CSS 指层叠样式表 (**C**ascading **S**tyle **S**heets)

1.样式表定义如何显示 HTML 元素，样式通常保存在外部的 .css 文件中。通过仅仅编辑一个简单 的 CSS 文档，外部样式表使你有能力同时改变站点中所有页面的布局和外观。

2.CSS声明总是以分号 (;) 结束，声明组以大括号 ({}) 括起来，p处为选择器:

p {color:red;

text-align:center;

}

3. id 和 class 选择器

如果你要在HTML元素中设置CSS样式，你需要在元素中设置"id" 和 "class"选择器。

Id选择器：HTML元素以id属性来设置id选择器, CSS 中 id 选择器以 "#" 来定义。

#para1{text-align:center;}

Class选择器：class 选择器用于描述一组元素的样式，class 选择器有别于id选择器，class可

以在多个元素中使用。类选择器以一个点"."号显示

所有： .center {text-align:center;}

或指定class="center" 的所有 <p> 元素 ： p.center {text-align:center;}

4.CSS创建

（1）外部样式表：每个页面使用 <link> 标签链接到样式表。浏览器从 mystyle.css读到样式

<head>  
 <link rel="stylesheet" type="text/css" href="mystyle.css">  
 </head>

（2）内部样式表：使用 <style> 标签在文档头部<head>定义内部样式表。

<style>  
 body {background-image:url("images/back40.gif");}  
 </style>

（3）内联样式：在相关的标签内使用样式（style）属性，Style 属性可以包含任何 CSS 属性。

<p style="color:sienna;margin-left:20px">This is a paragraph.</p>

优先级：内联样式>内部样式表>外部样式表>浏览器缺省设置

5.背景（background）

背景颜色：background-color: "#ff0000" or "rgb(255,0,0)" or “red”

背景图像：background-image: {background-image:url('paper.gif');}

平铺方式：background-repeat: 默认值: repeat ||repeat-x(只水平重复), repeat-y, no-repeat, inherit

随页面滚动background-attachment：默认值: scroll（随页面滚动）||fixed, inherit(从父元素继承）

起始位置：background-position：默认值： 0% 0%

简写属性： {背景颜色 背景图像 平铺方式 顺页面滚动 起始位置} 无需全部使用！

6.文本（Text）

文本颜色： color

文本对齐方式：text-align (center, left, right, justify (左右边距对其) )

文本修饰: text-decoration (none, overline, line-through, underline) 无,上,中,下线

文本转换:text-transform 大小写(uppercase, lowercase)每个单词首字大写(capitalize;)

文本缩进：p {text-indent:50px;}

7.字体（Fonts）

字体系列：{font-family:"Times New Roman", Times, serif;}

设置几个字体名称作为一种"后备"机制，如果浏览器不支持第一种字体，他将尝试下一种字体。

字体样式：font-style 正常：normal 斜体：italic 倾斜：oblique

字体大小： 像素：h1 {font-size:40px;}

用em设置：h1 {font-size:2.5em;}  1em的默认大小是16px

使用百分比和em组合 body {font-size:100%;} h1 {font-size:2.5em;}

不能通过调整字体大小使段落看上去像标题，务必使用正确的HTML标签，就<h1> - <h6>表示标题和<p>表示段落

字体粗细：font-weight bolder>bold(700)>normal(400)>lighter

8.链接（Link）

a:link {color:#FF0000;}      /\* 未访问链接\*/  
 a:visited {color:#00FF00;}  /\* 已访问链接 \*/  
 a:hover {color:#FF00FF;}  /\* 鼠标移动到链接上 \*/  
 a:active {color:#0000FF;}   /\* 鼠标点击时 \*/

