签主要用于存放文字，<p>不能存放块元素

（3）<h1>~<h6>等文字类块级标签不能存在其他块元素

### **2、行内元素**

常见的行内元素由：<a>/<strong>/<b>/<em>/<i>/<s>/<del>/<ins>/<u>/<span>

特点：（1）相邻行内元素在一行上，一行可以显示多个。



（2）高、宽直接设置是无效的。

（3）默认宽度就是它本身内容的宽度。

（4）行内元素只能容纳文本或其他行内元素。

注意：

1）链接里面不能再放链接。

2）特殊情况链接<a>里面可以放块级元素，但是给<a>转换一下块级模式最安全。

**3、行内块元素**

行内元素的几个特殊标签：<img/>、<input/>、<td>同时具有块元素和行内元素

特点：

（1）和相邻行内元素在一行上，但是他们之间的会有空白缝隙，一行可以显示多个

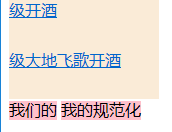
（2）默认宽度就是它本身内容的宽度。

（3）高度宽度以及外边距、内边距可以自己设置。

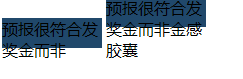
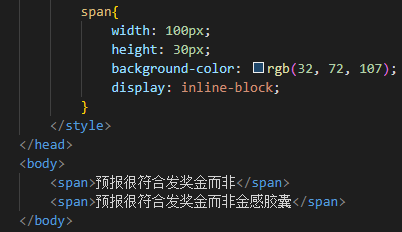
**4、元素显示模式的转换**

转换为块元素: display:block

转换为行元素：display:inline



转换为行内块元素：display:inline-block



**5、一个小工具的使用snipaste**

（1）按F1可以截图，同时测量大小，设置箭头、书写文字等

（2）F3在桌面置顶显示

（3）点击图片可以取色（按下shift可以切换取色模式）

（4）按下esc取消图片显示

**6、单行文字垂直居中**

方法：让文字的行高等于盒子高度

**第六章 背景颜色与图片**

**1、背景颜色**

Background-color：transparent(透明的）/任何一种颜色

**2、背景图片**

Background-image属性描述了元素的背景图像，实际开发常见于logo或者一些装饰性的图片中，或者超大的背景图片，优点是便于控制位置。

语法：background-image:url() #url放置图片地址。



**3、背景平铺**

Background-repeat：默认下是平铺的

## 语法：background-repeat: no-repeat（指定只显示一次背景图像）

|  |  |
| --- | --- |
| **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。 |

页面元素既可以添加背景图片和背景颜色，只不过背景图片会压住背景颜色

**4、背景位置--方位名词**

Background-position改变图片在背景的位置

Background-position:x y;(x坐标和y坐标，可以使用方位名词或者精确单位）

1、参数值：length:百分数，由浮点数字和单位标识符组成的长度值。

|  |  |
| --- | --- |
| **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) 或任何其他的 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。 |

(1)若x,y都是方位名词，则方位词前后顺序表达结果相同。Eg:top left和left top。

(2)若只指定了一个方位名词，第二个参数省略，则第二个默认居中对齐。

2、参数是精确单位

（1）如果参数是精确单位，则第一个是x坐标，第二个是y坐标（px）。

（2）若只指定一个数值，那该数值一定是x轴坐标。

3、参数是混合单位

（1）如果指定两个参数是精确单位和方位名词混合，则第一个值是x坐标，第二个值是y坐标。

**5、背景固定**

Background-attachment：scroll | fixed

|  |  |
| --- | --- |
| **值** | **描述** |
| scroll | 默认值。背景图像会随着页面其余部分的滚动而移动。 |
| fixed | 当页面的其余部分滚动时，背景图像不会移动。 |
| inherit | 规定应该从父元素继承 background-attachment 属性的设置。 |

**6、背景的复合属性**

Background：背景颜色，背景图片地址，背景平铺，背景图像滚动，背景图片位置

背景色半透明：指盒子背景半透明，盒子内容不影响。

Background：rgba(0,0,0,0.3)

最后一个参数是alpha透明度，范围在0-1之间，一般可以省略0，写成.3、.8等。

**第七章 CSS三大特性**

**1、层叠性：解决样式冲突问题**

原则：

（1）样式冲突（样式相同，选择器相同）,遵循就近原则，哪个样式离结构近，就执行哪个样式

（2）样式不冲突，不会层叠



**2、继承性**

（1）子标签会继承父标签的某些样式，如文本颜色和字号，降低复杂性。



（2）子元素可以继承父元素的样式（text/font-/line-这些元素开头可以继承，以及color属性）

（3）行高的继承：line-height，行高可以跟单位，也可以不跟单位

如果子元素没有设置行高，则会继承父元素的行高为1.5

此时子元素行高为：当前文字大小\*1.5



**3、 优先级**

选择器相同，则执行层叠性；

选择器不同，则根据选择器权重。

|  |  |
| --- | --- |
| 继承或者\*  优先级逐渐增高 | 0 0 0 0 |
| 元素选择器div{} | 0 0 0 1 |
| 类选择器，伪类选择器 | 0 0 1 0 |
| ID选择器 | 0 1 0 0 |
| 行内样式style=”” | 1 0 0 0 |
| !important | 无穷大 |

Eg:div{

Color:pink!important;

}

则优先执行div语句的样式。

注意：

（1）权重不会有进位

（2）id选择器永远大于类选择器，简记为通配符和继承权重为0，标签选择器为1，类选择器为10，id选择器为100，行内样式表为1000，！Important为无穷大。

（3）继承的权重为0，如果该元素没有直接选中，不管父元素权重多高，子元素权重得到的权重为0.



（4）权重的叠加

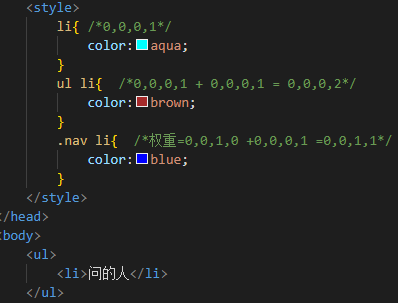
如果是复合选择器，就会有权重叠加问题，不会有进位.

Div ul li ------>0,0,0,3 （0,0,0,1+0,0,0,1+0,0,0,1）

.nav ul li ------->0,0,1,2 (0,0,1,0+0,0,0,1+0,0,0,1)

A:hover ------->0,0,1,1 (0,0,0,1+0,0,1,0)

.nav a ----->0,0,1,1 (0,0,1,0+0,0,0,1)



**第八章 盒子**

**1、盒子模型**

网页布局过程：

（1）先准备好相关的网页元素，网页元素基本都是盒子

（2）利用CSS设置盒子样式，然后摆放到相应位置

（3）往盒子里添加内容

**2、盒子模型的组成**

边框(border)、内容(content)、内边距(padding)、外边距(margin)

**3、盒子边框**

Border 可以设置边框宽度、边框样式、边框颜色

|  |  |
| --- | --- |
| **值** | **描述** |
| *border-width* | 规定边框的宽度px。参阅：[border-width](https://www.w3school.com.cn/cssref/pr_border-width.asp" \o "CSS border-width 属性) 中可能的值。 |
| *border-style* | 规定边框的样式。参阅：[border-style](https://www.w3school.com.cn/cssref/pr_border-style.asp" \o "CSS border-style 属性) 中可能的值。 |
| *border-color* | 规定边框的颜色。参阅：[border-color](https://www.w3school.com.cn/cssref/pr_border-color.asp" \o "CSS border-color 属性) 中可能的值。 |
| inherit | 规定应该从父元素继承 border 属性的设置 |

Border-style：solid（实线边框）、dotted(点线边框)、dashed(虚线边框)

复合属性：三个顺序可以任意摆放

Eg:

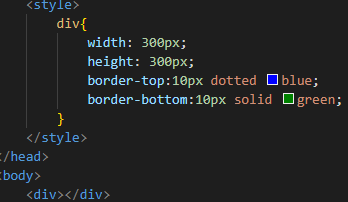
border:1px solid red

Border-top:上边框

Border-bottom：下边框

Border-left:左边框

Border-right:右边框



**4、表格细线边框**

Border-collapse：控制浏览器绘制边框的方式，他控制相邻单元格的边框。

|  |  |
| --- | --- |
| **值** | **描述** |
| separate | 默认值。边框会被分开。不会忽略 border-spacing 和 empty-cells 属性。 |
| collapse | 如果可能，边框会合并为一个单一的边框。忽略 border-spacing 和 empty-cells 属性。 |
| inherit | 规定应该从父元素继承 border-collapse 属性的值。 |

**5、边框会影响盒子的实际大小**

边框会增加盒子的实际大小，因此我们测量盒子大小时，不量边框

如果测量的时候包含了边框，则需要width/height减去边框的宽度。

**6、内边距：盒子内容与边框的距离**

Padding-top/bottom/left/right:上/下/左/右内边距

复合属性：padding:5px（上下左右都有5像素的内边距）

padding:5px 10px（上下都有5像素的内边距,左右是10像素内边距）

padding:5px 10px 20px（上内边距为5px,左右为10px,下为20px）

padding:5px 10px 20px 30px（上右左下）顺时针

注意：

（1）padding会影响盒子的实际大小，若一个盒子已经定义了高度宽度，再定义padding时会撑大盒子。

（2）如果盒子不指定宽度或者高度，padding不会撑大盒子。

**7、外边距**

Margin：设置外边距，即盒子与盒子的距离

Margin-left/right/top/bottom：复合属性，与padding一致

应用：外边距可以让盒子水平居中，但须满足两个条件：

（1）盒子必须指定了宽度（width）

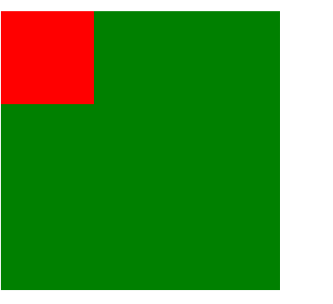
（2）盒子左右外边距都设置auto;eg: margin: 0 auto

**8、行内元素和行内块元素水平居中**

方法：为父元素添加text-algin:center

**9、外边距合并**

嵌套元素垂直外边距的塌陷：对于两个嵌套关系的块元素，父元素有上外边距同时子元素也有上外边距，此时父元素会塌陷较大的外边距值。



解决方案：

(1)可以为父元素定义上边框

(2)可以为父元素定义上内边距

(3)可以为父元素添加overflow:hidden

**10、清除内外边距**

即清除默认样式的内外边距

语法： \*{

Padding:0;

Margin:0;

}

行内元素为了照顾兼容性，尽量只设置左右内外边距，不设置上下内外边距，但是转换为块元素或者行内块元素就可以了。

**11、PS的基本操作**

Ctrl+R：打开标尺

Ctrl+（加号+/减号-）：放大/缩小视图

按住空格键，鼠标变小手，拖动PS视图，选区操作，可以测量大小，Ctrl+D可以取消选区。

1. **新知识**

去掉li前面的项目符号（小圆点）

方法：List-style：none；

**13、圆角边框原理：盒子变圆角**

Border-radius:\*\*px或者\*\*%

Eg:

（1）如果是正方形，想要设置为圆形，把数值修改为高度或者宽度的一半，或者直接写为50%:即为宽度和高度的一半。

（2）如果是个矩形，设置为高度的一半，圆角矩形:设置为高度的一半。

（3）border-radius：4个参数，代表左上角，右上角，右下角，左下角

Border-radius：2个参数，代表左对角线和右对角线

分开写：

border-top-left-radius、border-top-right-radius、border-bottom-right-radius、border-bottom-left-radius

**14、盒子阴影**

不占空间，不影响其他盒子排列，不写outset

Box-shadow: h-shadow v-shadow blur spread color inset;

|  |  |
| --- | --- |
| **值** | **描述** |
| *h-shadow* | 必需。水平阴影的位置。允许负值。 |
| *v-shadow* | 必需。垂直阴影的位置。允许负值。 |
| *blur* | 可选。模糊距离。 |
| *spread* | 可选。阴影的尺寸。 |
| *color* | 可选。阴影的颜色。请参阅 CSS 颜色值。 |
| inset | 可选。将外部阴影 (outset) 改为内部阴影。 |



注意：任何标签都可以使用：hover

**15、文字阴影**

Text-shadow : h-shadow v-shadow blur color

|  |  |
| --- | --- |
| **值** | **描述** |
| *h-shadow* | 必需。水平阴影的位置。允许负值。 |
| *v-shadow* | 必需。垂直阴影的位置。允许负值。 |
| *blur* | 可选。模糊的距离。 |
| *color* | 可选。阴影的颜色。参阅 [CSS 颜色值](https://www.w3school.com.cn/cssref/css_colors_legal.asp" \o "CSS 合法颜色值)。 |

**第八章 浮动**

**1、浮动简介**

改变元素标签默认的排列方式

网页布局的准则：

（1）多个块级元素纵向排列找标准流、多个块级元素横向排列找浮动。

语法： 选择器{

Float : 属性值

}

|  |  |
| --- | --- |
| **值** | **描述** |
| left | 元素向左浮动。 |
| right | 元素向右浮动。 |
| none | 默认值。元素不浮动，并会显示在其在文本中出现的位置。 |
| inherit | 规定应该从父元素继承 float 属性的值。 |

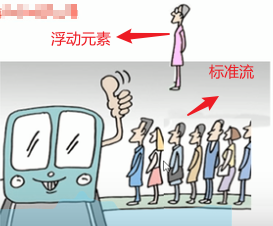
（2）float属性用于创建浮动框，将其移到一边，直到左边缘或右边缘触及包含块或者另一个浮动框的边缘。

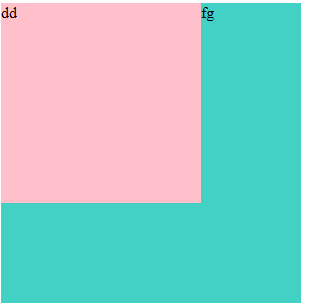


**2、浮动特性**

（1）脱离标准普通流的控制移动到指定位置，即脱标。

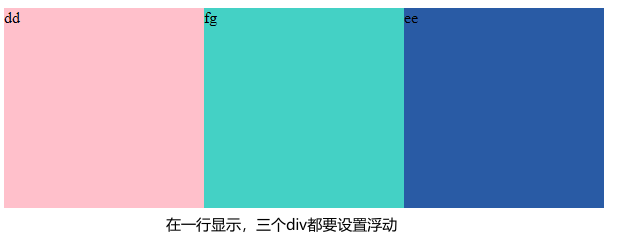
（2）浮动盒子不再保留原先的位置





若两个盒子中有一个发生浮动，则标准流的盒子会在浮动盒子浮动后占领原先浮动盒子位置，形成叠加效果。

（3）如果多个盒子都设置了浮动，则他们会按照属性值一行内显示并且顶端对齐排列。

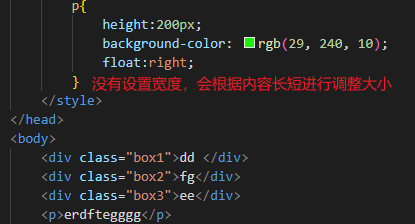


（4）浮动的元素是互相贴在一起的，不会有缝隙，如果父级元素宽度装不下这些浮动盒子，多出的盒子会另起一行。

（5）浮动元素都具有行内块元素特性，任何元素都可以浮动，不管原先是什么模式的元素，添加浮动之后具有行内块元素特性。（span等）

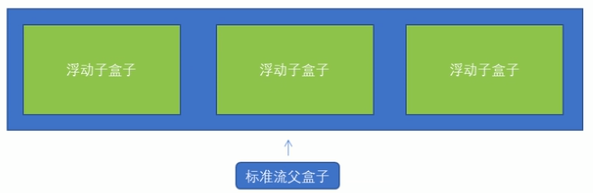
（6）如果行内元素添加了浮动，就不再需要添加行内块元素特性。

（7）如果块级元素没有添加宽度，默认宽度与父级元素一样宽，但是添加浮动后，它的大小根据内容来决定。



约束浮动元素位置的方法：

先用标准流的父元素排列上下位置，之后内部子元素采取浮动排列左右位置，符合网页布局的准则。

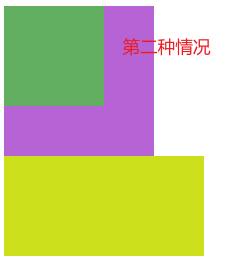
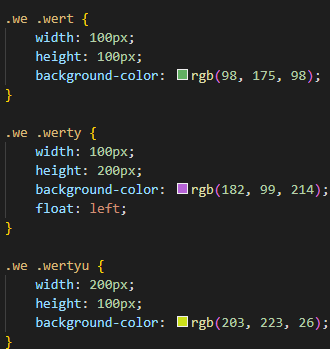


