

CSS3 的规范包括：（html 规范 css 规范 javascript 规范）

JavaScript 规范包括：

ECMA 规范：ECMAScript—ES 就是基础，看规则版本 262 404 要掌握

DOM 规范：DOM-level xml 是严格的 HTML 里面的 HTML 都可以用密码

HTML 5 是一整套技术集，包括 CSS，JS，HTML 的技术

css3 是 css2.0 之后出现的所有 css 相关特性的集合，只有模块

CSS 包括哪些内容

选择器：

字体图标

新的 UI 方案

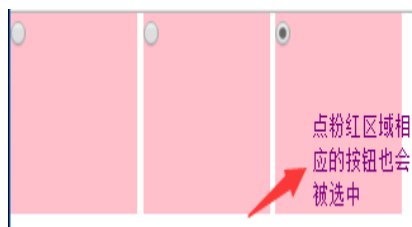
文本新增属性

布局扩展

过渡

2d/3d 变形

伪类的应用



同 radio 做一个特殊的 radio 根据 label 能绑定的表单元素，点击他的范围就能聚焦点到表单的数字（如下图解析），再由定位吧原来的 radio 点移到外面，在 label 中添加 span 来代替点，只要是点了粉红色，那个点就会被选中，span 只是代替的

```
label{
  position: relative;
  display: inline-block;
  width: 100px;;
  height: 100px;
  background: pink;
  border-radius: 50%;
}
```

```
label > input{
  position: absolute;
  left: -50px;
  top: -50px;
}
```

```
input > span{
  width: 100%;
  height: 100%;
  position: absolute;
  border-radius: 50%;//
```

变成一个圆

```

}
重点是 input 被选中时发
生的伪类，是后代 span 变背景
input:checked + span{
  background: deeppink;
}
```



通过 a 的伪类, **target** 设置一个锚点, 123 都是 a 标签, 在 a 标签跳转页里面填写黑色内容的 id, 点击之后把它的属性由 none 变为 block, 就可以显示了, 只会在点击才会变

```
.wrap{
position: relative;
margin: 0 auto;
height: 600px;
width: 600px;
//background-color: pink;
}
```

```
#box1,#box2,#box3{
position: absolute;
left: 0;
top: 30px;//让它们都根据父元素来定位到同一位置
```

```
background-color: black;
width: 100%;
height: 300px;
color: white;
display: none;
text-align: center;
}
```

```
:target{
display: block !important;
}
```

新的 UI 方案

文本新增样式:

opacity(默认值: 1.0 不可继承)

取值: 0-1, 1 是不透明

rgba()a 是设置图片的透明度, 和 **opacity** 差不多, 这个不回继承】

文本的属性

text-shadow(默认值: none 不可继承)

用来为文字添加阴影, 而且可以添加多层, 阴影值之间用逗号隔开。(多个阴影时, 第一个阴影在最上边值)

<offset-x> <offset-y>

必选。这些长度值指定阴影相对文字的偏移量。

<offset-x> 指定水平偏移量, 若是负值则阴影位于文字左边。

<offset-y> 指定垂直偏移量, 若是负值则阴影位于文字上面。

如果两者均为 0, 则阴影位于文字正后方(如果设置了 **<blur- radius>** 则会产生模糊效果)。属性指定了一个元素的透明度

eg:
text-shadow:10px 10px(两个是偏移量) 10px(阴影的程度) ， 颜色 10px（第二层也可以叠加）

<blur-radius>

可选。这是 <length> 值。如果没有指定，则默认为 0。
值越大，模糊半径越大，阴影也就越大越淡

border-radius（圆角）

50%以上的都是圆形

border-radius:50px ; 50px 圆的四个角都四周扯
border-radius:50px（左上角和右下角的扯/） 50px（右上角和左下角的扯\）
border-radius:50px（左上角） 50px(右上角和左下角/) 50px(右下角)
50px(右下角) 50px(右下角) 50px(左下角) 逆时针走

