## **HTML 5提供的一些新的标签**

**<article>**标签定义外部的内容。

**<aside>**标签定义 article 以外的内容。aside 的内容应该与 article 的内容相关。

**<audio>** 标签定义声音，比如音乐或其他音频流。

**<canvas>** 标签定义图形，比如图表和其他图像。这个 HTML 元素是为了客户端矢量图形而设计的。它自己没有行为，但却把一个绘图 API 展现给客户端 JavaScript 以使脚本能够把想绘制的东西都绘制到一块画布上。

**<command>** 标签定义命令按钮，比如单选按钮、复选框或按钮。

**<datalist>** 标签定义可选数据的列表。与 input 元素配合使用，就可以制作出输入值的下拉列表。

**<details>** 标签定义元素的细节，用户可进行查看，或通过点击进行隐藏。与 <legend> 一起使用，来制作 detail 的标题。该标题对用户是可见的，当在其上点击时可打开或关闭 detail。

**<embed>** 标签定义嵌入的内容，比如插件。

**<figcaption>** 标签定义 figure 元素的标题。”figcaption” 元素应该被置于 “figure” 元素的第一个或最后一个子元素的位置。

**<figure>** 标签用于对元素进行组合。使用 <figcaption> 元素为元素组添加标题。

**<footer>** 标签定义 section 或 document 的页脚。典型地，它会包含创作者的姓名、文档的创作日期以及/或者联系信息。

**<header>** 标签定义 section 或 document 的页眉。

**<hgroup>** 标签用于对网页或区段（section）的标题进行组合。

**<keygen>** 标签定义生成密钥。

**<mark>**主要用来在视觉上向用户呈现那些需要突出的文字。<mark>标签的一个比较典型的应用就是在搜索结果中向用户高亮显示搜索关键词。

**<meter>** 标签定义度量衡。仅用于已知最大和最小值的度量。必须定义度量的范围，既可以在元素的文本中，也可以在 min/max 属性中定义。

**<nav>** 标签定义导航链接的部分。

**<output>** 标签定义不同类型的输出，比如脚本的输出。

**<progress>** 标签运行中的进程。可以使用 <progress> 标签来显示 JavaScript 中耗费时间的函数的进程。

**<rp>** 标签在 ruby 注释中使用，以定义不支持 ruby 元素的浏览器所显示的内容。

**<rt>** 标签定义字符（中文注音或字符）的解释或发音。

**<ruby>** 标签定义 ruby 注释（中文注音或字符）。

**<section>** 标签定义文档中的节（section、区段）。比如章节、页眉、页脚或文档中的其他部分。

**<source>** 标签为媒介元素（比如 <video> 和 <audio>）定义媒介资源。

**<summary>** 标签包含 details 元素的标题，”details” 元素用于描述有关文档或文档片段的详细信息。”summary” 元素应该是 “details” 元素的第一个子元素。

**<time>** 标签定义日期或时间，或者两者。

**<video>** 标签定义视频，比如电影片段或其他视频流。

## **Css3总结**

Css3一些效果不兼容pc端页面，必须写兼容。

**渐变：**

linear-gradient(top,blue,white) 起点设置为头部

linear-gradient(left,blue,white) 起点设置为左边

linear-gradient(lefttop,blue,white) 起点设置为左上边

## **边框圆角：**

border-radius:10px; //四角

border-radius:10%;

border-radius:10px 20px; //斜对角

border-radius:10px 20px 10px 20px; //顺四角

**盒子阴影：**

box-shadow: 5px -5px 5px #b6ebf7;

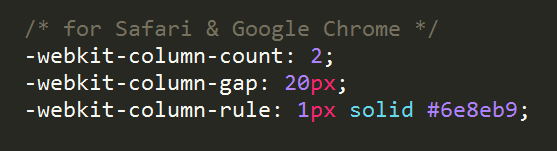
box-shadow:水平偏移的位置 垂直偏移的位置 阴影大小 颜色

**透明度（transparency）**

background-color: rgba(110, 142, 185, .5);

background-color: 前3个是颜色值，最后一个是透明度大小

**列表（Columns）布局**



-webkit-column-count: 2; 表示列的数目

-webkit-column-rule: 1px solid #6e8eb9; 表示中间分割线的样式

# 轮廓（outlines）



**outline**: 1px solid #699; 这个画框；

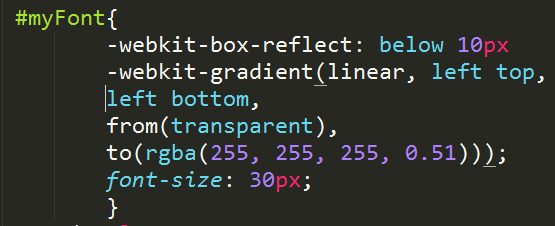
outline-offset 这个表示里面的偏移量，通俗点说 就是距离。

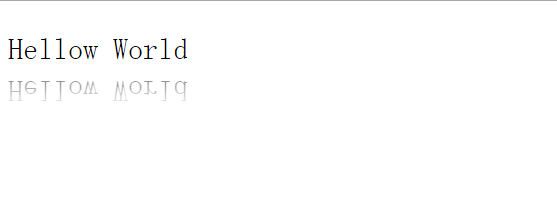
# 旋转（Rotate）

-webkit-transform: rotate(20deg);

