**（一）使用**

<style type=”text/css”>

具体css语句

</style>

**（二）语法**

<styletype="text/css">

mm {background-color:red}; /\*这行表示标签为mm的，背景色为红色，如果mm替换为p，则所有<p></p>标签覆盖内的变为红色\*/

</style>

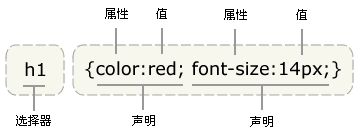
①包含在<style>之中；

②格式为：**标签名 { 具体样式 }**

③使用花括号来包含；

④有时候需要使用双引号来包含值

大概如下图



**（三）颜色**

mm {background-color:red} /\*这行表示标签为mm的，背景色为红色，如果mm替换为p，则所有<p></p>标签覆盖内的变为红色\*/

mm {color:rgb(50%,50%,0%)} /\*可以和上一行写一起，也可以分开写\*/

nn{color:rgb(50,55,155)} /\*另外一种颜色表示方法\*/

nn{background-color:#0F0} /\*第三种颜色表示方法\*/

**（四）多重声明：**

即把上面的第1、2行合并

mm {background-color:red;

color:rgb(50%,50%,0%)}

效果是一样的

**（五）大小写**

一般来说，CSS对大小写不敏感，但如果名称涉及HTML等，则需要注意大小写（如class=”样式”）

**（六）同一种标签里，通过添加新的标签，以使其和一般的标签不一样**

nn{color:rgb(50,55,155)} /\*另外一种颜色表示方法\*/

nn{background-color:#0F0;

font-size:30px;

} /\*第三种颜色表示方法\*/

nn different { /\*虽然在nn里，但背景颜色和普通的nn不同，不过字体依然使用nn的大小（其他的也是）\*/

background-color:rgb(10%,10%,10%)}

}

**（七）id选择器和类选择器**

在html可以在标签里加id，然后css可以对id相符的进行样式操作。

有标签：<mm id="nn">pppp</mm>

有操作：

#nn {

background-color:red;

position:absolute;

left:500px;

top:500px;

color:rgb(50%,50%,0%);

font-size:45px;

}

即可

**（八）绝对定位**

丢一个盒子的样式：

.box {

position: absolute;

height: 200px;

width: 200px;

background-color: #ddd;

border: 1px #eee;

border-top-width: 1px;

border-right-width: 1px;

border-bottom-width: 1px;

border-left-width: 1px;

border-top-style: none;

border-right-style: none;

border-bottom-style: none;

border-left-style: none;

border-top-color: rgb(238, 238, 238);

border-right-color: rgb(238, 238, 238);

border-bottom-color: rgb(238, 238, 238);

border-left-color: rgb(238, 238, 238);

-moz-border-top-colors: none;

-moz-border-right-colors: none;

-moz-border-bottom-colors: none;

-moz-border-left-colors: none;

border-image-source: none;

border-image-slice: 100% 100% 100% 100%;

border-image-width: 1 1 1 1;

border-image-outset: 0 0 0 0;

border-image-repeat: stretch stretch;

padding: 5px;

padding-top: 5px;

padding-right: 5px;

padding-bottom: 5px;

padding-left: 5px;

}

**（8）CSS的position**

语法为：

position:fixed

|  |  |
| --- | --- |
| **值** | **描述** |
| absolute | 生成绝对定位的元素，相对于 static 定位以外的第一个父元素进行定位。  元素的位置通过 "left", "top", "right" 以及 "bottom" 属性进行规定。 |
| fixed | 生成绝对定位的元素，相对于浏览器窗口进行定位。  元素的位置通过 "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。 |

（9）div里面套用div

html方面：

<div id="box">

<div class="smallbox">

<div class="smallerbox1">

No1：

</div>

<div id="no1" class="smallerbox2">

文字1

</div>

</div>

<div class="smallbox">

<div class="smallerbox1">

No2：

</div>

<div id="no1" class="smallerbox2">

文字2

</div>

</div>

</div>

css方面：

<style type="text/css">

#box{

position:absolute;

width:400px;

height:500px;

background-color:#ddd;

}

.smallbox{

position:abbr;

width: 400px;

height:100px;

left:20px;

top:20px;

background-color:#aaa;

}

.smallerbox1{

position:relative;

left:20px;

top:10px;

background-color:#777;

width:60px;

}

.smallerbox2{

position:relative;

left:80px;

top:-12px;

background-color:#333;

width:200px;