需按顺序才能看见效果

Color处也可以放text-decoration，background-color 等属性

9.列表（ul，ol）

列表项标志：ul.a {list-style-type: circle;} square

ol.c {list-style-type: upper-roman;} lower-alpha

列表项标志位置：list-style-position outside（标志和内容相距远些）inside（近些）

列表项标志使用自定义图像：ul { list-style-image: url('sqpurple.gif'); }

缩写属性： 顺序：list-style-type，list-style-position，list-style-image

10.表格（Table） 表格包括 <table> 元素， 表格行[<tr>](http://www.runoob.com/tags/tag-tr.html)、单元表头[<th>](http://www.runoob.com/tags/tag-th.html)  单元格[<td>](http://www.runoob.com/tags/tag-td.html) 元素。

表格边框：table, th, td{border: 1px solid black;} 表格有双边框。因为表和th和td元素有独立的边界。

折叠边框：table{border-collapse:collapse;}

表格宽度与高度：table {width:100%;} th{height:50px;}

表格文字对齐： 水平：td{text-align:right;} 左，中，右。td{vertical-align:bottom;} 上，中，下

表格填充：td{padding:15px;}

表格颜色：th{background-color:green;}

11. CSS 盒子模型 (Box Model) : 所有HTML元素可以看作盒子

Margin(外边距) - 清除边框外的区域，外边距是透明的。

Border(边框) - 围绕在内边距和内容外的边框。

Padding(内边距) - 清除内容周围的区域，内边距是透明的。

Content(内容) - 盒子的内容，显示文本和图像。

元素的宽度和高度：

div { width: 300px; （width为content宽度）

border: 25px solid green;

padding: 25px;

margin: 25px;

}

总元素的宽度=宽度+左填充+右填充+左边框+右边框+左边距+右边距

总元素的高度=高度+顶部填充+底部填充+上边框+下边框+上边距+下边距

解决IE8及更早版本不兼容问题可以在HTML页面声明 <!DOCTYPE html>即可。

12.边框（Border）

边框样式：**border-style** none: 默认无边框 dotted: 定义一个点线框

dashed: 定义一个虚线框 solid: 定义实线边界

边框宽度：**border-width**  可以指定像素长度值，或者使用 3 个关键字：thick 、medium（默认值） 和 thin。

边框颜色：**border-color** 属性： "red"，"rgb(255,0,0)"，"#ff0000" 或"transparent"（透明）

单独设置个边框：

p { border-top-style:dotted;

border-right-style:solid;

border-bottom-style:dotted;

border-left-style:solid; }

也可设置单一属性：border-style:dotted solid;

border-style属性可以有1-4个值：

border-style:dotted solid double dashed; border-style:dotted solid double;

上边框是 dotted 上边框是 dotted

右边框是 solid 左、右边框是 solid

底边框是 double 底边框是 double

左边框是 dashed

border-style:dotted solid; border-style:dotted;

上、底边框是 dotted 四面边框是 dotted

右、左边框是 solid

简写属性："border"属性中设置：border-width border-style (required) border-color

例如：border:5px solid red;

13.轮廓（outline）: 轮廓是绘制于元素周围的一条线，位于边框（Border）边缘的外围, 属性有样式、颜色和宽度。

轮廓样式：[outline-style](http://www.runoob.com/cssref/pr-outline-style.html)

轮廓宽度：[outline-width](http://www.runoob.com/cssref/pr-outline-width.html)

轮廓颜色：[outline-color](http://www.runoob.com/cssref/pr-outline-color.html)

简写属性：[outline](http://www.runoob.com/cssref/pr-outline.html)：outline-color outline-style outline-width inherit

14.外边距（Margin）：没有背景颜色，是完全透明的。

可单独设置每个边距：margin-top:100px;

margin-bottom:100px;

margin-right:50px;

margin-left:50px;

## 简写属性：margin：（属性可以有1-4个值，和边框（Border）用法一样）

15. 填充 （Padding） ：元素边框（Border）与元素内容（Content）之间的空间。

可单独设置每个边距：padding -top:100px;

padding -bottom:100px;

padding -right:50px;

padding -left:50px;

