1. 半透明border

border: 10px solid rgba(255, 255, 255, 0.5);

background-clip: padding-box;

background-clip为背景裁剪区域，默认border-box

1. 多层border
2. box-shadow：box-shadow: 0 0 0 10px red, 0 0 0 20px deeppink;偏移为0，多个用逗号隔开。此时创建的假border并不会响应鼠标事件。投影的行为跟边框不完全一致，因为它不会影响布局，而且也不会
3. 受到box-sizing 属性的影响。
4. outline：outline: green dashed 1px;只是用两层边框，在添加border的基础上设置outline。还可以通过outline-offset（可正可负）设置偏移。
5. 背景图片的位置
6. background-position：background-position: right 20px bottom 20px; 背景图位于右下角，并距右下角右边和下边各20px。css3中对于background-position有扩展，对于不支持次扩展的浏览器，要做响应的回退方案，将位置放在background属性中。
7. background-origin：background-origin: content-box; background-position 属性相对于什么位置来定位。默认为padding-box，这也是border不会遮挡背景的原因。但是如果背景图像的 background-attachment 属性为 "fixed"，则该属性没有效果。
8. calc()：background-position: calc(100% - 20px) calc(100% - 10px);此方案是以左上角偏移的思路来考虑。（请不要忘记在calc() 函数内部的- 和+ 运算符的两侧各加一个空白符，否则会产生解析错误！）
9. 边框内圆角

.border-radius-inside {

border-radius: 20px;

box-shadow: 0 0 0 10px pink;

outline: pink solid 10px;

}

描边(outline)并不会跟着元素的圆角走，所以会显示出直角。但是box-shadow会跟着圆角走显示出圆角。outline和box-shadow并不占据元素的空间位置，即不会算在width和height内。

1. 复杂背景
2. 条纹背景

只要每个色标的位置值相等，就可以实现无渐变的条纹

.background-line {

background: linear-gradient(yellow 50%, pink 50%);

background-size: 100% 30px;

}



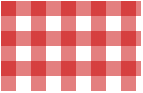
1. 网格背景
2. 使用linear-gradient，多个叠加使用

.background-line {

background: linear-gradient(90deg, rgba(200,0,0,.5) 50%, transparent 0), linear-gradient(rgba(200,0,0,.5) 50%, transparent 0);

background-size: 30px 30px;

}



1. 使用固定像素值—在某些情况下，我们希望网格中每个格子的大小可以调整，而网格线条的粗细同时保持固定。下边例子使用长度而不是百分比作为色标的场景

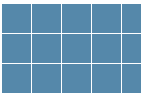
.background-grid {

background: #58a;

background-image: linear-gradient(white 1px, transparent 0), linear-gradient(90deg, white 1px, transparent 0);

background-size: 30px 30px;

}



1. 波点背景
2. 圆点

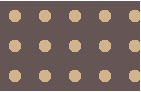
.background-spot {

background: #655;

background-image: radial-gradient(tan 30%, transparent 0);

background-size: 30px 30px;

}



1. 错列圆点—可以生成两层圆点阵列图案，并把它们的背景定位错开，这样就可以得到真正的波点图案了

.background-spot {

background: #655;

background-image: radial-gradient(tan 30%, transparent 0), radial-gradient(tan 30%, transparent 0);

background-size: 30px 30px;

background-position: 0 0, 15px 15px;

}



1. 棋盘

.background-spot {

background: #eee;

background-image: linear-gradient(45deg, #bbb 25%, transparent 0), linear-gradient(45deg, transparent 75%, #bbb 0), linear-gradient(45deg, #bbb 25%, transparent 0), linear-gradient(45deg, transparent 75%, #bbb 0);

background-size: 30px 30px;

background-position: 0 0, 15px 15px, 15px 15px, 0 0;

}