}

</style>

**（9）CSS padding margin border属性详解**

**关于定义：**

margin：层的边框以外留的空白

background-color：背景颜色

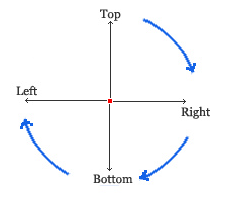
background-image：背景图片

padding：层的边框到层的内容之间的空白

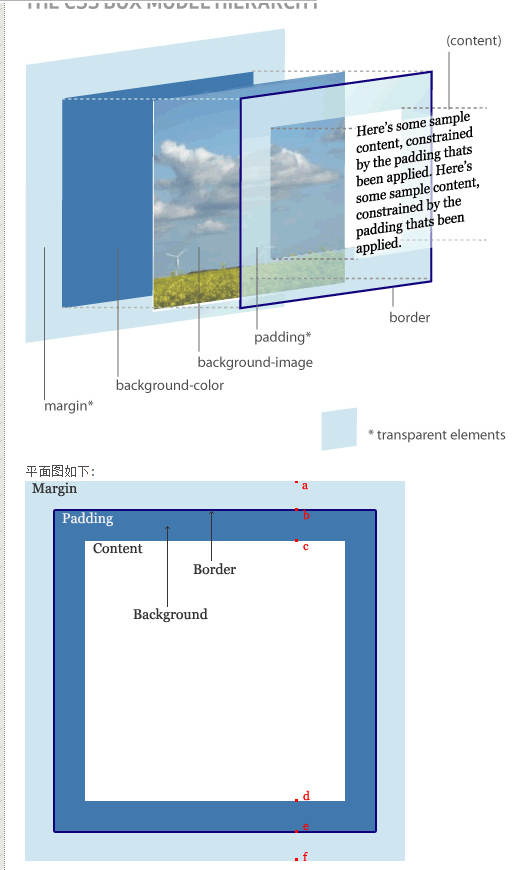
border：边框

content：内容

关于四个方向：



先上图：



测试：（有些结论是验证后copy现有的，因为纯测试不可能推出必然结果）

①首先定义三个：body为背景，one为测试块，another为对比块

body

{

background-color:#000;

}

.one

{

position:absolute;

background-color:#333; /\*黑色背景\*/

width:200px;

height:200px;

text-align:center;

font-size:50px;

}

.another

{

position:absolute;

top:250px;

background-color:#999;

width:200px;

height:200px;

text-align:center;

font-size:50px;

}

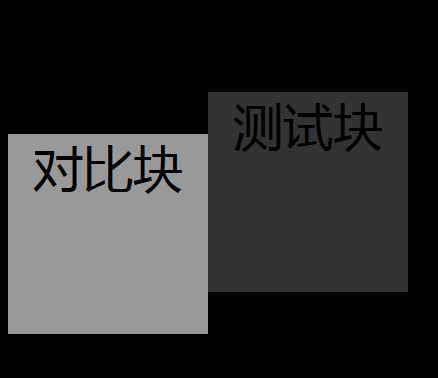
效果：



②向测试块追加属性margin（**控制块级元素之间的距离**）：

margin:200px;

效果为：



说明：

<1>margin本身是透明的（背景还是黑色）；

<2> margin:200px;是向四个方向追加200px，相当于上下、左右各增加400px；

<3>单纯追加margin-top:200px; 将导致原有块往下移动200px距离；

假如删除两个块的： position:absolute; 将导致当前块和之后的块都移动相对位置。

变化前：



变化后：



<4>margin-left:200px 不会使得后面的块占用之前块的位置（因为相当于前块width增加200px）

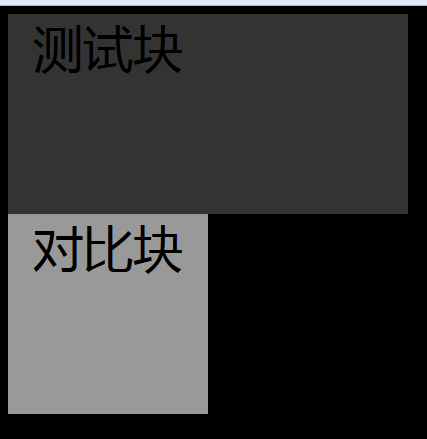
③向块内追加padding属性：（**控制块级元素内部**）

padding:200px;追加后：



<1>说明padding分为四个方向。200相当于每边追加200px

padding-right:200px; 追加后：



<1>说明padding不影响块内文字原有布局（文字还在原来位置）。

④关于border：

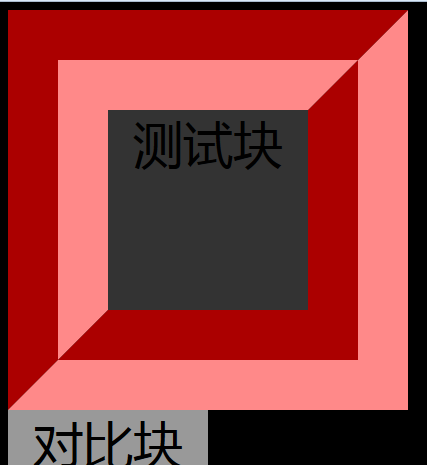
语法：

border: 粗细 样式 颜色;

**关于样式：**

* none: 无样式
* hidden: 除了同表格的边框发生冲突的时候,其它同none
* dotted: 点划线
* dashed: 虚线
* solid: 实线
* double: 双线,两条线加上中间的空白等于[border-width](http://www.dreamdu.com/css/property_border-width/)的取值
* groove: 槽状
* ridge: 脊状,和groove相反
* inset: 凹陷
* outset:凸出,和inset相反

追加border:100px groove red;



效果说明：

<1>border的第一个参数同样指每个边增加的数值。

<2>第二个参数是样式（更多样式参考上面）；

<3>第三个参数是颜色（也可以写为#000这种形式）；

⑤border和padding的双重追加

padding-right:200px;

border:50px double #fff;



效果说明：

①border在padding外侧（二者没有重叠部分）；

⑥border、padding、margin三者同时追加：

padding-right:200px;

border:50px double #fff;

margin-bottom:100px;

效果：



效果说明：

①margin在border的更外面。

**总结：**

**①最外面是margin：**

不能染色（透明的），属于块间元素（控制块和块的距离）；可以选择宽度，可以只影响某一个方位。

**②次外面是border：**

可以染色（同时也受背景图片/背景颜色的影响），可以选择样式，可以选择宽度，可以只在某一个方位应用如：border-left:50px double #fff;