（8）一个元素浮动了，理论上其余的兄弟元素也要浮动。

（9）若有三个盒子，第一个和第三个不浮动，第二个浮动。则第一个盒子是标准流，会独占一行，第二个浮动会重启一行，第三个标准流也会在第二行的开头，叠加第二个元素。

（10）若有三个盒子，第一个和第三个浮动，第二个不浮动。则第二个盒子是标准流，会独占一行且叠加在第一个盒子后面，第三个会在第二行的开头重启一行。

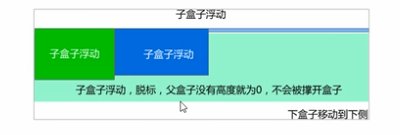




（11）一个盒子里面有多个子盒子，如果一个盒子浮动了，其他兄弟也应该浮动，以免引起问题。浮动的盒子只会影响浮动盒子后面的标准流，不会影响前面的标准流。

**3、清除浮动**

原因：由于父级盒子很多情况下，不方便给高度 ，但是子盒子浮动又不占位置，最后父级盒子高度为0时，就会影响下面的标准流的盒子。



本质：

（1）清除浮动元素造成的影响。

（2）如果父盒子本身有高度，则不需要清除浮动。清除浮动之后，父级就会根据浮动盒子自动检测高度，父级有了高度，就不会影响下面的标准流了。

语法：选择器{clear:属性值;}

|  |  |
| --- | --- |
| **值** | **描述** |
| left | 在左侧不允许浮动元素。 |
| right | 在右侧不允许浮动元素。 |
| both | 在左右两侧均不允许浮动元素。 |
| none | 默认值。允许浮动元素出现在两侧。 |
| inherit | 规定应该从父元素继承 clear 属性的值。 |
| 在实际工作中，几乎只会使用clear:both | |

策略：闭合浮动，只让浮动在父盒子内部影响，不影响父盒子外面的其他盒子。

方法：

（1）额外标签法也称隔墙法：在浮动元素末尾添加一个空标签，例如：<div style=”clear:both”> </div>，或者其他标签（如<br/>等）新增盒子必须是块级元素，不能是行内元素（span）。

优点：通俗易懂，书写方便

缺点：添加许多无意义的标签，结构比较差。

（2）父级添加overflow属性

为父元素添加overflow，将其属性设置为hidden/auto/scroll.三个属性都可以，一般用hidden。

父级添加:after伪元素：相当于在后面添加一个盒子

:after是额外标签法的升级版，也是父元素添加



复制和调用即可，为父元素添加这个类

优点：没有增加标签，结构简单

缺点：照顾低版本浏览器

（3）父级添加双伪元素：相当于在前后都添加盒子



复制和调用即可，为父元素添加这个类



**4、常见图片格式**

（1）Jpg格式：对色彩保留较好，高清，颜色较多，我们产品类图片经常使用jpg

（2）gif格式：最多只能储存256色，通常用来显示简单的图形和字体，但是可以保存透明背景和动画效果，实际经常用于一些图片小动画效果。

（3）Png格式：结合了jpg和gif的特点，具有存储形式丰富的特点，能够保持透明背景，如果想要切成背景透明图片，请选择png格式

（4）Psd格式：PS专用格式，可以存放图层、通道、遮罩等多种设计稿，可以直接复制文字，获得图片大小和测量距离。

**5、PS切图**

（1）图层切图：右击图层，快速导出为PNG

若一张图片由文字和图，想要带有图片的图，可以采用合并图层方法，先选中一个图层，按shift，再点击另一个图层，最后点击图层栏的合并图层（ctrl+e）即可。

（2）切片切图

步骤：利用切片工具手动划出，导出选中图片（文件菜单-->导出-->存储为web设备所用格式-->选择我们要的图片格式-->存储-->选中的切片）

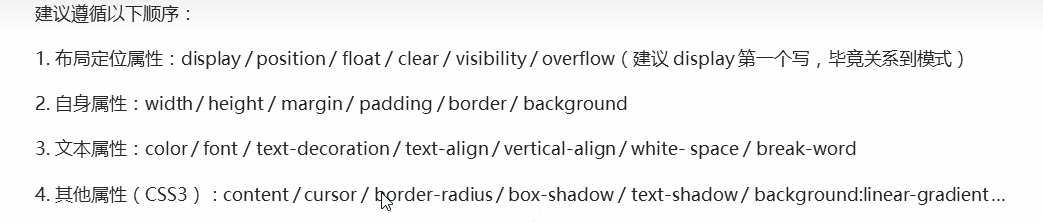
可以采用透明背景切图，把背景的眼睛点没，就不会带有白色的效果图。



（3）PS插件切图:Cutterman(http://www.cutterman.cn/zh/cutterman)



**第九章 CSS书写顺序**



1. **定位**

**1、定位简介**

（1）定位可以使盒子自由地在某个盒子内移动位置或者固定屏幕中某个位置，并且可以压住其他盒子。

（2）浮动可以让多个块级盒子一行没有缝隙排列显示，经常用于横向排列盒子。

（3）组成：定位模式+边偏移。

定位模式：指定一个元素在文档中的定位方式，通过CSS的position属性来设置，其值可以分为四个：

|  |  |
| --- | --- |
| absolute | 生成绝对定位的元素，相对 static 定位以外的第一个父元素进行定位。  元素位置通过 "left", "top", "right" 以及 "bottom" 属性进行规定。 |
| fixed | 生成固定定位的元素，相对于浏览器窗口进行定位。  元素位置通过 "left", "top", "right" 以及 "bottom" 属性进行规定。 |
| relative | 生成相对定位的元素，相对于其正常位置进行定位。  因此，"left:20" 会向元素的 LEFT 位置添加 20 像素。 |
| static | 默认值。没有定位，元素出现在正常的流中（忽略 top, bottom, left, right 或者 z-index 声明）。 |
| inherit | 规定应该从父元素继承 position 属性的值。 |

