后盾网 人人做后盾

www.houdunwang.com

CSS3

后盾网 2011-2016

<选择器>，<选择器>　　选择多个元素

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

span和p标签里面的文字全部变红

```
<span>span</span>
<p>p</p>
```

<选择器>　<选择器>　　第一个选择器的后代，且匹配第二个选择器

```
div span{color: red;}
```

span会变红

```
<div>
    <p><span>span</span></p>
</div>
```

<选择器>＞<选择器>　　第一个选择器的直接后代，且匹配第二个选择器

```
div>span{color: red;}
```

span不会变红

```
<div>
    <p><span>span</span></p>
</div>
```

子元素选择器

<选择器> + <选择器>　　紧挨第一个选择器，且匹配第二个选择器

```
div+span{color: red;}
```

第一个span会变红

```
<div>div</div><span>span</span><span>span</span>
```

<选择器> ~ <选择器>　　第一个选择器之后，且匹配第二个选择器

```
div~span{color: red;}
```

所有与div相邻的span兄弟元素都变红

```
<div>div</div><span>span</span><span>span</span>
```

子元素选择器

E[attr]　　包含指定属性

```
a[aid]{color: red;}
a[target]{color: green;}
```

第一个红色，第二个绿色

```
<a href="" aid="1">aid</a> <a href="" target="_blank">blank</a>
```

E[attr＝“text”]　　属性值为text的元素

```
input[type=text]{color: red;}
```

input里面的值会变成红色

```
<input type="text" name="" id="" value="input text"/>
```

属性选择器

E[attr^=“value”]　　以指定值开始的元素

```
a[target^=_]{background-color:yellow;}
target以_开头的a连接全部背景变为黄色
<a href=“http://www.baidu.com”>baidu</a>
<a href=“http://www.houdunwang.com" target="_blank">houdunwang</a>
<a href=“http://www.kuaixuewang.com" target="_parent">kuaixue</a>
```

E[attr$=“value”]　　以指定值结束的元素

```
div[class$=“test”]{background:red}
div的class中以test结尾的背景变为红色
<div class="first_test">第一个test</div>
<div class="second">第二个</div>
<div class="test">第三个</div>
<p class="test">第四个</p>
```

属性选择器

E[attr*=“value”]　　包含指定值的元素，无论其位置

```
div[class*=“test"]{background:red;}
```

div的class所有包含test的元素背景全部变红

```
<div class="first_test">1</div>
<div class="second">2</div>
<div class="test-third">3</div>
<p class="test">4</div>
```

E[attr~=“value”]　　元素字义了attr属性，属性值是以空格分格的多值，值为value

```
[title~=flower]{border:5px solid yellow;}
```

包含flower并且是空格分开的，第一个和第三个会有黄色边框

```
<img src="1.jpg" title="klematis flower" width="150" height="113" />
<img src="2.gif" title="flowers" width="224" height="162" />
<img src="3.jpg" title="flower landscape" width="160" height="120" />
```

属性选择器

tr:nth-child(2)　　tr父元素里的第2个子元素并且是tr

```
p:nth-child(2){background:#ff0000;}
先找p的父级是div,然后再找div里面第二个元素，并且为p
<div>
   <h1>1</h1>
   <p>2</p> //这个行会变红
   <p>3</p>
   <p>4</p>
   <p>5</p>
</div>
```

tr:nth-child(2n)　　匹配偶数行

```
p:nth-child(2n){background:#ff0000;}
先找p的父级是div,然后再找div里面偶数元素，并且为p
<div>
<h1>1</h1>
<p>2</p> //这个行会变红
<p>3</p>
<p>4</p>//这个行会变红
<p>5</p>
</div>
```

tr:nth-child(even)　　匹配第偶数tr行 和上面一样，只不过上面可以写3n，4n...