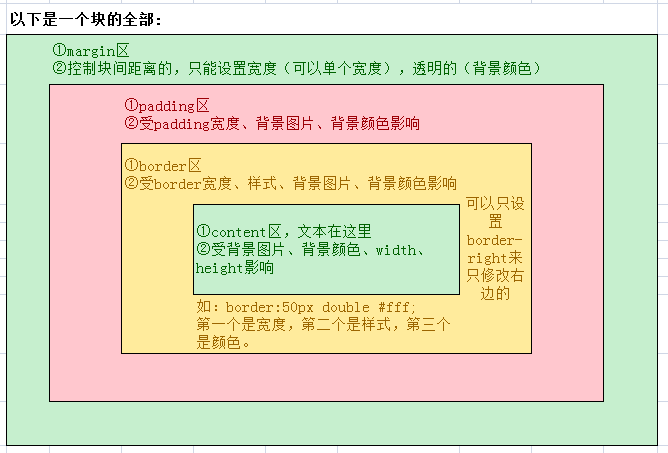
**③再次是padding：**

可以染色（根据background-color），受背景图片的影响，可以选择宽度，可以只影响某一个方位。不影响最内层的文字。

**④最内层是content：**

文本内容区，受width、height、背景图片、背景颜色等影响。

如图：



**（10）打开一个网页时，以新窗口形式打开**

将

<a href="www.baidu.com"></a>

新增属性：

<a href="www.baidu.com" **target="\_blank"**></a>

即可。

**（11）关于块间的莫名空隙**

假设有html代码：

<div id="a">

<div id="mm">

aa

</div>

</div>

<div id="b">

<div id="mm">

bb

</div>

</div>

其有css属性：

#mm{

margin:20px;

}

**问题：**

若给a块和b块染色，会发现，a块和b块之间将会有一定空隙（透明的）。

**解决办法：**

①给mm的父块a和b，增加一条属性：

padding:1px;

即可（不能是0）。（背景色将自动填充空隙区域）

②如果想避免在其他方向出现的空隙，那么在只在需要的方向（例如只对padding-top和padding-buttom设置属性）

**（12）背景图片（完全填充）**

HTML代码：

<!--背景图片-->

<div id="Layer1">

<img src="image/index\_bg2.jpg" height="100%" width="100%"/>

</div>

CSS代码：

#Layer1

{ /\*由于背景图片不能缩放，故如此设置\*/

position:absolute;

width:100%;

height:100%;

z-index:-1;

opacity:0.7;