边偏移：top、bottom、left、right+像素来表示，定义元素相对于其父元素上下左右边线的距离。

**2、定位类型**

(1) 静态定位：元素默认定位方式，无定位的意思。没有边偏移边偏移，静态定位在布局时很少使用。

语法：

选择器{

Position:static；

}

1. 相对定位relative：元素移动位置时，是相对于它原来的位置来说的。原来在标准流 的位置继续占有，后面盒子仍然以标准流的方式对待它。（其他盒子不脱标）

语法：

选择器{

Position:relative；

}



（3）绝对定位absolute：元素移动位置时，是相对于祖先元素来说的。（拼爹型）无父级元素则相对于浏览器来移动。

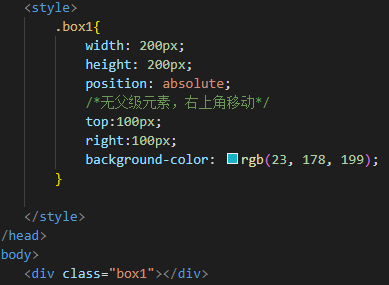
语法：

选择器{

Position:absolute

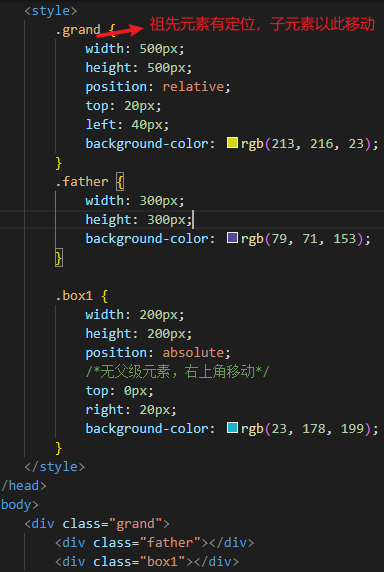
}

如果没有祖先元素或者祖先元素没有定位，则以浏览器为准定位。



如果祖先元素有定位（相对、绝对、固定定位），则以最近一级带有定位祖先元素为参考点移动位置。





绝对定位不再占有原来的位置。（脱标）

子绝父相:子级使用绝对定位，父级则需要相对定位

①子级绝对定位，不会占有位置，可以放到父盒子里面的任何一个地方，不会影响其他兄弟盒子。

②父盒子需要加定位限制子盒子在父盒子内显示。

③父盒子布局时，需要占有位置，因此父亲只能是相对定位。

（4）固定定位fixed：固定与浏览器可视区位置，可以在浏览器滚动页面时保持不动。

语法：

选择器{

Position:fixed

}

①以浏览器的可视窗口为参照点移动元素

②跟父元素没有任何关系

③不随滚动条滚动

④固定定位不占有原先位置，固定定位是脱标的，其实固定定位可以看作一种特殊的绝对定位。

注意：

固定定位：固定在版心右侧

算法：让固定定位盒子left:50%，走到可视区的一半位置；让盒子margin-left:版心宽度的一半距离。

（5）粘性定位sticky：可以认为相对定位和固定定位的混合，即滚动拉到某个定位才固定。兼容性差，IE不支持。

选择器{

Position:sticky;

}

以浏览器可视窗口为参照点移动元素，粘性定位占有原先位置，必须添加top、left、right、bottom元素其中一个才有效果。

定位叠放顺序z-index

选择器{

Z-index:1;

}

①数值可以是正整数、负整数、或者0，默认是auto,数值越大，盒子越靠上

②如果属性值相同，按照书写顺序，则后来者居上。

③数字后面不能加单位。

④只有定位的盒子才有z-index属性。

（6）定位拓展：绝对定位盒子居中算法

加了绝对定位的盒子不能通过margin:0 auto来水平居中。

通过以下算法可以实现：

Left:50%:让盒子的左侧移动到父级元素的水平中心位置

Margin-left:-100px:让盒子向左移动自身宽度的一半。

垂直居中：top:50% Margin-top:-100px

（7）绝对定位和固定定位也和浮动类似

①行内元素添加绝对定位和固定定位，可以直接设置宽度高度。

②块级元素添加绝对定位和固定定位，如果不给宽度或者高度，默认大小是内容的大小。

③脱标的盒子不会触发外边距塌陷：浮动元素、绝对定位元素的都不会触发外边距合并问题。

④绝对定位（固定定位）会完全压住盒子：浮动元素不同，只会压住它下面标准流的盒子，但是不会压住下面标准流盒子里面的文字（图片），但是绝对定位（固定定位）会压住下面标准流所有的内容。

浮动最初是用于作文字环绕效果的。

（8）元素的显示与隐藏

①display属性：设置一个元素应如何显示；

Display：none；隐藏对象

display:block;显示元素，或者转换为块级元素。

Display：隐藏元素后，不再占有位置。

②visibility：可见性，隐藏元素后，继续占有位置

Visibility：visible;元素可视

Visibility：hidden;元素隐藏

③overflow:溢出，该属性指定如果内容溢出一个元素的框时，可以设置以下值处理。



|  |  |
| --- | --- |
| **值** | **描述** |
| visible | 默认值。内容不会被修剪，会呈现在元素框之外。 |
| hidden | 内容会被修剪，并且其余内容是不可见的。 |
| scroll | 内容会被修剪，但是浏览器会显示滚动条以便查看其余的内容。溢出与不溢出都显示滚动条 |
| auto | 如果内容被修剪，则浏览器会显示滚动条以便查看其余的内容。溢出时显示滚动条 |
| inherit | 规定应该从父元素继承 overflow 属性的值。 |

如果有定位的盒子，需要谨慎使用overflow:hidden，因为它会隐藏多余的部分。

**3、CSS精灵技术（CSS Spirtes/CSS 雪碧）**

①目的：为了减少服务器接收和发送请求次数，提高页面的加载速度。

②核心原理：将网页中的一些小背景图像整合到一张大图中，这样服务器只需要一次请求就可以了。

③精灵技术针对于背景图片使用，就是把多个背景图片整合到一张大图中。

④移动背景图片位置，此时可以使用background-position

⑤移动距离就是这个目标图片的x和y坐标，注意网页中的坐标有所不同。一般往上往左移动是负值，使用精灵图的时候需要精确测量，每个小背景图片的大小和位置。

**4、字体图标**

字体图标的产生iconfont：看上去是图标，本质是文字。

（1）精灵图缺点：图片文件较大、图片本身放大和缩小会失真、一旦图片制作完毕想要更换非常复杂。

字体图标优点：

①一个字体图标要比一系列的图像要小，一旦字体加载了，图标就会马上渲染出来，减少服务器请求。

②本质是文字，可以随意的改变颜色，产生阴影、透明效果、旋转等。

③几乎所有浏览器支持。

注意：字体图标不能替代精灵技术，只是对工作中图标部分技术的提升和优化。

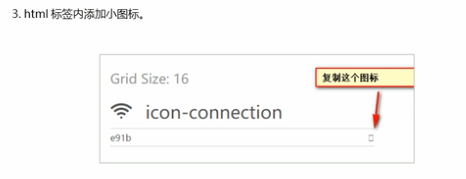


1. icomoon字体 <http://icomoon.io>

阿里iconfont字库：<http://www.iconfont.cn/>

使用：原先下载的文件不要删，解压，最为重要的是fonts文件，把fonts文件夹复制一份放到页面根目录下（即自己打的代码的根目录）；通过CSS引入文件；在html标签里面添加小图标。

操作方法：





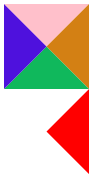
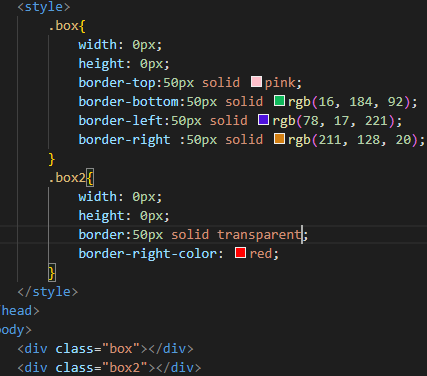
（3）字体图标的追加

原来的字体图标不够用了，我们需要添加新的字体图标到原来字体文件中，点击import icons按钮，把压缩包里面的selection.json重新上传，然后选中自己想要的新图标，重新下载安装包，并替换原来文件。



1. CSS三角、鼠标样式与解决文字图片出现的问题

**1、CSS三角制作：将宽度高度设置为0。**



CSS三角强化：将左边和底部的边框设置为0；首先需要判断是哪两个边框变0，左下或者右上。

见测试25.html（制作直角三角形）

2、**用户的鼠标样式cursor**

语法：li {

cursor:pointer;

}

Default:默认小白

Pointer:小手

Move:移动

Text:文字

Not-allowed:禁止



**3、取消表单轮廓和防止拖拽文本域resize**

给表单添加outline：0；或者outline:none；样式之后可以去掉默认蓝色边框。

防止拖拽：textarea {

resize:none;