伪类选择器 带有:就叫伪

以下参考前面的偶数，下面两个是一样的

tr:nth-child(2n+1)　　匹配奇数tr

tr:nth-child(odd)　　匹配奇数tr行

tr:nth-last-child(2)　　选择父元素倒数第2个

```
p:nth-last-child(2){background:#ff0000;}
```

先找p的父级是div,然后再找div里面倒数第二个元素，并且为p

```
<div>
<p>1</p>
<p>2</p>
<p>3</p>
<p>4</p> //这个行会变红
<span>test</span>
</div>
```

伪类选择器

tr:nth-of-type (2)　　tr父元素的第2个tr元素

```
p:nth-of-type(2){background:#ff0000;}
```

先找到父级div,然后找p标签的第二个，和p:nth-child(2) 对比

```
<div>
<h1>1</h1>
<p>2</p>
<p>3</p>//这行会变红
<p>4</p>
<p>5</p>
</div>
```

tr:nth-last-of-type(2)　　选择倒数第2个tr

伪类选择器

tr:first-child　　tr父元素的第1个子元素

p:first-child{background:red}和p:nth-child(1){background:red;}是一样的

先找p的父级是div,然后再找div里面第一个元素，并且为p，下面代码没有一行会变红，因为子元素第一个是h1

```
<div>
<h1>1</h1>
<p>2</p>
<p>3</p>
<p>4</p>
<p>5</p>
</div>
```

tr:last-child　　tr父元素的最后1个子元素

伪类选择器

tr:first-of-type 同辈的第1个tr元素

和tr:nth-of-type (1)是一样的

tr:last-of-type 同辈的最后一个元素

和tr:nth-last-of-type (1)是一样的

p:only-child p是唯一子元素

```
p:only-child{background:#ff0000;}
```

找到p元素，找父级，如果只包含一个子集p就算匹配成功

```
<div><p>1</p></div> //变红
<div><p>2</p><span>test</span></div>
```

p:only-of-type 父元素特定类型的唯一子元素

```
p:only-of-type{background:#ff0000;}
```

找到p的父级，如果父级里面只包含一个p元素或者多个子元素但是只有一个p元素

```
<div><p>1</p></div> //变红
<div><p>2</p><span>3</span></div>//变红
<div><p>4</p><p>5</p></div>
```

伪类选择器

p:empty　　不含子节点的tr元素

```
p:empty
{
width:100px;
height:20px;
background:#ff0000;
}
<p></p> //变红
<p>1</p>
<p>2</p>
```

div p:not(.hd)　　div中除了.hd的p元素

```
p{color:#000000;}
:not(p){color:#ff0000;}
<h1>这是标题</h1> 红色
<p>这是一个段落。</p> 黑色
<p>这是另一个段落。</p> 黑色
<div>这是 div 元素中的文本。</div> 红色
```

伪类选择器

E:enabled	选择所有可用元素
E:disabled	选择不可用元素

```
input[type="text"]:enabled
{
background:red;
}
input[type="text"]:disabled
{
background:blue;
}
```

```
<form action="">
姓: <input type="text" value="马" /><br>
名: <input type="text" value="震宇" /><br>
国家: <input type="text" disabled="disabled" value="中国" /><br>
</form>
```

伪类选择器

E:checked　　选择所有选中元素

```
<style>
input:checked+span{color: red;}
</style>
选中字体颜色变为红色
<form action="">
<input type="radio" checked="checked" value="男" name="sex" />
<span>男</span>
<br>
<input type="radio" value="女" name="sex" />
<span>女</span>
</form>
```

伪类选择器

:invalid 验证失败的表单

```
<style>
input:invalid{border:2px solid red;}
</style>
<h3> :invalid 选择器实例演示。</h3>
<input type="email" value="supportEmail" />
<p>请输入合法 e-mail 地址，查看样式变化。</p>
```

:valid 验证成功的表单

```
<style>
input:valid{background-color: green;color:white;}
</style>
<h3> :valid 选择器实例演示。</h3>
<input type="email" value="support@exampel.com" />
<p>请输入非法 e-mail 地址，查看样式变化。</p>
```

伪类选择器

:required	有required属性的元素
:optional	没有required属性的元素
:focus	匹配获得焦点的元素

```
<style>
input:required{background-color: yellow;}
input:optional{background:red;}
input:focus{background:blue;color:white;}
</style>
<p>红色:<br><input></p>
<p>黄色:<br><input required></p>
//获得焦点之后就会变为蓝色背景白色文字
```