}

即可

**（13）输入框在空的时候有文字，输入内容后消失**

<input id="keyword" type="text" placeholder="过滤" value="" />

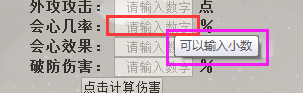
这里的placeholder="过滤"，就是底文字

**（14）鼠标移动到上面的时候，显示提示**

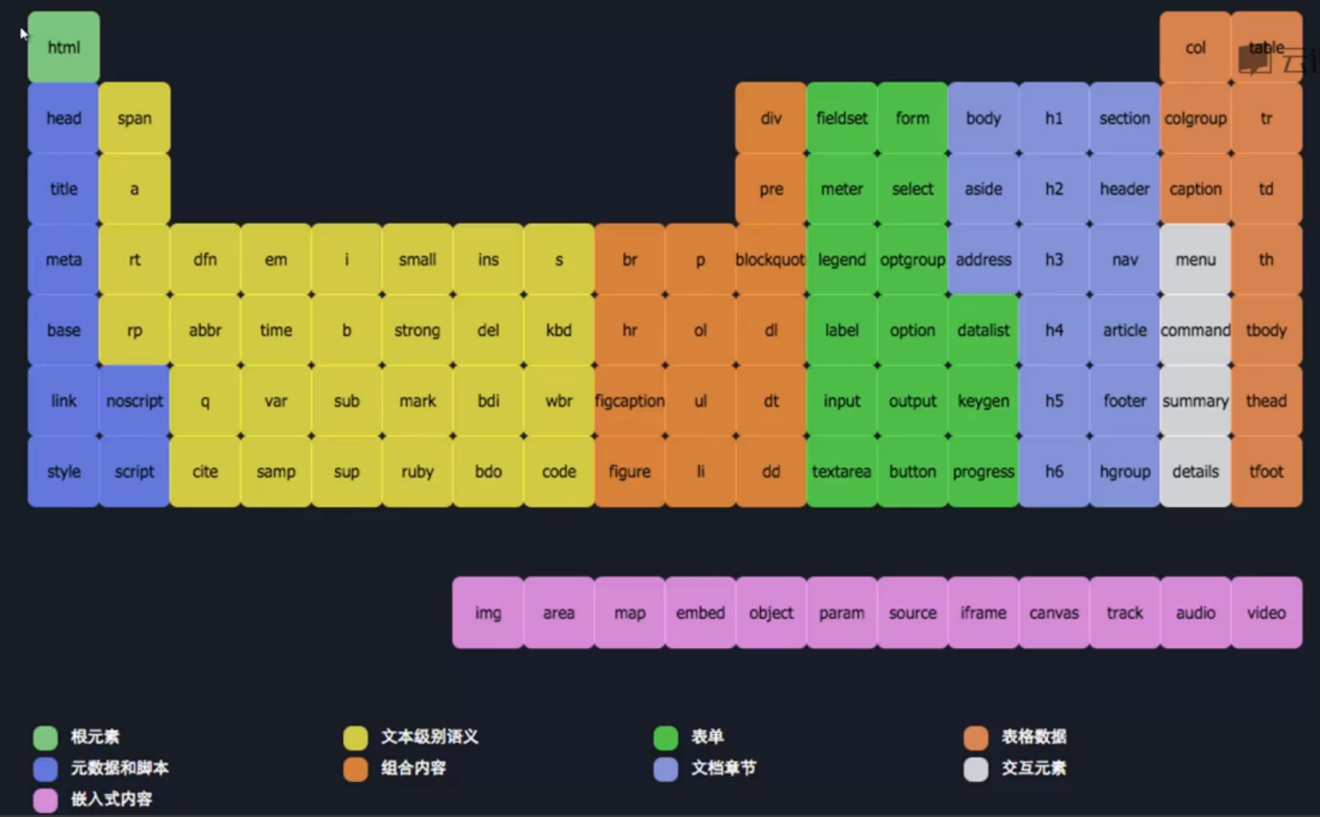
如代码：

<input id="baoji" type="text" placeholder="请输入数字" style="text-align:right;width:75px" title="可以输入小数">

效果：



**（15）HTML标签图**



**（16）设置文本垂直居中**

简单办法是：在CSS属性里，设置

line-height: xx px;

其中的xx，值和这个div块的height的一样。

**（17）选择器整理**

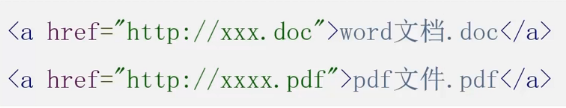
|  |  |  |  |
| --- | --- | --- | --- |
| 选择器 | 方法 | HTML | 示例 |
| id | #id名 | <p id="a"> | #a{xxxx} |
| class | .class名 | <p class="a"> | .a{xxx} |
|  |  |  |  |
| 通配符 | 通配符名 | <div> | div{xxxx}  如果是\*，表示所有元素 |
| 属性 | disabled | <input disabled>  <xx type="button> | [disabled]{xxxx}  [type=button]{xxx} |
| 选择所有 有class="a"的 | [class~=class名] | <x class="a b">  <x class="a">  两个被同时选中 | [class~=a]{xxx} |
| 选择所有某属性的值里有pp | [属性|=pp] | <p lang="pp">  <p lang="bpp">  都要被选中 | [lang|=pp]{xxx} |

②



选中href里面，有#的（第2、3个），这个符号是英文状态下，shift + 6

③





选中href里面，格式是pdf的（即第二个）

④

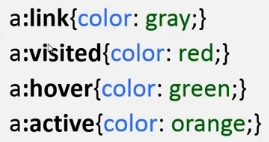




选择href里面，值有lady.163.com的所有标签（即1和2）

④伪类选择器





a:link表示，所有链接的样式；

a:visited表示，访问过的链接的样式；

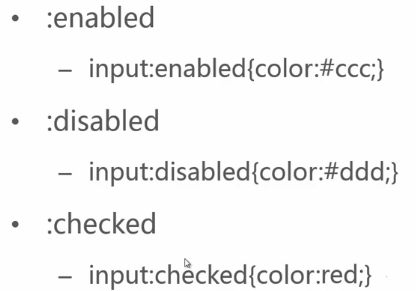
a:hover表示，鼠标移动上去之后的样式；

a:active表示，用户鼠标点击上去之后的样式

其中hover和active还可以对其他的生效。

如果四个都用，要按顺序来写。

⑤不同状态下的样式



enabled表示，可用时的样式；

disabled表示，不可用时的样式；

checked表示，选中时的样式。（比如单选框）

⑥和列表相关的



li:first-child{xxxx}表示第一个的样式

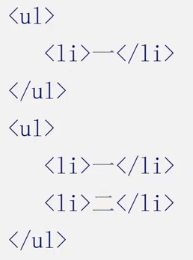
li:last-child{xxx}表示最后一个的样式

li:nth-child(even){xxx}表示奇偶相间的，偶数项的样式，odd表示奇数项。注意，如果有其他<p>或者<li>标签或其他，虽然标签名不同，但依然会影响起效果的顺序（比如奇数变偶数）。