}

1. **vertical-align**

设置图片或者表单（行内元素或者行内块元素）和文字垂直对齐。

|  |  |
| --- | --- |
| **值** | **描述** |
| baseline | 默认。元素放置在父元素的基线上。 |
| sub | 垂直对齐文本的下标。 |
| super | 垂直对齐文本的上标 |
| top | 把元素的顶端与行中最高元素的顶端对齐 |
| text-top | 把元素的顶端与父元素字体的顶端对齐 |
| middle | 把此元素放置在父元素的中部。 |
| bottom | 把元素的顶端与行中最低的元素的顶端对齐。 |
| text-bottom | 把元素的底端与父元素字体的底端对齐。 |
| length |  |
| % | 使用 "line-height" 属性的百分比值来排列此元素。允许使用负值。 |
| inherit | 规定应该从父元素继承 vertical-align 属性的值。 |

1. **解决图片底侧默认空白缝隙问题**



原因：图片是和文字基线对齐的。

方法：

1、给图片添加vertical-align:middle/top/bottom等；

2、把图片转换成块元素display:block；

1. **单行文字溢出省略号显示**



满足三个条件：

（1）先强制一行内显示文本（white-space：nowrap）normal默认自动换行

（2）超出的部分隐藏（overflow:hidden）

（3）文字省略号替代超出的部分（text-overflow:ellipsis）

1. **多行文字溢出省略号显示**

满足条件：

（1）overflow:hidden

（2）text-overflow:ellipsis

（3）Display:-webkit -box(弹性伸缩盒子模型显示)

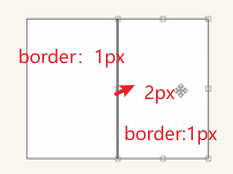
（4）-webkit-line-clamp:2;（限制一个块元素显示的文本行数）

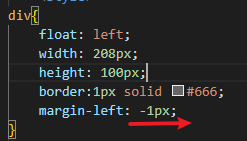
（5）-webkit-box-orient:vertical；（设置或者检索伸缩盒子对象的子元素的排列方式）

有较大兼容性问题，适合于webkit浏览器或者移动端

1. **布局技巧---margin负值运用**

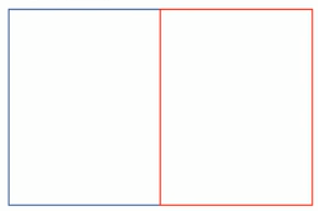
①处理两个边框叠加的效果，可以用margin-left:-1px







②处理选中盒子边框变颜色，盒子显示边框不全的问题

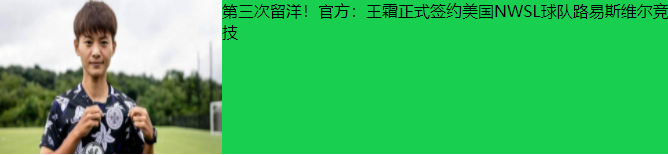


鼠标经过某个盒子的时候，提高盒子的层级（如果没有定位，则加相对定位保留位置，如果有定位，则加z-indenx提高层级）

1. **文字围绕浮动元素**



例子：



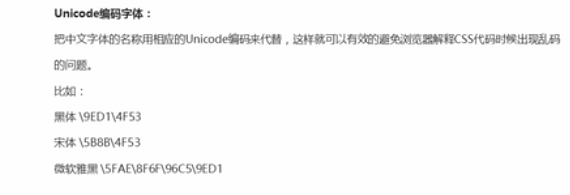
方法：先用一个盒子设置高度宽度装，另一个盒子设定高度宽度装照片，添加浮动即可，文字不必加浮动。

1. **行内块的巧妙运用**

只要给父元素行内块元素添加text-align:center，其他子元素就会一起居中。详细见23例子。

1. **CSS初始化**

例如：清除浮动，清除图片空白缝隙，删除内外边距



第十二章 html新特性

html新特性：基本是IE9+以上版本浏览器才支持

**1、语义化标签**

<header>:头部标签

<nav>导航标签

<article>:内容标签

<section>:定义文档某个区域

<aside>:侧边栏标签

<footer>:尾部标签

注意：这种语义化标签主要是针对搜索引擎的，这些标签在页面可以使用多次，在IE9中，需要把这些元素转换为块级元素。

**2、HTML视频标签**

音频：<audio>

视频：<video>

当前， <video> 元素支持三种视频格式： MP4, WebM, 和 Ogg:

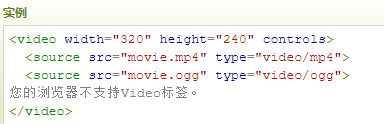
|  |  |  |  |
| --- | --- | --- | --- |
| **浏览器** | **MP4** | **Ogg/webM** | |
| Internet Explorer | YES | | NO |
| Chrome | YES | | YES |
| Firefox | YES | | YES |
| Safari | YES | | NO |
| Opera | YES (从 Opera 25 起) | | YES |

当前<video>元素支持三种视频格式，尽量使用mp4格式。



语法：<video src=”文件地址” controls=”controls”></video>

或者

（仅为参考，一般以第一种为主）

属性值

|  |  |  |
| --- | --- | --- |
| **属性** | **值** | **描述** |
| [autoplay](https://www.w3school.com.cn/tags/att_video_autoplay.asp" \o "HTML5 <video> autoplay 属性) | autoplay | 如果出现该属性，则视频在就绪后马上播放。 |
| [controls](https://www.w3school.com.cn/tags/att_video_controls.asp" \o "HTML5 <video> controls 属性) | controls | 如果出现该属性，则向用户显示控件，比如播放按钮。 |
| [height](https://www.w3school.com.cn/tags/att_video_height.asp" \o "HTML5 <video> height 属性) | *pixels* | 设置视频播放器的高度。 |
| [loop](https://www.w3school.com.cn/tags/att_video_loop.asp" \o "HTML5 <video> loop 属性) | loop | 如果出现该属性，则当媒介文件完成播放后再次开始播放。 |
| [muted](https://www.w3school.com.cn/tags/att_video_muted.asp" \o "HTML5 <video> muted 属性) | muted | 规定视频的音频输出应该被静音。 |
| [poster](https://www.w3school.com.cn/tags/att_video_poster.asp" \o "HTML5 <video> poster 属性) | *URL* | 规定视频下载时显示的图像，或在用户点击播放按钮前显示的图像。 |
| [preload](https://www.w3school.com.cn/tags/att_video_preload.asp" \o "HTML5 <video> preload 属性) | preload | 如果出现该属性，则视频在页面加载时进行加载，并预备播放。  如果使用 "autoplay"，则忽略该属性。 |
| [src](https://www.w3school.com.cn/tags/att_video_src.asp" \o "HTML5 <video> src 属性) | *url* | 要播放的视频的 URL。 |
| [width](https://www.w3school.com.cn/tags/att_video_width.asp" \o "HTML5 <video> width 属性) | *pixels* | 设置视频播放器的宽度。 |
| Poster | *Url* | 加载等待的页面图片 |