伪类选择器

p:first-line　　匹配文本块的首行

```
p:first-line{background-color:yellow;}
```

文本的第一行会变成黄色

```
<p>后盾网是专注于培养中国互联网顶尖PHP程序语言专业人才的专业型培训机构，拥有五年培训行业经验。后盾网拥有国内最顶尖的讲师和技术团队，团队成员项目经验均在8年以上，团队曾多次为国内外上市集团、政府机关的大型项目提供技术支持，其中包括新浪、搜狐、腾讯、宝洁公司、联想、丰田、工商银行、中国一汽等众多大众所熟知的知名企业。</p>
```

p:first-letter　　匹配文本块的首字母

```
p::first-letter{font-size:200%;color:red;}
```

p标签的第一个字母放大200%，颜色为红色

```
<p>My name is 马震宇.</p>
<p>I live in 中国.</p>
<p>My best friend is 后盾学员</p>
```

文本选择器

:before	在选中元素的内容前插入内容
:after	在选中元素的内容后插入内容

```
<p>bbs.houdunwang.com</p>
<div>kuaixuewang.com</div>
<style type="text/css">
  p:before{
    content: "后盾论坛 ";
  }
  div:after{
    content: "这是快学网网址";
  }
</style>
```

文本选择器

```
<style>
	/*隐藏input*/
	label input{
		display: none;
	}
	/*默认span*/
	label span{
		display: inline-block;
		width:30px;height:30px;
		border:solid 2px #999;
	}
	/*选中input后的span元素*/
	label input:checked+span{
		border:solid 2px #008000;
		line-height: 2em;
		overflow: hidden;
		text-align: center;
		color:red;
	}
	label input:checked~span:before{
		content:'√';
	}
</style>
<label>
	<input type="radio" name="c"/>
	<span></span>
</label>
<label>
	<input type="radio" name="c"/>
	<span></span>
</label>
```

选择器实例“个性的单选框”

rgba颜色：

- rgb表示红、绿、蓝、a表示透明度1为不透明，0.5为半透明

```
#div1~div {
  background: rgba(255,0,0,0.5);
}
```

也可以没有a的用法,但是不能设置透明度

```
body{
  background:rgb(255,0,0)
}
```

RBGA颜色设置

px:

使用具体像素点为单位，好处是比较稳定和精确。但在浏览器中放大或缩放浏览页面时会存在一个问题

em:

em是以父级大小为参考的单位。好处是字体可以自适应。但父元素标签发生改变时字体大小将不确定。

rem：

rem是相对于根元素<html>，这样就意味着，我们只需要在根元素确定一个参考值即可。任意浏览器的默认字体高都是16px。所有未经调整的浏览器都符合: 1em=16px。

示例：

```
html {font-size: 62.5%;/*10 ÷ 16 × 100% = 62.5%*/}
body {font-size: 1.4rem;/*1.4 × 10px = 14px */}
h1 { font-size: 2.4rem;/*2.4 × 10px = 24px*/}
```

也可以这样设置

在根元素html（font-size:10px的时候）1rem为10px；换算工具

http://pxtoem.com/

字体单位

-webkit-line-clamp:

是一个不规范的属性，它没有出现在 CSS 规范草案中。为了实现该效果，它需要组合其他外来的WebKit属性

为了实现该效果，它需要组合其他外来的WebKit属性。常见结合属性：

- display: -webkit-box; 必须结合 ， 将对象作为弹性伸缩盒子模型显示 。
- -webkit-box-orient 必须结合，设置或检索伸缩盒对象的子元素的排列方式 。
- text-overflow，用在多行文本的情况下，用省略号“...”隐藏超出范围的文本 。

限制在一个块元素显示的文本的行数

属性规定当文本溢出包含元素时发生的事情：

- text-overflow: clip | ellipsis;

参数说明：

clip	修剪文本。
ellipsis	显示省略符号来代表被修剪的文本。

文字样式

多行文本溢出：

```
<p style="
  overflow : hidden;
  text-overflow: ellipsis;
  display: -webkit-box;
  -webkit-line-clamp: 2;
  -webkit-box-orient: vertical;
">
overflow : hidden; 隐藏才能出现截取的效果text-overflow: ellipsis; 用在多行文本的情况下，用省略号“...”隐藏超出范围的文本 display: -webkit-box; 将对象作为弹性伸缩盒子模型显示 -webkit-line-clamp: 2; 几行开始隐藏文本-webkit-box-orient: vertical;设置或检索伸缩盒对象的子元素的排列方式 vertical为垂直分布，一般我们会设置个，还有水平分布等等
</p>
```