even可以修改，比如使用2n+2（n从0开始），表示选择2、4、6项个子元素。

也可以改为：li:nth-last-child(2n+2){xxx}表示倒着数第2、4、6个，这里是5、3、1项

⑦只选中只有一个子元素的项





html代码如：

<html>

<body>

<ol>

<li>aa</li>

</ol>

<ul>

<li>bb</li>

<li>cc</li>

</ul>

<p>qq</p>

</body>

<style>

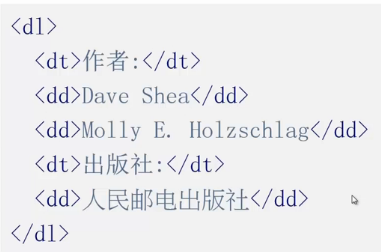
:only-child{color:red;}

</style>

</html>

只影响aa

⑧对于自定义列表，选择第一个<dd>



dd:first-of-type{color:red}

效果是Dave Shea那项变红色（但人民邮电出版社不变）

first-of-type表示第一个这个类型（dd）的元素。

可以更改为last-of-type（最后一个）

nth-of-type(even) 表示偶数个

nth-last-of-type(2n)倒数的偶数项的元素

⑨选择在A标签范围内，只有一个B标签的B标签



如第一行有2个<span>，而第二行只有一个<span>

若对第二行CSS染色，使用

span:only-of-type{xxxx}

⑩不常用的

:empty 选中页面中没有子元素的元素

:root 选中根标签

:not() 不包含某个选择器的元素，参数是简单选择器

:target 选中被锚点选中的目标元素

:lang() language为某些特殊值的元素

简单选择器组合起来之后，表示两个条件都符合的元素。

**（18）设置文本框输入的最大字符数**

①属性：maxlength="15" 表示限制最多输入15个字符（中文或者数字、英文都是15个字符）。

**（19）min-width和max-width的替代**

在某些版本IE中（有一说是IE6和以下，但我自己实测IE8也不行），min-width和max-width不可用。

可用以下代码替代：

min-width:200px的替代

**\_width:expression((documentElement.clientWidth < 1000) ? "200px" : "auto" );**

max-width:1000px的替代

\_width:expression((documentElement.clientWidth >1000) ? "1000px" : "auto" );

解释：

①后面的auto根据实际需要使用。

②前面的documentElement.clientWidth按查到的说明，指的是屏幕空间宽度，**但实际应用中，不知为何，documentElement.clientWidth的值总为0**

**（20）绝对定位、定宽情况下，设置水平居中**

①思路：

首先绝对定位：position:absolute;

其次设置左方位置：left:50%;

然后定宽（假设300px）：width:300px;

让内容往左边移动宽带一半的位置：margin-left:-150px;

于是内容中间的位置，就在其父块的中间

②如果定高，垂直居中的话，同理。

**（21）滚动条的美化**

**for chrome内核：**

/\* 有这行才有效，滚动条的宽度 \*/

::-webkit-scrollbar {

width: 12px;

}

/\* 滚动条的背景 \*/

::-webkit-scrollbar-track {

-webkit-box-shadow: inset 0 0 6px rgba(0,0,0,0.3);

-webkit-border-radius: 10px;

border-radius: 10px;

}

/\*滚动条\*/

::-webkit-scrollbar-thumb {

-webkit-border-radius: 10px;

border-radius: 10px;

background: rgba(255,0,0,0.8);

-webkit-box-shadow: inset 0 0 6px rgba(0,0,0,0.5);

}

/\*貌似是切换窗口时显示的颜色\*/

::-webkit-scrollbar-thumb:window-inactive {

background: rgba(255,0,0,0.4);

}

更详细的说明：

http://blog.csdn.net/sweetsuzyhyf/article/details/8089158

**for IE：**

html {

overflow: auto;

SCROLLBAR-FACE-COLOR: #BED8EB;

SCROLLBAR-SHADOW-COLOR: #DDF8FF;

SCROLLBAR-HIGHLIGHT-COLOR: #92C0D1;

SCROLLBAR-3DLIGHT-COLOR: #DDF8FF;

SCROLLBAR-DARKSHADOW-COLOR: #92C0D1;

SCROLLBAR-TRACK-COLOR:#BED8EB;

SCROLLBAR-ARROW-COLOR: #92C0D1

}

**for firefox：**

firefox的需要使用插件。

这里放一个我copy到的内容，需要的时候找吧

<link rel="stylesheet" href="js/malihu-custom-scrollbar-plugin/jquery.mCustomScrollbar.min.css">  
<!-- latest jQuery direct from google's CDN -->  
<script type="text/javascript" src="js/jquery-1.11.1.min.js">  
</script>  
<script type="text/javascript" src="js/jquery-migrate-1.2.1.min.js">  
</script>  
<script src="js/malihu-custom-scrollbar-plugin/jquery.mCustomScrollbar.concat.min.js">  
</script>  
<script>  
 **if** (!$.browser.webkit) {  
 $.mCustomScrollbar.defaults.scrollButtons.enable = **true**; //enable scrolling buttons by default  
 $.mCustomScrollbar.defaults.axis = "yx"; //enable 2 axis scrollbars by default  
 $(".container").mCustomScrollbar({theme: "dark"});  
 }

**（22）Tab标签的内容区，高度自适应父容器**

假设：父div高度大小不确定，子div分为多层，一层需要自适应，其他固定高度

类似：

<**div** class="parent">  
 <**div** class="child">  
 <**div** class="tab">  
  
 </**div**>  
 <**div** class="content">  
   
 </**div**>  
 </**div**>  
</**div**>

父容器的高度不确定，并且是可变的；

我们需要child能填满父容器，并且tab的高度定高，content的高度不定高，content里面的内容可能较多，需要滚动条。

那么这么设置：

<**style**>  
 .child{  
 height:100%;  
 width:100%;  
 padding-top:20px;  
 }  
 .tab{  
 margin-top:-20px;  
 height:20px;  
 width:100%;  
 }  
 .content{  
 height:100%;  
 width:100%;  
 }  
</**style**>