3、**音频标签**

语法：<sudio src=”文件地址” control=”controls”></audio>

或者

<audio controls>

<source src="horse.ogg" type="audio/ogg">

<source src="horse.mp3" type="audio/mpeg"> 您的浏览器不支持 audio 元素。 </audio>

目前, <audio>元素支持三种音频格式文件: MP3, Wav, 和 Ogg。

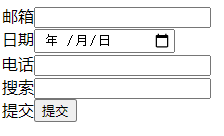
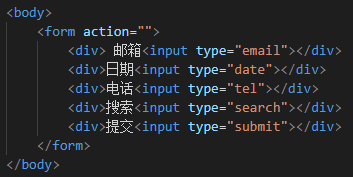
属性值

|  |  |  |
| --- | --- | --- |
| **属性** | **值** | **描述** |
| [autoplay](https://www.w3school.com.cn/tags/att_audio_autoplay.asp" \o "HTML5 <audio> autoplay 属性) | autoplay | 如果出现该属性，则音频在就绪后马上播放。 |
| [controls](https://www.w3school.com.cn/tags/att_audio_controls.asp" \o "HTML5 <audio> controls 属性) | controls | 如果出现该属性，则向用户显示控件，比如播放按钮。 |
| [loop](https://www.w3school.com.cn/tags/att_audio_loop.asp" \o "HTML5 <audio> loop 属性) | loop | 如果出现该属性，则每当音频结束时重新开始播放。 |
| [muted](https://www.w3school.com.cn/tags/att_audio_muted.asp" \o "HTML5 <muted> loop 属性) | muted | 规定视频输出应该被静音。 |
| [preload](https://www.w3school.com.cn/tags/att_audio_preload.asp" \o "HTML5 <audio> preload 属性) | preload | 如果出现该属性，则音频在页面加载时进行加载，并预备播放。如果使用 "autoplay"，则忽略该属性。 |
| [src](https://www.w3school.com.cn/tags/att_audio_src.asp" \o "HTML5 <audio> src 属性) | *url* | 要播放的音频的 URL。 |

注意：谷歌浏览器把音频和视频自动播放禁止了。

**4、input标签**

* Color：生成颜色选择表单
* Date：日期类型
* datetime
* datetime-local
* email
* Month：月类型
* Number：数字类型
* range
* Search：搜索框
* Tel：手机号码
* Time：时间类型
* Url
* Week：周类型
* email：邮件类型

我们验证的时候必须添加form表单域

Input表单属性

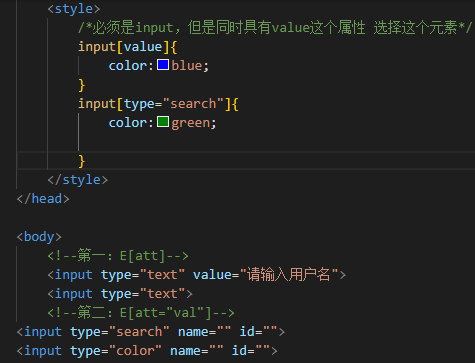
|  |  |  |
| --- | --- | --- |
| **属性** | **值** | **描述** |
| [autocomplete](https://www.w3school.com.cn/tags/att_input_autocomplete.asp" \o "HTML5 <input> autocomplete 属性) | * on * off | 规定是否使用输入字段的自动完成功能。“=on”时表示有提示，”=off”表示满意提示 |
| [autofocus](https://www.w3school.com.cn/tags/att_input_autofocus.asp" \o "HTML5 <input> autofocus 属性) | autofocus | 规定输入字段在页面加载时是否获得焦点。（光标）  （不适用于 type="hidden"） |
| [multiple](https://www.w3school.com.cn/tags/att_input_multiple.asp" \o "HTML5 <input> multiple 属性) | multiple | 可以多选文件提交。 |
| [placeholder](https://www.w3school.com.cn/tags/att_input_placeholder.asp" \o "HTML5 <input> placeholder 属性) | *text* | 规定帮助用户填写输入字段的提示。 |
| [required](https://www.w3school.com.cn/tags/att_input_required.asp" \o "HTML5 <input> required 属性) | required | 指示输入字段的值是必需的。 |

第十三章 CSS3新特性

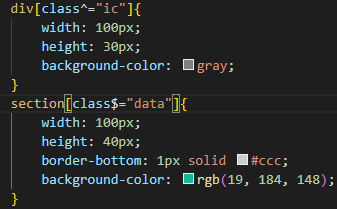
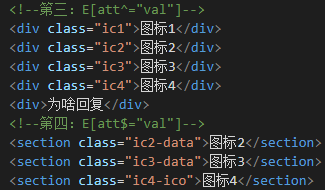
**1、新增选择器**

①属性选择器：根据元素特定属性来选择元素

|  |  |
| --- | --- |
| 选择符 | 简介 |
| E[att] | 选择具有att属性的E元素 |
| E[att=”val”] | 选择具有att属性且属性值等于val的E元素 |
| E[att^=”val”] | 匹配具有att属性且值以val开头的E元素 |
| E[att$=”val”] | 匹配具有att属性且值以val结尾的E元素 |
| E[att\*=”val”] | 匹配具有att属性且值中含有val的E元素 |







②结构伪类选择器：根据文档结构来选择器元素，用于根据父级选择器里面的子元素

|  |  |
| --- | --- |
| 选择符 | 简介 |
| E:first-child | 匹配父元素中的第一个子元素E |
| E:last-child | 匹配父元素中的最后一个E元素 |
| E:nth-child(n) | 匹配父元素中的第n个子元素 |
| E:first-of-type | 指定类型E的第一个 |
| E:last-of-type | 指定类型E的最后一个 |
| E:nth-of-type(n) | 指定类型E的第n个 |

nth-child会把所有孩子排列序号

E:nth-child(n)可以是数字，关键字和公式;

n是数字代表第几个；

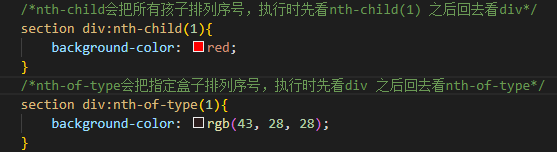
n是关键字，如even偶数，odd奇数；

n是公式，从0 计起

常见公式如下：

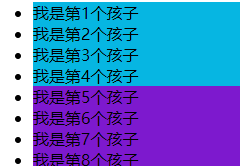
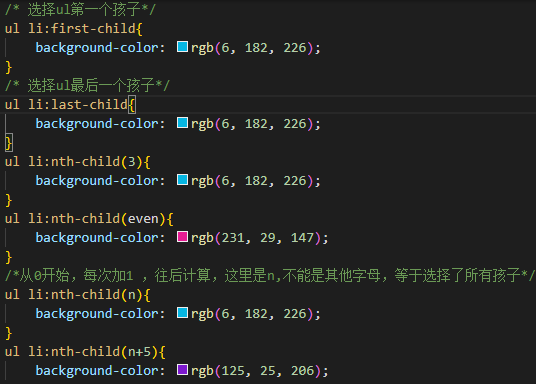
|  |  |
| --- | --- |
| 公式 | 取值 |
| 2n | 偶数 |
| 2n+1 | 奇数 |
| 5n | 5 10 15... |
| n+5 | 从第五个开始（包含第五个）到最后 |
| -n+5 | 前5个（包含第五个） |

区别：





类选择器、属性选择器、伪类选择器权重为10；



③伪元素选择器

|  |  |
| --- | --- |
| 选择符 | 简介 |
| ：：before | 在元素内部的前面插入内容 |
| ：：after | 在元素内部的后面插入内容 |

Before和after创建一个元素，但是属于行内元素；

新创建的这个元素在文档树中是找不到的，所以我们称为伪元素；

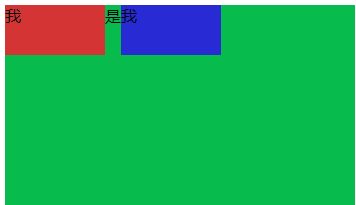
语法：

* element::before{}
* Before和after必须有content属性
* Before在父元素内容的前面创建元素，after在父元素内容的后面插入元素；

哈哈哈

After

Before



* 伪元素选择器和标签选择器一样，权重为1。

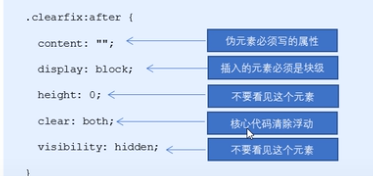
例子



**2、伪元素清除浮动**

①额外标签发法

②伪元素after



③伪元素before和after清除浮动



第十四章 CSS3盒子与图片处理

**1、CSS3盒子模型**

（1）CSS3 中可以通过box-sizing来指定盒模型，有2个值：content-box 、border-box这样子计算盒子大小方式发生了变化。

可以分为两种情况：

（1）box-sizing:content-box 盒子大小为width+padding+border（以前默认的）

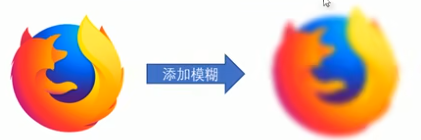
（2）box-sizing:border-box 盒子大小为width

如果盒子模型我们改为box-sizing:border-box，那padding和border就不会撑大盒子（前提padding和border不会超过width宽度）