rotate（角度）里面的数组就是角度的意思，deg是英文单词degree的缩写 度数的意思。

# 反射（reflect）注：反射，其实也就是倒影效果





**Box-sizing 属性**

box-sizing 设置为 "border-box" 注：把边框和内边距放入框中。

box-sizing 设置为“content-box” 注：默认值

**Css3弹性盒子**

1. **使用"display: flex"声明使用弹性盒布局。**

**2水平或垂直分布：**

box-orient定义分布的坐标轴：vertical和horizional

vertical 垂直

horizional 水平

**3反向分布**

box-direction可以设置盒子出现的顺序

定义“box-direction”的属性值为“reverse”，则反转盒子的排列顺序

**4具体分布**

属性“box-ordinal-group”定义盒子分布的顺序;

可以随意的控制其分布顺序。这些组以一个从“1”开始的数字定义，

盒模型将首先分布这些组，所有这些盒子将在每个组中。分布将从小到大排列。

**5.盒子尺寸**

默认情况下，盒子并不具有弹性，如果display:flex的属性值至少为1时，

则变得富有弹性。

**6.溢出管理**

因为是弹性盒子、非弹性盒子混排，有可能所有盒子的尺寸大于或小于父盒子的尺寸。这样就有可能空间太多或空间不足。

**7.空间太多如何处理**

可利用空间的分布取决于两个属性值：box-align 和 box-pack。

**属性“box-pack”管理水平方向上的空间分布，有以下四个可能属性：start、**

**end、 justify、 or center。**

start 所有盒子在父盒子的左侧，余下的空间在右侧；

end所有盒子在父盒子的右侧，余下的空间在左侧；

justify 余下的空间在盒子间平均分配；

center 可利用的空间在父盒子的两侧平均分配。

**属性“box- align”管理垂直方向上的空间分布，有以下五个可能属性之：**

start、 end,、center、 baseline和 stretch。

start 每个盒子沿父盒子的上边缘排列，余下的空间位于底部；

end 每个盒子沿父盒子的下边缘排列，余下的空间位于顶部；

center 可用空间平均分配，上面一半，下面一半；

baseline 所有盒子沿着它们的基线排列，余下的空间可前可后；

stretch 每个盒子的高度调整到适合父盒子的高度

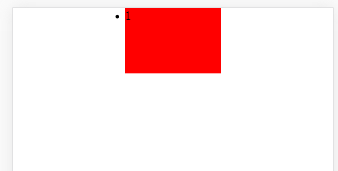
**8.垂直居中兼容写法**

--webkit-box-pack: center;

-moz-justify-content: center;

-webkit-justify-content: center;

justify-content: center;



http://www.ccwebsite.com/flex-layout-old-and-new-compatible/?utm\_source=tuicool&utm\_medium=referral 这里讲解了一些多栏多列布局，还有一些参数的意思。

**移动端总结**

1. **做页面之前要先在<head>里面添加屏幕配置**

**强制让文档的宽度与设备的宽度保持1:1，并且文档最大的宽度比例是1.0，且不允许用户点击屏幕放大浏览。**

<meta name="viewport" content="width=device-width, initial-scale=1.0,maximum-scale=1.0, user-scalable=no"/>

**WebApp全屏模式   隐藏地址栏。**

<meta name="apple-mobile-web-app-capable" content="yes" />

<meta name="apple-touch-fullscreen" content="yes">

1. **rem单位的换算：**

首先要书写核心js代码，控制住页面的初始大小：我是以750px（即iPhone6）的标准,设置font-size：100；

**(function(a, d) {**

**var b = a.documentElement,**

**e = "orientationchange" in window ? "orientationchange" : "resize",**

**c = function() {**

**var a = b.clientWidth;**

**a && (b.style.fontSize = Math.min(a, 640) / 320 \* 100 + "px")**

**};**

**console.log(b.style.fontSize);**

**a.addEventListener && (d.addEventListener(e, c, !1),**

**a.addEventListener("DOMContentLoaded", c, !1))**

**})(document, window);**