是可以达到我们的要求的。

另外，只要child里还可以添加更多的定高标签，只要只有一个要求自适应即可，方法依然是调整父容器的padding属性，和其他定高/定宽标签的margin属性

**（23）textarea的拼写检查**

默认是有的，对拼写不正确的单词下面添加红色波浪线；

属性设置方法：

<标签名 **spellcheck=true|false**>

如果值是true，则检查（有红色波浪线）；

如果是false，则不检查。

例子：

<textarea spellcheck=false></textarea>

禁止拉伸：

在css属性里添加

resize: none

**（24）chrome开启账号密码自动填充后，背景色为黄色的问题**

http://www.tuicool.com/articles/EZ777

**（25）关于图片**

有时候会遇见这样的事情：

我们需要一个图片，他平常假设是20px大小，当我们鼠标移动上去之后，他的大小变为30px，且是以图片中心放大的。

实现方法是这样：

.img {  
 height: 20px;  
 width: 20px;  
 position: relative;  
}  
.img:hover {  
 height: 30px;  
 width: 30px;  
 top: -5px;  
 left: -5px;  
}

其关键是让图片的position属性变为relative，于是，图片首先没有脱离文档流，不存在其他dom结点占用了文档流的位置，又或者是图片的变大，把其他图片遮挡住了；

然后设置其变大后的宽高，并且向左移动增加宽度的一半，向上移动增加高度的一半，以确保结果是以图片原中心为中心变化的；

而:hover表示只有鼠标移动上去之后才会这样，鼠标没有移动上去的时候是保持原样的。

另外一个问题是，我们如何让图片出现在我们指定的位置，并且大小为我们指定的大小。

解决办法是，设置一个div，用这个div的contentbox区域来确定图片所在的位置，然后该div内只设置这一个图片，以避免不同图片在同一个div容易导致的冲突（比如位置不好确定）。

然后让图片填充满contentbox的区域；

注意：这个div的主要目的是为了定位图片的位置，之所以是contentbox区域，是因为避免和相邻div，或者其他什么东西所导致的冲突，而这些冲突有时候在遇见之前有时候是无法预料到的（虽然我一时也想不起来有什么，但是以一个良好的习惯来避免麻烦的发生，是一种明智的抉择）；

html代码大概是这样：

<**div** class="parent">  
 <**div** class="icon">  
 <**img** class="img" src="1.png">  
 </**div**>  
 <**div** class="icon">  
 <**img** class="img" src="2.png">  
 </**div**>  
 <**div** class="icon">  
 <**img** class="img" src="3.png">  
 </**div**>  
</**div**>

**（26）enctype属性**

<form enctype="*value*">

|  |  |
| --- | --- |
| **值value** | **描述** |
| application/x-www-form-urlencoded | 在发送前编码所有字符（默认） |
| multipart/form-data | 不对字符编码。在使用包含文件上传控件的表单时，必须使用该值。 |
| text/plain | 空格转换为 "+" 加号，但不对特殊字符编码。 |

**（27）自定义checkbox或者radio**

html代码：

<**label**>  
 <**input** type="checkbox" class="radio" name="TableChecked"/>  
 <**span**></**span**>  
 <**span**>选择磁盘</**span**> <!--这个是描述这个checkbox的label内容-->  
</**label**>

CSS代码：

.radio { /\*这个描述的是输入框的基本样式，透明，绝对定位，浮动在最下\*/  
 opacity: 0;  
 position: absolute;  
 z-index: -1;  
 /\*overflow: auto; 我找到的代码有这一行，但我自己试验后感觉并没有必要\*/  
}  
  
.radio[type=checkbox] + span { /\*如果类型是radio，那么这里就是type=radio，当然不写也可以。+span表示这个元素后的第一个span元素\*/  
 display: inline-block; /\*因为需要写span元素的大小\*/  
 height: 20px; /\*图标的大小\*/  
 width: 20px;  
 background-image: url(01.png);  
 background-position: center; /\*让背景图居中\*/  
 position: relative; /\*相对定位\*/  
 top: 3px; /\*可能需要根据实际情况来调整位置\*/  
 left: 5px;  
}  
  
.radio[type=checkbox]:checked + span { /\*选中时的样式\*/  
 background-image: url(02.png);  
}

**（28）禁止元素被选中**

在css的样式表里，添加这样属性（可兼容所有浏览器）

{

-moz-user-select: -moz-none;

-khtml-user-select: none;

-webkit-user-select: none;

-ms-user-select: none;

user-select: none;

}

关于IE10的：

/\*

Introduced in IE 10.

See http://ie.microsoft.com/testdrive/HTML5/msUserSelect/

\*/

**（29）label的用法**

用法一：（带id的）

<**input** type="checkbox" id="check">  
<**label** for="check">点击我，左边也可以被改变</**label**>

方法二：被label包含在一起的

<**label**><**input** type="checkbox" id="check">点击我，左边也可以被改变</**label**>

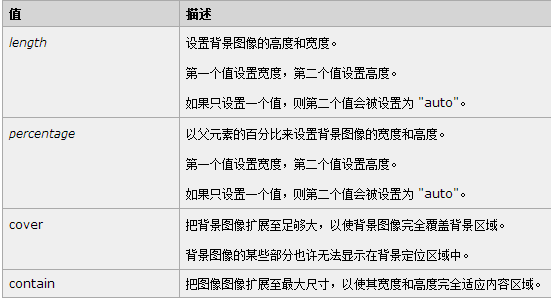
**（30）设置background-image大小**

背景图片的默认大小由图片来源决定；

假如我们需要调整其大小，有两种办法，更换为image标签，或者是使用

background-size属性

其值有多种：



【1】length顾名思义，就是100px之类；假如原有图像分辨率为30px\*30px，那么这里写成40px，图片就被放大了；

【2】百分比：这里指的是相对于其块的大小。假如一个div其大小为40px\*40px，背景图大小为16px\*16px，那么100%的话，就是把背景图变为40px\*40px大小；