## 简写属性：padding：（属性可以有1-4个值，用法同上）

**16.**分组选择器（Grouping Selectors）： 嵌套选择器 ：

每个选择器用逗号分隔 为所有class="marked"元素内的p元素指定一个样式

h1,h2,p .marked p  
 { {  
 color:green; color:white;  
 } }

17.尺寸（Dimension）：允许你控制元素的高度和宽度。同样，它允许你增加行间距。

|  |  |
| --- | --- |
| [height](http://www.runoob.com/cssref/pr-dim-height.html) | 设置元素的高度。 |
| [line-height](http://www.runoob.com/cssref/pr-dim-line-height.html) | 设置行高。 |
| [max-height](http://www.runoob.com/cssref/pr-dim-max-height.html) | 设置元素的最大高度。 |
| [max-width](http://www.runoob.com/cssref/pr-dim-max-width.html) | 设置元素的最大宽度。 |
| [min-height](http://www.runoob.com/cssref/pr-dim-min-height.html) | 设置元素的最小高度。 |
| [min-width](http://www.runoob.com/cssref/pr-dim-min-width.html) | 设置元素的最小宽度。 |
| [width](http://www.runoob.com/cssref/pr-dim-width.html) | 设置元素的宽度。 |

18. 显示（Display）和可见性（Visibility）

隐藏一个元素可以通过把display属性设置为"none"，或把visibility属性设置为"hidden"。

h1.hidden {visibility:hidden;}

visibility:hidden可以隐藏某个元素，但隐藏的元素仍需占用与未隐藏之前一样的空间。

h1.hidden {display:none;}

display:none可以隐藏某个元素，且隐藏的元素不会占用任何空间。

块元素：占用了全部宽度，在前后都是换行符。如：<h1> <p> <div>

内联元素：只需要必要的宽度，不强制换行。 如<a>

display 属性的值为 "inline"：转换为内联函数。 "block"：转换为块元素。

19.定位（Positioning）： 使用position属性允许你为一个元素定位。

Absolute：绝对定位的元素的位置相对于最近的已定位父元素，如果元素没有已定位的父元素，那么它的位置相对于

<html>:   
 fixed：元素的位置相对于浏览器窗口是固定位置。即使窗口是滚动的它也不会移动。

p.pos\_fixed  
 {  
 position:fixed;  
 top:30px;  
 right:5px;  
 }  
 relative：相对定位元素的定位是相对其正常位置。  
 static：HTML元素的默认值，即没有定位，元素出现在正常的流中。  
 inherit：元素可以覆盖页面上的其它元素，z-index属性指定了一个元素的堆叠顺序（哪个元素应该放在前面，或后

面）如果两个定位元素重叠，没有指定z - index，最后定位在HTML代码中的元素将被显示在最前面。

img{

position:absolute;

left:0px;

top:0px;

z-index:-1;

}

## CSS定位属性：

## top, bottom, left, right: 元素和上下左右的距离（值：auto，px，%，inherit）

20.浮动（Float）：会使元素向左或向右移动，直到它的外边缘碰到包含框或另一个浮动框的边框为止，浮动元素之前的

元素将不会受到影响，浮动元素之后的元素将围绕它，往往是用于图像。

img { float:right; } 或left，none，inherit

如果把几个浮动的元素放到一起，如果有空间的话，它们将彼此相邻。

清楚浮动： 使用clear属性。clear 属性指定元素两侧不能出现浮动元素。

.text\_line { clear:both; } left，right，none，inherit

21.水平对齐（Horizontal Align）

块元素：占用全宽，前后自动换行。 如<h1>, <p>, <div>

中心对齐：margin:auto; 如果width：100%，则没有效果。

左右对齐： 使用position属性 ： 如右对齐：{ position:absolute;

right:0px;

width:300px;