border-radius:50px(短半径)/60px(长半径)(左上) 画的是椭圆



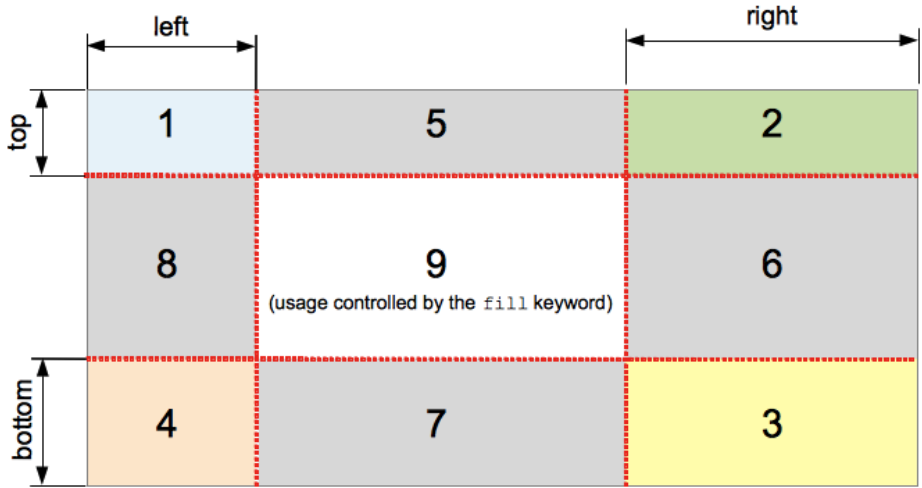
border-image-source（none 不可继承）

取值：

取值	None	url(image.jpg)	linear-gradient(to top, red, yellow)
说明	按照border来渲染	四个角都是这个img	四个角由颜色变成图片来渲染
图片说明			

border-image-slice （默认值100%，不可继承）

值得百分比参照于 image 本身!!
属性会通过规范将 border-image-source 的图片明确的分割为 9 个区域: 四个角, 四边以及中心区域。并可指定偏移量






中间的区域将不会被边框使用，但当设置有 fill 关键词时将会被作为 background-image。这个关键词可以被设置在属性的任何一个位置(前面、后面或者两个值之间)

取值	: 30%	: 30% 30%	: 30% 50% 60%	: 四个数
说明	图片按照图片的 30%来切割，四条边只有原图片 30%的大小	第一个是垂直，第二个是水平切割，对应边同，中间撑开	上边框的宽 top 水平切多少(水平) 下边框的宽(bottom)	top ， right ， bottom， left，
分析图				
运行图				注意：这里不需要加单位 px 之类的

fill 值的使用

中间的区域将不会被边框使用，但当设置有 **fill** 关键词时将会被作为 **background-image**。这个关键词可以被设置在属性的任何一个位置(前面、后面或者两个值之间) **border-image-slice**:

取值	: 30% fill	Repeat: stretch	Repeat: round
分析:	30%是按照如上切割 fill 是表示中间的不再空白	和 fill 连用，默认是 stretch，中间是拉伸的	同右边：不过中间是完整的显示并且平铺
运行图			

border-image-width （默认值：1 不可继承）

定义的是图片框内的图片大小，但是边框的大小不变

border-image-repeat:（默认值 stretch;）运用如上

border-image-outset(默认值：0 不可继承)

属性定义边框图像可超出边框盒的大小

正值： 可超出边框盒的大小 只有正值

背景图片

background-color 设置元素的背景

background-image:url() 设置元素的背景图片

如背景颜色小于元素大小，则会默认将图片平铺

可以同时为元素设置背景颜色和背景图片

background-repeat 设置元素的背景是否重复

repeat:默认值，背景图片会双方向重复（平铺）

no-repeat 背景图片不平铺

repeat-x 背景图片沿水平方向重复

repeat-y 背景图片沿垂直方向重复

background-position 设置元素的背景开始的位置，单值第二个值为 50%

可选用：top right left bottom 用两个值可以定位了

可以直接指定两个偏移量 background-position: 200px(水平方向偏移量) 180px;

background-attachment 用来设置背景是否随着页面滚动

scroll 默认值随着页面滚动

fixed 固定在页面某个位置，一般设置给页面背景

background: 以上都可以写，没有数量和顺序要求

background-origin，默认从padding开始，还可以box-sizing: box

background-clip设置背景图片大小(默认值: auto auto 不可继承)

background-size 设置背景图片大小(默认值: auto auto 不可继承)