【3】和【4】见描述，其中【3】是xy比例不变放大，【4】是拉伸至填满；

一般需要配合background-position:center来使用；让图片居中显示；

有两个参数，第一个是x轴的，第二个是y轴的，如果省略只写一个的话，应该是等比例y大小随之变化。按说明，只写一个y的属性值为auto；

**（31）使用padding来进行左定宽右自适应布局，及其遇见问题和解决办法**

①**情景：**

有一父容器div，其高和宽不定，称之为P；

该父容器有两个子div，左右布局，左定宽，满高，右自适应剩余区域；

其中，定宽称之为A，变宽称之为B；

A和B是等高的；

P必然能容纳A和B

**可能的附加条件：**

【1】A和B的宽高度可能不会撑满P，即上下左右都可能留有空隙，但这些空隙的宽或者高是已知的；

【2】A和B之间可能有一定间隙；

**备注：**

【1】左自适应右定宽方法同理；

【2】上下布局同理；

②**方法：**

原理：利用P的padding属性和A的margin属性来布局；

【1】由于P宽高不定，因此可以忽视对P的宽高设置；

【2】假设A的宽度为100px；A距离左侧10px，距离B有10px。因此，B的左侧实际有120px宽度；先设置P的css属性如下设置：

.P {

padding-left: 120px;

width: 略;

height: 略;

}

【3】设置A的CSS属性，如果要加border属性，那么需要注意A和B的box-sizing要设置为border-box

.A{

margin-left: -110px;

width: 100px;

box-sizing: borer-box;

float: left;

}

**Note：**

《1》如果有border属性，那么content区域要对应减少border的宽度（并且如果两侧都有，那么是双倍的宽度）；

《2》必须添加float属性，以使其脱离文档流；假如抛弃float属性，而采用display: inline-block属性的话，会导致右侧的B会靠近A，而不是在我们预想的区域之中。

【4】设置B的CSS属性，没有什么特殊的，只需要设置宽高为100%即可；

.B{

width: 100%;

height: 100%;

box-sizing: border-box;

}

③如此，便能实现左定宽，右变宽的布局了；并且由于没有使用CSS3属性，并且margin的范围没有超出P的盒模型，因此相对兼容性也很好。

④**可能存在的问题：**

【1】由于使用了float属性，也许在某些版本的浏览器中可能出现问题（真有这种可能？）；

解决办法：

取消A的float属性，用以下CSS替代：

display: inline-block; ——A和B都设置

vertical-align: top; ——A和B都设置

margin-right: 若干px; ——A设置，注意：这行的值可能并非10px，也许只有5px

**（31）CSS3属性pointer-events**

①语法：

**pointer-events**：auto | none | visiblepainted | visiblefill | visiblestroke | visible | painted | fill | stroke | all

**默认值**：auto

**适用于**：所有元素

**继承性**：有

**动画性**：否

**计算值**：指定值

②效果：

当属性值为none时，元素永远不会成为鼠标事件的目标。例如点击事件，打开新连接（a href='xxxx'）事件等。

如代码：

<!DOCTYPE html>

<html lang="zh-cmn-Hans">

<head>

<meta charset="utf-8" />

<title>pointer-events\_CSS参考手册\_web前端开发参考手册系列</title>

<meta name="author" content="Joy Du(飘零雾雨), dooyoe@gmail.com, www.doyoe.com" />

<style>

a[href="http://example.com"] {

pointer-events: none;

}

</style>

<script>

function abc(){

alert("1");

}

</script>

</head>

<body>

<ul>

<li><a onclick="abc()" >MDN</a></li>

<li><a onclick="abc()" href="http://example.com">一个不能点击的链接</a></li>

</ul>

</body>

</html>

上面这段代码，第二行文字既不会触发alert事件，也不会触发打开新链接的事件。

**（32）旋转的盒子**

①关键之处在于：

必须在box外面套一层壳，然后透视的css属性设置给他，否则旋转的时候，旋转的是box，那么透视角度会跟着box一起转，导致视角出错。

②html：

<**div** id="Box\_Background">  
 <**div** class="Box">  
 <**div** class="surface top">  
 top  
 </**div**>  
 <**div** class="surface bottom">  
 bottom  
 </**div**>  
 <**div** class="surface left">  
 left  
 </**div**>  
 <**div** class="surface right">  
 right  
 </**div**>  
 <**div** class="surface back">  
 back  
 </**div**>  
 <**div** class="surface front">  
 front  
 </**div**>  
 </**div**>  
</**div**>

③CSS代码：

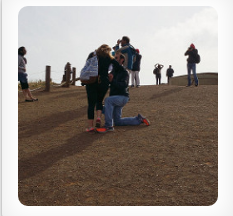
#Box\_Background {  
 position: absolute;  
 top: 300px;  
 left: 500px;  
 width: 300px;  
 height: 300px;  
 border: 1px solid #aaa;  
 perspective: 500px;  
 perspective-origin: 50% 50%;  
}  
  
.Box {  
 position: absolute;  
 top: 100px;  
 left: 100px;  
 width: 100px;  
 height: 100px;  
 text-align: center;  
 transform-style: preserve-3d;  
}  
  