}

使用float属性： float:right或left，如果不设定width，则没有效果。

22.组合选择符：组合选择符说明了两个选择器直接的关系。

后代选择器：E F （匹配所有属于E元素里边的F元素，E和F之间用空格分隔）

div p { background-color:yellow; }

子元素选择器： E>F（匹配所有E元素的子元素F），F必须直接在E里边

相邻兄弟选择器：E+F 匹配所有紧随E元素之后的同级元素F，E和F要有相同父元素。

普通相邻兄弟选择器：E~F 匹配所有在E元素之后的同级F元素。

23.伪类（Pseudo-classes）：伪类是用来添加一些选择器的特殊效果

语法： selector:pseudo-class {property:value;}

或 selector.class:pseudo-class {property:value;}

Anchor伪类：a:link {color:#FF0000;} /\* 未访问的链接 \*/  
 a:visited {color:#00FF00;} /\* 已访问的链接 \*/  
 a:hover {color:#FF00FF;} /\* 鼠标划移动到链接上 \*/  
 a:active {color:#0000FF;} /\* 鼠标点击链接时 \*/

:first-child伪类： 匹配父元素的第一个子元素

例：选择器匹配作为任何元素的第一个子元素<p> p:first-child{color:blue;}

选择相匹配的所有<p>元素中的第一个 <i> 元素 p > i:first-child{color:blue;}

选择器匹配所有元素的第一个子元素 <p> 中的所有 <i> 元素：p:first-child i{color:blue;}

E:lang(c)伪类 ：匹配lang属性等于c的E元素

例:匹配属性值为 no 的q元素定义引号的类型： q:lang(no) {quotes: "~" "~";}

其中：<q lang="no">A quote in a paragraph</q>

24.伪元素

语法 ： selector:pseudo-element {property:value;}

或selector.class:pseudo-element {property:value;}

E:first-line 匹配E元素的第一行   
 E:first-letter 匹配E元素的第一个字母

E:before 在E元素之前插入生成的内容

例：在每个 <h1>元素前面插入一幅图片： h1:before {content:url(smiley.gif);}

其中content属性与 :before 以及 :after 伪元素配合使用，来插入生成内容。

E:after 在E元素之后插入生成的内容

25.导航栏 ：使用 <ul> 和 <li>元素建立导航栏

<ul>  
 <li><a href="default.asp">Home</a></li>  
 <li><a href="news.asp">News</a></li>  
 <li><a href="contact.asp">Contact</a></li>  
 <li><a href="about.asp">About</a></li>  
 </ul>

删除列表前小标志，边距，填充：ul{ list-style-type:none;  
 margin:0;  
 padding:0; }

垂直导航栏: a{ display:block; 整体变为块元素，允许我们制定宽度。

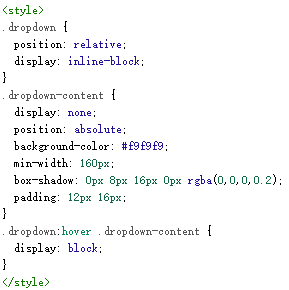
width:60px;} 块元素默认情况为最大宽度

水平导航栏：使用**内联**或**浮动**的列表项。

1. 内联列表项 li{display:inline;} （默认情况下，<li>元素是块元素。）
2. 浮动列表项 浮动 <li>元素，并指定为 <a>元素的宽度：

li{float:left;}  
 a{display:block;

width:60px;}



26.下拉菜单

HTML：<div class="dropdown"> CSS:

<span>鼠标移动到我这！</span>

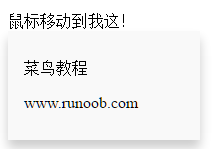
<div class="dropdown-content">

<p>菜鸟教程</p>

<p>www.runoob.com</p>

</div>

</div>



**HTML 部分：**

我们可以使用任何的 HTML元素来打开下拉菜单，如：<span>, 或 <button> 元素。

使用容器元素 (如： <div>) 来创建下拉菜单的内容（菜鸟教程 & www.runoob.com），并放在任何你想放的位置上。

使用 <div> 元素来包裹这些元素，并使用 CSS 来设置下拉内容的样式。

**CSS 部分：**

.dropdown 类使用 position:relative, 这将设置下拉菜单的内容放置在下拉按钮 (使用 position:absolute) 的右下角位置。

display:inline-block: 对象呈现为inline对象，但是对象的内容(此处为菜单的内容)作为block对象呈现,

之后的内联对象会被排列在同一行内（和鼠标移动到我这!同行）.