单行文本溢出：

```
<style>
        h2{
            width:100px;
            overflow: hidden;
            /*文字不换行 这样才能出省略号*/
            white-space: nowrap;
            text-overflow:ellipsis;
        }
</style>
<h2>我自北京后盾网PHP培训65班的学生</h2>
```

文本溢出

text-shadow：

- 向文本添加阴影

语法：text-shadow: h-shadow v-shadow blur color

说明：

h-shadow 水平阴影的位置。允许负值

v-shadow 垂直阴影的位置。允许负值

blur　　模糊距离（可选）

color　　阴影的颜色（可选）

```
#div1 {
  width: 300px;
  height: 300px;
  text-shadow: 5px 5px 10px #333;
}
```

文本阴影

box-shadow：

- 向元素添加盒子阴影

语法：box-shadow: h-shadow v-shadow blur spread color

说明：

h-shadow	水平阴影的位置。允许负值
v-shadow	垂直阴影的位置。允许负值
blur	模糊距离
spread	阴影的尺寸
color	阴影的颜色

```
#div1 {
  width:300px;height: 300px;
  box-shadow: -10px 10px 20px 2px #333;
}
```

盒子阴影

```
<style type="text/css">
     .button{
          background: #FCAD26;
          color: white;
          padding: 10px 20px;
          border: none;
          font-size: 15px;
          border-radius: 3px;
          box-shadow: 0 5px 2px 0px darkorange;
          cursor: pointer;
          position: absolute;
          left: 0;
          top: 0;
     }
     /*按钮按下*/
     .button:active{
          top: 2px;
          box-shadow: 0 0;
     }
</style>
<button class="button">开始游戏</button>
```

按钮实例

min-width	元素最小可接受的宽度
max-width	元素最大可接受的宽度
min-height	元素最小可接受的高度
max-height	元素最大可接受的高度

图片响应式

```
<img src="shanghai.jpg" alt="..." style="max-width: 100%;height: auto;">
```

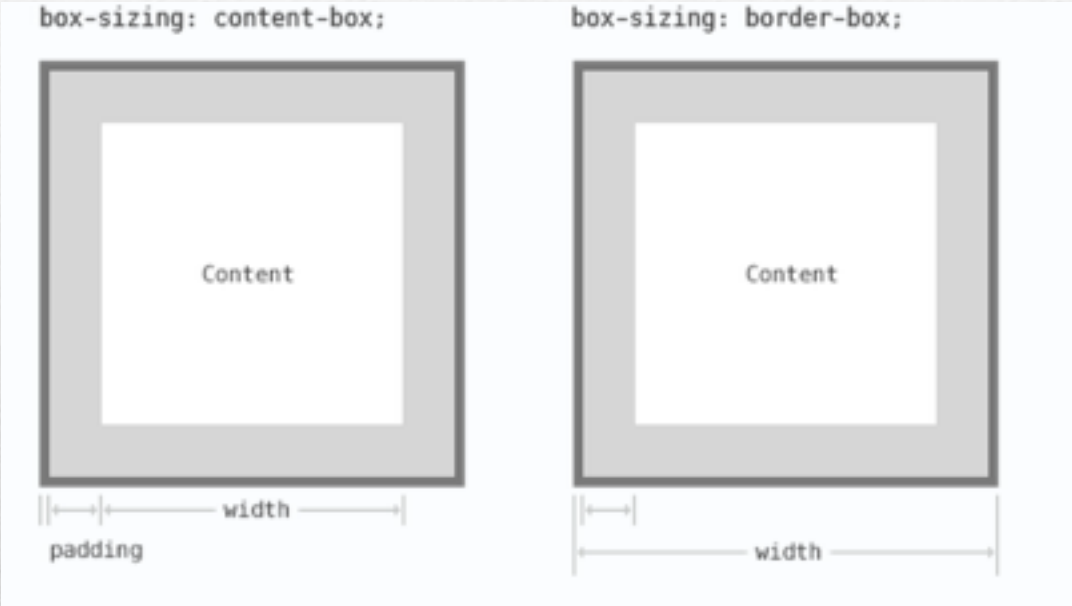
盒子尺寸

属性允许您以特定的方式定义匹配某个区域的特定元素

box-sizing: content-boxlborder-boxlinherit;