取值	说明	取值	图片
百分比	指定背景图片相对背景区的百分比。背景区由 background-origin 设置，默认为盒模型的内容区与内边距,也可以用怪异模型	: 100%水平完整显示	
		: 100% 100% 水平垂直在盒子里都完整显示	
auto	盒子固定大小，那么图片就按照比例，从左上角开始显示	auto: 以背景图片的比例缩放背景图片。	

注意:

单值时，这个值指定图片的宽度，图片的高度隐式的为 auto
两个值：第一个值指定图片的宽度，第二个值指定图片的高度

渐变

为了创建一个线性渐变，你需要设置一个起始点和一个方向（指定为一个角度）的渐变效果。你还要定义终止色。终止色就是你想让浏览器去平滑的过渡过去，并且你必须指定至少两种，当然也可以指定更多的颜色去创建更复杂的渐变效果。

-默认从上到下发生渐变

`linear-gradient(red,blue);`双值渐变，从红色到蓝色

-改变渐变方向：（top bottom left right）

`linear-gradient(to 结束的方向,red,blue);`

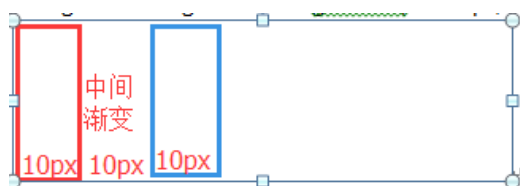
-使用角度：

`linear-gradient(角度,red,blue,yellow);`从什么变到什么 渐变的度数

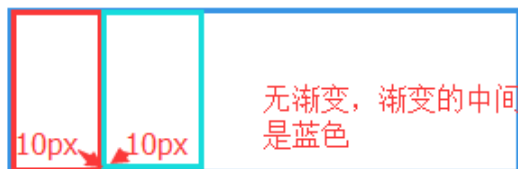
`linear-gradient(角度,red 20%,blue, yellow 80%);`红色占 20% 蓝色占 40%渐变的度数，80%开始是黄色

重复渐变：`repeat-linear-gradient(角度,red ,blue, yellow 80%);`重复的是渐变区
第一个值最好不要给值

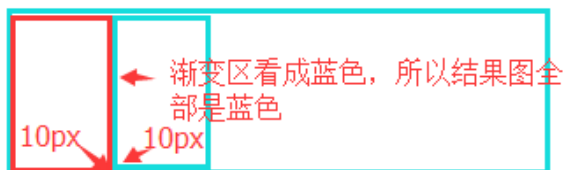
`background-image : linear-gradient(red 10px,blue 20px)`



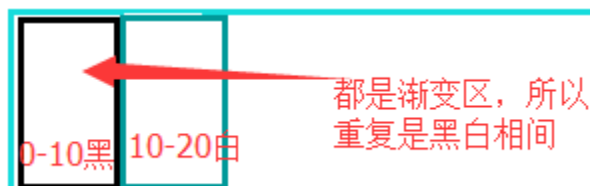
`background-image : linear-gradient(red 10px,blue 10px)`



`background-image : repeat-linear-gradient(red 10px blue 10px);`



`background-image:repeat-linear-gradien(block 0 ,block 10px ,white 10px white 20px);`



`radial-gradient(circle ,red,blue)`

`circle`

`ellipse`（默认为椭圆）

-渐变形状的大小

`radial-gradient(closest-corner circle ,red,blue)`

`closest-side` 最近边

`farthest-side` 最远边

`closest-corner` 最近角

`farthest-corner` 最远角

-改变圆心

`radial-gradient(closest-corner circle at 10px 10px,red,blue);`