.dropdown-content 类中是实际的下拉菜单。默认是隐藏的，在鼠标移动到指定元素后会显示。 注意 min-width 的值设置为 160px。你可以随意修改它。 注意: 如果你想设置下拉内容与下拉按钮的宽度一致，可设置 width 为 100% ( overflow:auto 设置可以在小尺寸屏幕上滚动)。

我们使用 box-shadow 属性让下拉菜单看起来像一个"卡片"。(添加阴影)

:hover 选择器用于在用户将鼠标移动到下拉按钮上时显示下拉菜单。

如果希望鼠标点到的时候有反应：.dropdown-content p:hover {background-color: #f1f1f1}

27. 图像透明/不透明

img { opacity:0.4; % Opacity属性值从0.0 - 1.0。值越小，使得元素更加透明。

或 filter:alpha(opacity=40); }IE8及其更早版本，值是从0 – 100，较低的值，元素更加透明

28. 图像合并技术

有许多图像的网页可能需要很长的时间来加载和生成多个服务器的请求。

使用图像拼合会降低服务器的请求数量，并节省带宽。

只显示图像一部分： img.home{ width:46px;  
 height:44px;  
 background:url(img\_navsprites.gif) 0 0;}

<img class="home" src="img\_trans.gif" />

因为不能为空, src属性只定义了一个小的透明图像。显示的图像将是我们在CSS中指定的背景图像

宽度：46px;高度：44px; - 定义我们使用的那部分图像

background:url(img\_navsprites.gif) 0 0; - 定义背景图像和它的位置（左0px，顶部0px）

29.媒体类型：媒体类型允许你指定文件将如何在不同媒体呈现。如屏幕上，在纸张上，或听觉浏览器等等。

sans - serif字体比较适合在屏幕上阅读，而serif字体更容易在纸上阅读。

@media 规则允许在相同样式表为不同媒体设置不同的样式。

@media screen （用于电脑显示器）

{ p.test {font-family:verdana,sans-serif;font-size:14px;} }

@media print （用于打印机）

{ p.test {font-family:times,serif;font-size:10px;} }

30.属性选择器

把所有包含title的元素变为蓝色： [title] { color:blue; }

改变了标题title='runoob'元素的边框样式: [title=runoob] { border:5px solid green; }

title属性包含指定值的的元素： [title~=hello] { color:blue; }

或：[lang|=en] { color:blue; }

表单样式： input[type="text"]

{ width:150px;

display:block; }

<form name="abc" >

Firstname:<input type="text" name="fname" value="Peter" size="20">

</form>

# CSS3

1.css 边框

（1）圆角边框：border-radius

div  
 { border:2px solid;  
 border-radius:25px; }

（2）添加阴影：box-shadow

例：box-shadow: 10px 10px 5px #888888; 前两个是相对位置，第三个是模糊程度，第四个是颜色

（3）边界图片：border-image属性，可以使用图像创建一个边框

例：平铺：border-image:url(border.png) 30 10 round;

拉伸：border-image:url(border.png) 30 30 stretch;

2.圆角：使用 CSS3 border-radius 属性（border-radius: 25px;）

**四个值:** **第一个值为左上角，第二个值为右上角，第三个值为右下角，第四个值为左下角。**

**三个值:** 第一个值为左上角, 第二个值为右上角和左下角，第三个值为右下角

**两个值:** 第一个值为左上角与右下角，第二个值为右上角与左下角

**一个值：** 四个圆角值相同

[border-top-left-radius](http://www.runoob.com/cssref/css3-pr-border-top-left-radius.html) 定义了左上角的弧度