.Box .surface {  
 position: absolute;  
 border: 1px solid black;  
 width: 100px;  
 height: 100px;  
 line-height: 100px;  
 text-align: center;  
}  
  
**@keyframes** top {  
 0% {  
 }  
 100% {  
 transform: rotateX(90deg) translateZ(50px)  
 }  
}  
  
**@keyframes** bottom {  
 0% {  
 }  
 100% {  
 transform: rotateX(-90deg) translateZ(50px)  
 }  
}  
  
**@keyframes** left {  
 0% {  
 }  
 100% {  
 transform: rotateY(-90deg) translateZ(50px)  
 }  
}  
  
**@keyframes** right {  
 0% {  
 }  
 100% {  
 transform: rotateY(90deg) translateZ(50px)  
 }  
}  
  
**@keyframes** front {  
 0% {  
 }  
 100% {  
 transform: translateZ(50px)  
 }  
}  
  
**@keyframes** back {  
 0% {  
 }  
 100% {  
 transform: translateZ(-50px) rotateY(180deg)  
 }  
}  
  
**@keyframes** all {  
 0% {  
 }  
 100% {  
 transform: rotate3d(1, 1, 1, 360deg)  
 }  
}  
  
**@keyframes** pp {  
 0% {  
  
 }  
}  
  
.Box .top {  
 animation: top 1s both;  
}  
  
.Box:hover .top {  
 background: red;  
}  
  
.Box .bottom {  
 animation: bottom 1s both 1s;  
}  
  
.Box:hover .bottom {  
 background: blue;  
}  
  
.Box .left {  
 animation: left 1s both 2s;  
}  
  
.Box:hover .left {  
 background: green;  
}  
  
.Box .right {  
 animation: right 1s both 3s;  
}  
  
.Box .Box:hover .right {  
 background: yellow;  
}  
  
.Box .back {  
 animation: back 1s both 4s;  
}  
  
.Box:hover .back {  
 background: pink;  
}  
  
.Box .front {  
 animation: front 1s both 5s;  
}  
  
.Box:hover .front {  
 background: GreenYellow;  
}  
  
.Box {  
 animation: all 3s both infinite 6s linear;  
}

**（33）css滤镜**

页面：

<http://html5-demos.appspot.com/static/css/filters/index.html>

原图：



①该滤镜让图片变为黑白

-webkit-filter: grayscale(1);  
filter: gray;  
filter: grayscale(1);

值0-1，越大灰白程度越高

②滤镜，模糊：

-webkit-filter: blur(3px);  
-moz-filter: blur(3px);  
-o-filter: blur(3px);  
-ms-filter: blur(3px);  
filter: blur(3px);

值越大越模糊。

③外阴影，可偏移：  
-webkit-filter: drop-shadow(1px 1px 20px black);



④有点像日照度提高：参数0~1

-webkit-filter: sepia(0.2);



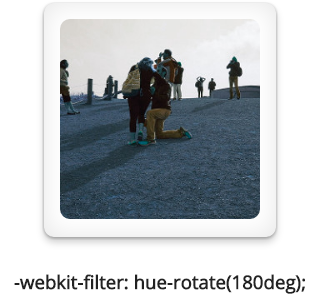
⑤似乎是明亮度，参数0~10



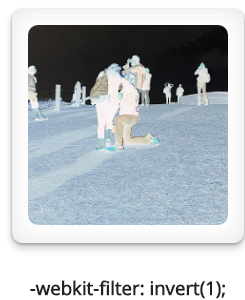
⑥变黑，参数0~10，1为原值



⑦这个应该是颜色转化，类似在调色板上的颜色变化，参数0~360deg，360deg时变为原色



⑧颜色倒置：



⑨色彩饱和度：



（10）透明度



**（34）上传文件后缀名限制**

引用：

<http://www.dewen.net.cn/q/3470>

<http://www.w3cschool.cn/htmltags/att-input-accept.html>

*可以使用accept属性  
<input type="file" accept="image/gif, image/jpeg" >  
如果不限制图像的格式，可以写为：accept="image/\*"。  
accept：MIME\_type 用逗号隔开的 MIME 类型列表。*

注：

第二个链接更全一些

|  |  |
| --- | --- |
| **值** | **描述** |
| audio/\* | 接受所有的声音文件。 |
| video/\* | 接受所有的视频文件。 |
| image/\* | 接受所有的图像文件。 |
| *MIME\_type* | 一个有效的 MIME 类型，不带参数。请参阅 [IANA MIME 类型](http://www.iana.org/assignments/media-types/)，获得标准 MIME 类型的完整列表。 |

（35）nth伪类选择器

1. **:first-child**选择某个元素的第一个子元素；
2. **:last-child**选择某个元素的最后一个子元素；
3. **:nth-child()**选择某个元素的一个或多个特定的子元素；
4. **:nth-last-child()**选择某个元素的一个或多个特定的子元素，从这个元素的最后一个子元素开始算；
5. **:nth-of-type()**选择指定的元素；
6. **:nth-last-of-type()**选择指定的元素，从元素的最后一个开始计算；
7. **:first-of-type**选择一个上级元素下的第一个同类子元素；
8. **:last-of-type**选择一个上级元素的最后一个同类子元素；
9. **:only-child**选择的元素是它的父元素的唯一一个了元素；
10. **:only-of-type**选择一个元素是它的上级元素的唯一一个相同类型的子元素；
11. **:empty**选择的元素里面没有任何内容