参数说明：

- content-box　border和padding不计算入width之内
- border-box　 border和padding计算入width之内，其实就是怪异模式了



盒子空间

border-box 优点：

- 无论如何改动 border 与 padding 的值，都不会导致 box 总尺寸发生变化，也就不会打乱页面整体布局。而在 Firefox 等现代浏览器下，如果我们要改变一下 padding 的值，就不得不重新计算 box 的 width。
- Firefox 请使用 -moz-box-sizing，Safari / WebKit 请使用 -webkit-box-sizing，Opera 直接用 box-sizing 即可。

盒子空间

-webkit-column-width　栏宽

-webkit-column-count　　列数

-webkit-column-gap　　列间距

-webkit-column-rule　　分隔线

说明：

-moz是针对firefox的设置，-webkit是针对chrome与苹果safari的设置

分栏

```
<style>
.newspaper {
     font-size:12px;
     -webkit-column-count: 3;
     -webkit-column-gap: 2rem;
     -webkit-column-rule: solid 1px #f00f00;
     -moz-column-count: 3;
     -moz-column-gap: 2rem;
     -moz-column-rule: solid 1px #f00f00;
}
</style>

<div class="newspaper">
“当我年轻的时候，我梦想改变这个世界；当我成熟以后，我发现我不能够改变这个世界，我将目光缩短了些，决定只改变我的国家；当我进入暮年以后，我发现我不能够改变我们的国家，我的最后愿望仅仅是改变一下我的家庭，但是，这也不可能。当我现在躺在床上，行将就木时，我突然意识到：如果一开始我仅仅去改变我自己，然后，我可能改变我的家庭；在家人的帮助和鼓励下，我可能为国家做一些事情；然后，谁知道呢?我甚至可能改变这个世界。”
</div>
```

分栏

background-size:

- 规定背景图像的尺寸

语法：

- background-size:宽度 高度

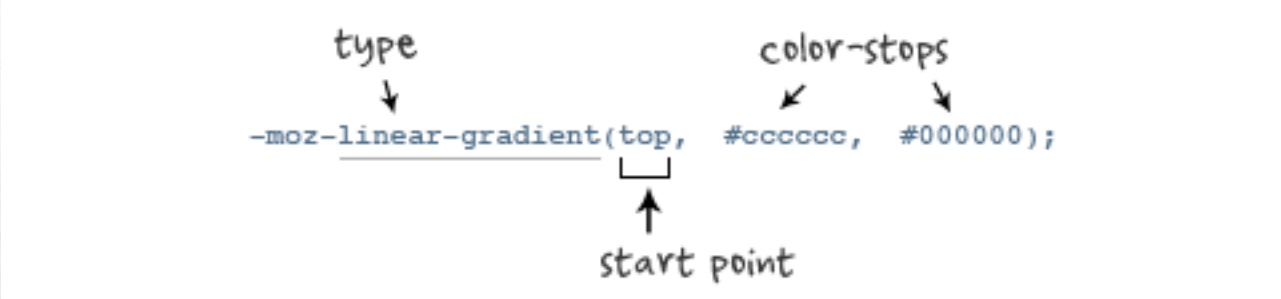
示例：

```
div{
width:500px;height: 500px;
 background: url(1.jpg);
 background-size:100px 100px;
}
```

更改背景图像大小

-webkit-linear-gradient:

- 语法：-webkit-linear-gradient([<point> ll <angle>,]? <stop>, <stop> [, <stop>]*)



示例：可以通过不断的改角度，看颜色移动的规律

```
<style type="text/css">
#d1{
width:100px;
height:100px;
background: -webkit-linear-gradient(-36deg,green,red,blue);
}
</style>
<div id="d1"></div>
```

线性渐变 -webkit-linear-gradient

-webkit-repeating-radial-gradient:

- 语法：-webkit-repeating-radial-gradient ([<point> ll <angle>,]? <stop>, <stop> [, <stop>]*)

shape 参数定义了形状。它可以是值 circle 或 ellipse。其中，circle 表示圆形，ellipse 表示椭圆形。默认值是 ellipse。

示例：

```
<style type="text/css">
#d1{
width:100px;
height:100px;
background:-webkit-repeating-radial-gradient(circle,black,#fff,black,#fff,black,#fff);
}
</style>
<div id="d1"></div>
```

径向渐变 -webkit-repeating-radial-gradient

outline**轮廓线**:

- outline （轮廓）是绘制于元素周围的一条线，位于边框边缘的外围，可起到突出元素的作用
- 轮廓线不同于边框，不占用空间尺寸。
- 只有在获得焦点或者激活时呈现

属性值:

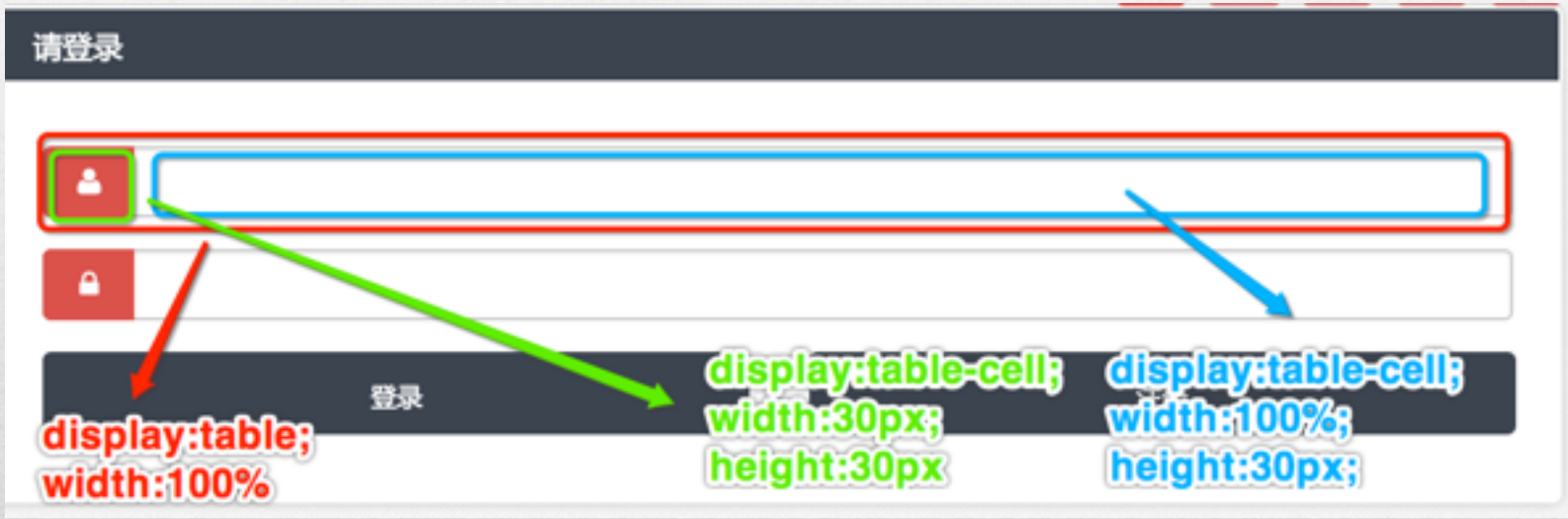
outline-color	规定边框的颜色
outline-style	规定边框的样式
outline-width	规定边框的宽度

示例:

```
<style type="text/css">
        [type=text]:focus{
        outline: solid 3px #f00;
}
</style>
<input type="text" />
```

outline轮廓线

任意缩放屏幕，左侧的红图标不变，右侧的input自动伸缩



display:table

```
<style type="text/css">
	.right{
		width: 100%;
		background: blue;
		overflow:hidden;
	}
	.right .input_box{
		display: table;
		width: 100%;
	}
	.right .input_box .btn{
		width: 30px;height: 30px;
		background: red;
		box-sizing: border-box;
		display: table-cell;
	}
	.right .input_box input{
		width: 100%; height: 30px;
		box-sizing: border-box;
		display: table-cell;

	}
</style>

<div class="right">
	<div class="input_box">
		<div class="btn"></div>
		<input type="text" name="" id="" />
	</div>
</div>
```

display:table