[border-top-right-radius](http://www.runoob.com/cssref/css3-pr-border-top-right-radius.html) 定义了右上角的弧度

[border-bottom-right-radius](http://www.runoob.com/cssref/css3-pr-border-bottom-right-radius.html) 定义了右下角的弧度

[border-bottom-left-radius](http://www.runoob.com/cssref/css3-pr-border-bottom-left-radius.html) 定义了左下角的弧度

3.背景

background-image： 添加背景图片，不同的背景图像和用逗号隔开，第一张显示在最顶端

例：background-image: url(img\_flwr.gif), url(paper.gif);

background-position: right bottom, left top;

background-size：指定背景图像的大小，可以指定像素或百分比大小（宽度和高度）。

background-size : 80px 60px ; 或 background-size : 100% 100% ;

background-Origin：指定background-position属性应该是相对位置。

属性： padding-box 背景图像填充框的相对位置

border-box 背景图像边界框的相对位置

content-box 背景图像的相对位置的内容框

background-clip：从指定位置开始（向内）绘制

属性也是padding-box，border-box，content-box

4.渐变（Gradients）：两个或多个指定的颜色之间显示平稳的过渡。

CSS3 定义了两种类型的渐变（gradients）：

线性渐变（Linear Gradients）- 向下/向上/向左/向右/对角方向

径向渐变（Radial Gradients）- 由它们的中心定义

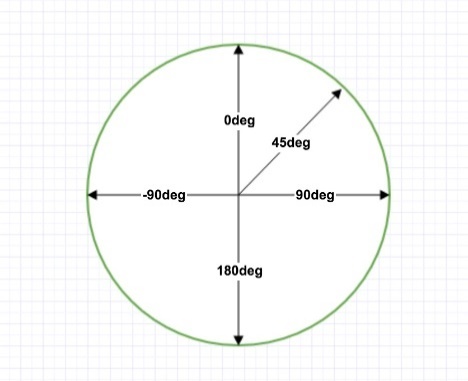
（1）线性渐变：至少定义两种颜色结点，同时，也可以设置一个起点和一个方向。

语法：background: linear-gradient(*direction*, *color-stop1*, *color-stop2, ...*);

**从上到下（默认情况下）：**background: linear-gradient(red, blue);

**从左到右：**background: linear-gradient(to right, red , blue);

**对角（左上到右下）：**background: linear-gradient(to bottom right, red , blue);

 使用角度：background: linear-gradient(*angle*, *color-stop1*, *color-stop2*);

上到下：background: linear-gradient(180deg, red, blue);

使用多个颜色结点：background: linear-gradient(red, green, blue);

彩虹：background: linear-gradient(to right, red, orange, yellow, green,

blue, indigo, violet);

使用透明度（transparency）：可用于创建减弱变淡的效果，rgba( ) 函数来定义

颜色结点。函数的最后一个参数是从 0 到 1 的值，0 表示完全透明，1 表示完全不透明。

例：从左边开始的线性渐变。起点是完全透明，慢慢过渡到完全不透明的红色

background: linear-gradient(to right, rgba(255,0,0,0), rgba(255,0,0,1));

重复的线性渐变：repeating-linear-gradient()

（2）径向渐变：至少定义两种颜色，可以指定渐变的中心、形状（原型或椭圆形）、大小。默认情况渐变的

中心是 center，渐变的形状是 ellipse（椭圆形），渐变的大小是 farthest-corner（到最远的角落）。

语法：background: radial-gradient(center, shape size, start-color, ..., last-color);

颜色结点均匀分布（默认情况下）**：**background: radial-gradient(red, green, blue);

颜色结点不均匀分布：background: radial-gradient(red 5%, green 15%, blue 60%);

设置形状：shape设置形状。可以是circle 或 ellipse。circle是圆形，ellipse是椭圆形。默认值是 ellipse

不同尺寸大小关键字：closest-side，farthest-side，closest-corner，farthest-corner

重复的径向渐变：repeating-radial-gradient( )