**2、CSS3图片模糊处理**

CSS3 滤镜filter:将模糊或者颜色偏移等图形效果应用于元素

Filter函数 eg：filter：blur(5px) blur 模糊处理 数值越大越模糊。

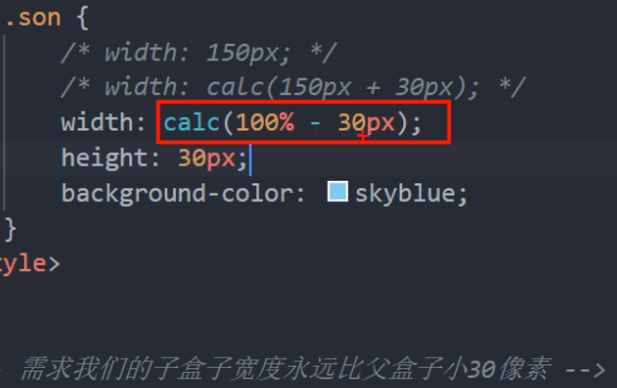


**3、计算盒子宽度calc函数**

语法：

Width：calc(100% - 80px) 100%表示父盒子的宽度，括号里可以用+-\*/来计算

例子：



1. **CSS3过渡**

定义：从一个样式转换为另一个样式（谁做过渡给谁加）可以写多个属性，用逗号隔开





Transition：要过渡的属性 花费时间 运动曲线 何时开始

①属性：想要变化的CSS属性，宽度高度 背景颜色 内外边距都可以 如果想要所有属性过渡，写一个all就可以

②花费时间：单位是秒，比如0.5秒

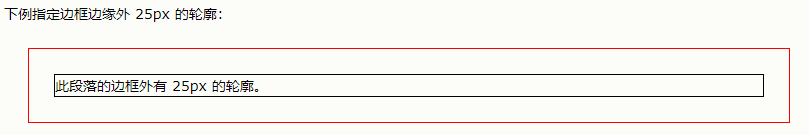
③运动曲线：默认是ease(可以省略)

④何时开始：单位是秒，可以设置延迟出发时间 默认是0s（可以省略）.

|  |  |
| --- | --- |
| **值** | **描述** |
| [transition-property](https://www.w3school.com.cn/cssref/pr_transition-property.asp" \o "CSS3 transition-property 属性) | 规定设置过渡效果的 CSS 属性的名称。 |
| [transition-duration](https://www.w3school.com.cn/cssref/pr_transition-duration.asp" \o "CSS3 transition-duration 属性) | 规定完成过渡效果需要多少秒或毫秒。 |
| [transition-timing-function](https://www.w3school.com.cn/cssref/pr_transition-timing-function.asp" \o "CSS3 transition-timing-function 属性) | 规定速度效果的速度曲线。 |
| [transition-delay](https://www.w3school.com.cn/cssref/pr_transition-delay.asp" \o "CSS3 transition-delay 属性) | 定义过渡效果何时开始。 |
| **值** | **描述（**[transition-timing-function](https://www.w3school.com.cn/cssref/pr_transition-timing-function.asp" \o "CSS3 transition-timing-function 属性)**）** | |
| linear | 规定以相同速度开始至结束的过渡效果（等于 cubic-bezier(0,0,1,1)）。 | |
| ease | 规定慢速开始，然后变快，然后慢速结束的过渡效果（cubic-bezier(0.25,0.1,0.25,1)）。 | |
| ease-in | 规定以慢速开始的过渡效果（等于 cubic-bezier(0.42,0,1,1)）。 | |
| ease-out | 规定以慢速结束的过渡效果（等于 cubic-bezier(0,0,0.58,1)）。 | |
| ease-in-out | 规定以慢速开始和结束的过渡效果（等于 cubic-bezier(0.42,0,0.58,1)）。 | |
| cubic-bezier(n,n,n,n) | 在 cubic-bezier 函数中定义自己的值。可能的值是 0 至 1 之间的数值。 | |

1. **广义的html5=html5本身+CSS3+JavaScript。**

**6、outline-offset**:属性在元素的轮廓与边框之间添加空间。



p {

margin: 50px;

border: 1px solid black;

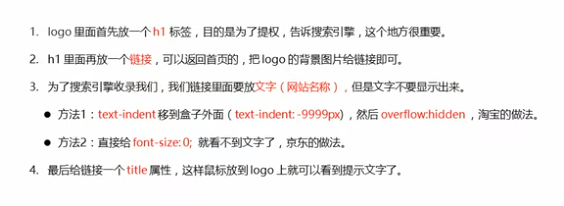
outline: 1px solid red;

outline-offset: 25px;

}

第十五章 品优购项目

**1、logo优化**

**2、搜索框**

（1）先弄一个大盒子，再用input:search和button放进盒子里；

（2）如果出现用ul li 制作的盒子里面，本来一行显示4个，但是只显示三个，另一个在下一行显示，则应该调整ul的宽度

第十六章 服务器

（1）服务器（我们也会称为主机）是提供计算服务的设备，它也是一台计算机，在网络环境下，根据服务器提供的服务类型不同，服务器又分为文件服务器，数据库服务器，应用程序服务器，web服务器等。

（2）Web服务器一般指网站服务器，是指驻留于因特网上某种类型计算机的程序，可以向浏览器等web客户端提供文档，也可以放置网站文件，让全世界浏览，可以放置数据文件，让全世界下载。

（3）根据服务器在网络中所在位置不同，又可分为本地服务器和远程服务器。

本地服务器：可以把自己的电脑设置为本地服务器，这样同一个局域网内地的同学就可以访问你的网页。

远程服务器：别的公司为我们提供一个主机（电脑），我们只要把网站项目上传到这台电脑中，任何人都可以利用域名访问我们的网站了。

免费远程服务器（免费空间）：http://free.3v.do/

第十七章 2D转换（在二维坐标系进行改变）

transform转换：可以实现移动（translate）、旋转（ritate）、缩放（scale）

移动translate。

**1、移动位置的语法：定位、外边距、移动。**

语法：transform:translate(x,y)(x,y表示在x轴和y轴上移动位置）

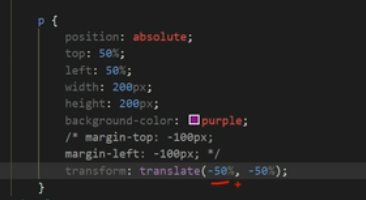
若只移动一个轴，则另一个置0；或者transform：translateX(100px)

或者transform：translateY(100px)

1. **translate最大优点**

不会影响其他元素的位置。

（1）translate中的百分比单位是相对于自身元素的translate:(50%,50%)，对行内标签没有效果。



（2）rotate旋转

语法：transform：rotate(度数）

①rotate里面是度数，单位是deg，比如rotate(45deg)

②角度为正时，顺时针，负时，逆时针

③默认旋转中心是元素的中心点

（3）设置转换中心

语法：transform-origin:x y;

注意：参数后面用空格分开，x,y默认转换中心点是（50%，50%），还可以给x,y设置像素或者方位名词（top/bottom/left/right/center）

左下角：transform-origin:left bottom

（4）缩放scale

语法：transform：scale(x,y)

注意：其中的x和y用逗号分隔，里面写数字不跟单位，是倍数的意思。

①Transform:scale(1,1)：宽和高都放大一倍，相对于没有放大;

②Transform:scale(2,2)：宽和高都放大了2倍

③Transform:scale(2)：只写一个参数，第二个参数则和第一个参数一样，相当于s scale(2,2)