5.文本效果

文本阴影：text-shadow：

例：text-shadow: 5px 5px 5px #FF0000; 前两个是相对文字位置，第三个是模糊程度，第四个是颜色

盒子阴影：box-shadow:

例：box-shadow: 10px 10px 5px #888888; 前两个是相对盒子位置，第三个是模糊程度，第四个是颜色

Text-Overflow: 指定应向用户如何显示溢出内容（属性：clip 截短，ellipsis 使用省略号）

自动换行：{word-wrap:break-word;}

单词拆分：word-break（属性：keep-all 不许拆分，break-all 可以拆分）

## 6.字体：使用CSS3，我们可以使用自己喜欢的任何字体。只需简单的将字体文件包含在网站中，它会自动下载给

## 需要的用户。所选择的字体在新的CSS3版本有关于@font-face规则描述。

## @font-face 规则中，必须首先定义字体的名称（比如 myFirstFont），然后指向该字体文件。

## font-family ：用来引用字体的名称

@font-face { font-family: myFirstFont;

src: url(sansation\_light.woff); }

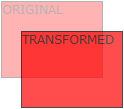
div { font-family:myFirstFont; }

## 使用粗体：设置 "myFirstFont" 应该呈现的粗体样式

@font-face { font-family: myFirstFont;

src: url(sansation\_bold.woff);

font-weight:bold; }

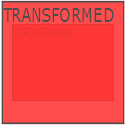


7. 2D转换 ：移动，比例化，反过来，旋转，和拉伸元素。使用transform 属性.

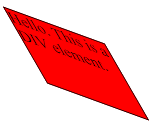
translate( )：根据左(X轴)和顶部(Y轴)位置给定的参数，从当前元素位置移动。

 transform: translate(50px,100px);

rotate( )：给定一个顺时针旋转角度。负值代表逆时针旋转。

 transform: rotate(30deg); （30度）

scale( )：增加或减少元素的大小，取决于宽度（X轴）和高度（Y轴）的参数：

 transform: scale(2,3); 宽度变2倍，高度变3倍

skew ( )：transform:skew(<angle> [,<angle>]);分别表示X轴和Y轴倾斜的

角度，如果第二个参数为空，则默认为0，参数为负表示向相反方向倾斜

transform: skew(30deg,20deg); (X轴和Y轴上倾斜20度和30度。)

matrix( )：和2D变换方法合并成一个。

matrix 方法有六个参数，包含旋转，缩放，移动（平移）和倾斜功能。

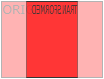
transform:matrix(0.866,0.5,-0.5,0.866,0,0);



8. 3D转换 同样使用transform 属性.

rotateX( )：围绕其在一个给定度数X轴旋转的元素。

例：transform: rotateX(120deg);

 rotateY( )：围绕其在一个给定度数Y轴旋转的元素。

例：transform: rotateY(130deg);

9. 过渡：元素从一种样式逐渐改变为另一种的效果。使用transition属性 （简写属性）

必须规定两项内容：1.指定要添加效果的CSS属性

2.指定效果的持续时间。

例：鼠标移动到元素时，宽度属性改变，时长为 2 秒：

div { transition: width 2s; } /如果未指定期限，transition将没有渐变效果（直接变），默认值是0。

div:hover { width:300px; } /指定的CSS属性的值更改时效果会发生变化。

多项改变：添加属性用逗号分隔

例：div { transition: width 2s, height 2s, transform 2s; }

div:hover { width: 200px;

height: 200px;

transform: rotate(180deg);}

10.动画 ：@keyframes规则创建动画。 @keyframes指定一个动画使元素从一种样式逐渐变化为另一种样式

例：div{ width:100px;

height:100px;

background:red;

animation:myfirst 5s; }

@keyframes myfirst{

from {background:red;}

to {background:yellow;} }

可以改变任意多的次数。用百分比来规定变化发生的时间，或用 "from" 和 "to"等同于 0% 和 100%。

例: @keyframes myfirst {0% {background:red; left:0px; top:0px;}

25% {background:yellow; left:200px; top:0px;}

50% {background:blue; left:200px; top:200px;}

75% {background:green; left:0px; top:200px;}

100% {background:red; left:0px; top:0px;} }

11.多列：像报纸一样多列布局

创建多列：column-count 指定了分割列数。 例：column-count: 3;