④Transform:scale(0.5,0.5)：缩小（小于1）

scale缩放量最大的优势：可以设置转换中心点缩放，默认以中心点缩放的，而且不影响其他盒子。

**3、2D转换之书写顺序**

①同时使用多个转换，书写顺序为：transform:translate() rotate() scale()等。

②其顺序会影响转换效果（先旋转会改变坐标轴方向）

③当我们同时有位移和其他属性的时候，记得要将位移放在最前面

1. CSS动画（animation）

CSS动画可以实现更多变化，更多控制，连续自动播放。

**1、用 keyframes定义动画**

在@keyframe中规定CSS某项样式，就能创建由当前样式逐渐改为新样式的动画效果

用百分比来规定变化发生的时间，或用关键词from和to，等于0%和100%。

@keyframe 动画名称{

0%{ 动画开始

Width:100px;

}

100%{ 动画完成

Width:200px;

}

}

**2、使用动画**

Div{

Width:200px;

Height:200px;

Background-color:pink;

Animation-name:动画名称;

Animation-duration:持续时间;

}

**3、动画常见属性**

|  |  |
| --- | --- |
| **属性** | **描述** |
| [@keyframes](https://www.w3school.com.cn/cssref/pr_keyframes.asp" \o "CSS @keyframes 规则) | 规定动画模式。 |
| [animation](https://www.w3school.com.cn/cssref/pr_animation.asp" \o "CSS animation 属性) | 设置所有动画属性的简写属性。 |
| [animation-delay](https://www.w3school.com.cn/cssref/pr_animation-delay.asp" \o "CSS animation-delay 属性) | 规定动画开始的延迟、何时开始 |
| [animation-direction](https://www.w3school.com.cn/cssref/pr_animation-direction.asp" \o "CSS animation-direction 属性) | 定动画是向前播放、向后播放还是交替播放。反方向播放：alternate，默认normal |
| [animation-duration](https://www.w3school.com.cn/cssref/pr_animation-duration.asp" \o "CSS animation-duration 属性) | 规定动画完成一个周期应花费的时间。 |
| [animation-fill-mode](https://www.w3school.com.cn/cssref/pr_animation-fill-mode.asp" \o "CSS animation-fill-mode 属性) | 规定元素在不播放动画时的样式（在开始前、结束后，或两者同时）。保持：forwards；回到起始处：backwards |
| [animation-iteration-count](https://www.w3school.com.cn/cssref/pr_animation-iteration-count.asp" \o "CSS animation-iteration-count 属性) | 规定动画应播放的次数。默认1， 循环播放infinite |
| [animation-name](https://www.w3school.com.cn/cssref/pr_animation-name.asp" \o "CSS animation-name 属性) | 规定 @keyframes 动画的名称。 |
| [animation-play-state](https://www.w3school.com.cn/cssref/pr_animation-play-state.asp" \o "CSS animation-play-state 属性) | 规定动画是运行还是暂停。默认running还有paused |
| [animation-timing-function](https://www.w3school.com.cn/cssref/pr_animation-timing-function.asp" \o "CSS animation-timing-function 属性) | 规定动画的速度曲线。默认ease |

**4、动画简写**

格式：animation:动画名称 持续时间 移动曲线 何时开始 播放次数 是否反方向 动画起始或者结束状态

Eg:animation:move 5s linear(匀速) 2s infinite alternate

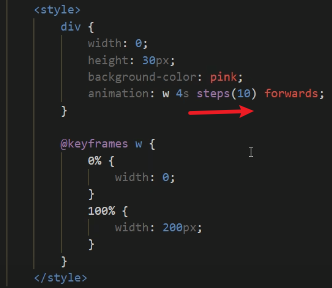
注意：简写属性里面不包含animation-play-state

暂停动画:animation-play-state:paused:经常和鼠标经过等其他配合使用

**5、速度曲线细节**

|  |  |
| --- | --- |
| **值** | **描述** |
| linear | 动画从头到尾的速度是相同的。 |
| ease | 默认。动画以低速开始，然后加快，在结束前变慢。 |
| ease-in | 动画以低速开始。 |
| ease-out | 动画以低速结束。 |
| ease-in-out | 动画以低速开始和结束。 |
| cubic-bezier(*n*,*n*,*n*,*n*) | 在 cubic-bezier 函数中自己的值。可能的值是从 0 到 1 的数值。 |
| Steps() | 指定了时间函数中的间隔数量（步长） |

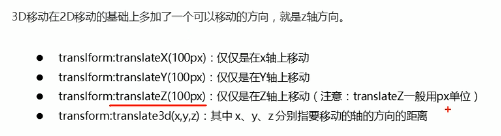
Eg:



第十九章 3D转换

**1、translate3d(x,y,z)**

translate3d(x,y,z)=translateX(100px) translateY(100px) translateZ(100px)

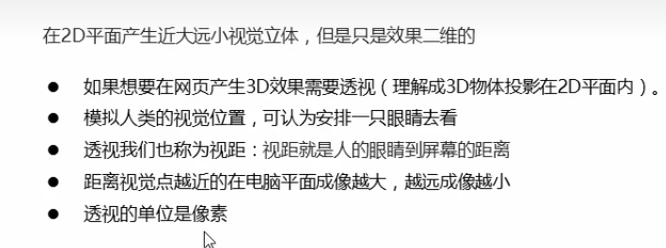


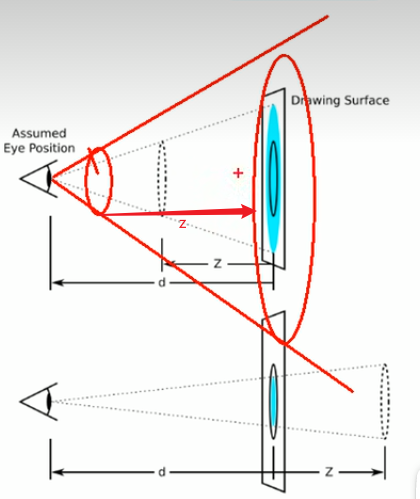
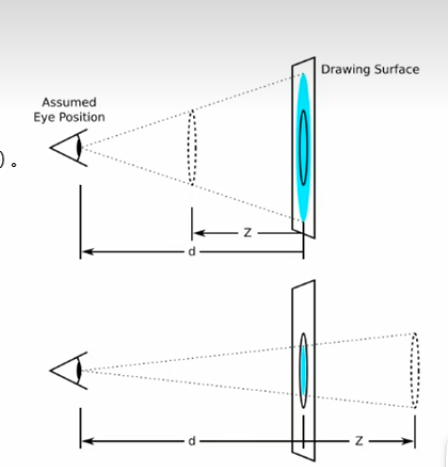
Z轴后面单位一般跟px，借助透视才可以看见；

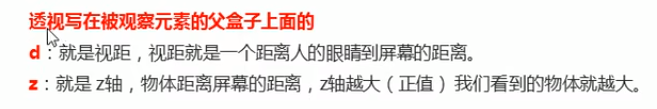
translate3d(x,y,z)里面的xyz不可以省略，没有为0；

translateZ (100px):向外移动，眼前移动

**2、perspective透视**



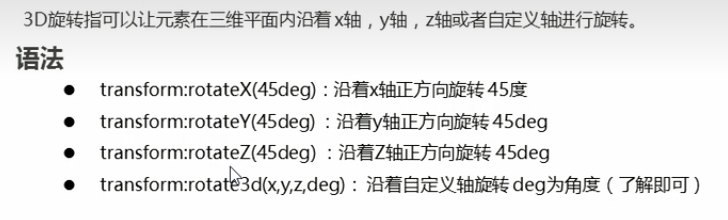




如果只有一个盒子，其父盒子就是body

语法：perspective:\*\*px，透视值越大，看到的物体越小

**3、3D旋转rotate3d**



Eg:transform: rotate3d(1,1,0,80deg);

（1,1,0）按照矢量进行旋转