列与列间隙：column-gap  例： column-gap: 40px;

列边框：样式：column-rule-style  例：column-rule-style: solid;

厚度：column-rule-width 例：column-rule-width: 1px;

颜色：column-rule-color 例：column-rule-color: lightblue;

简写属性：column-rule  例：column-rule: 1px solid lightblue;

指定元素跨越多少列：column-span 例：column-width: 100px;

指定列宽：column-width  例：column-width: 100px;

12.用户界面：用来调整元素尺寸，框尺寸和外边框。

resize：指定一个元素是否是由用户调整大小，默认none ( resize:none;)

none 用户无法调整元素的尺寸。

both 用户可调整元素的高度和宽度。

horizontal 用户可调整元素的宽度。

vertical 用户可调整元素的高度。

box-sizing: 定义某些元素，以适应指定区域。默认content-box ( box-sizing: content-box )

content-box 指定元素的宽度和高度（最小/最大属性）适用于box的宽度和高度。

元素的填充和边框布局和绘制指定宽度和高度除外

border-box 指定宽度和高度（最小/最大属性）确定元素边框box。

padding(内边距) 和 border(边框) 也包含在 width 和 height 中

内容的宽度和高度减去各自双方该边框和填充的宽度从指定的"宽度"和"高度"属性计算

inherit 指定box-sizing属性的值，应该从父元素继承

outline-offset : 对轮廓进行偏移，并在超出边框边缘的位置绘制轮廓。轮廓不占用空间

outline-offset:15px;  这个轮廓在div 边框外 15 像素处

outline:2px solid red;

13.图片

圆角图片： img { border-radius: 8px; }

椭圆形图片:img { border-radius: 50%;}

响应式图片：自动适配各种尺寸的屏幕。

img { max-width: 100%;  
     height: auto; }

图片上添加文字：.topleft { position: absolute;

top: 8px;

left: 16px;

font-size: 60px; }

14.按钮

按钮颜色： .button1 {background-color: #4CAF50;} /\* Green \*/

按钮大小： .button3 {font-size: 16px;}

圆角按钮： .button1 {border-radius: 2px;}

边框颜色： .button1 { border: 2px solid #4CAF50; /\* Green \*/ }

鼠标悬停： .button:hover { background-color: #4CAF50; /\* Green \*/  
    color: white; }

按钮阴影：.button1{box-shadow:0 4px 8px 0 rgba(0,0,0,0.2);}

禁用按钮：.disabled { opacity: 0.6;  
    cursor: not-allowed; } 设置一个禁用的图片

按钮宽度： .button1 {width: 250px;}

按钮组： .button {  float: left; }

带边框的按钮组：.button { float: left;  border: 1px solid green }

15.分页

.active ：设置当期页样式

ul.pagination li a.active { background-color: #4CAF50;  
     color: white; }

  设置非当前页样式：ul.pagination li a:hover:not(.active) {background-color: #ddd;}

16.弹性窗口（ flex box）：当页面需要适应不同的屏幕大小以及设备类型时确保元素拥有恰当的行为的布局方式。

17.多媒体查询：针对不同媒体类型可以定制不同的样式规则。

语法：@media not|only mediatype and (expressions) { CSS 代码...; }

例：在屏幕可视窗口尺寸大于 480 像素的设备上修改背景颜色

@media screen and (min-width: 480px) {  
    body {  background-color: lightgreen; }  
 }