1. 强调标签：

strong：使用粗体标识；em：使用斜体表示；span标签是没有语义的，是为了设置单独样式。

1. 文本引用

<q>...</q>标签：短文本引用；<blockquote>...</blockquote>：长文本引用。

1. 换行：<br/>。在html中是忽略空格、回车的。空格：&nbsp；
2. 添加水平横线：<hr/>
3. 地址标签：<address>...</address>
4. 标志代码

一行代码：<code>...</code>；一段代码：<pre>...</pre>

1. 列表

无序列表：<ul>...</ul>；有序列表：<ol>...</ol>

1. 表格标签

① <table>...</table>：整个表格以<table>标记开始、</table>标记结束

② <tbody>...</tbody>：当表格内容非常多时，表格会下载一点显示一点，加上<tbody>标签后，表格会等内容全部下载完毕才会显示

③ <tr>...</tr>：表格的一行

④ <td>...</td>：表格的一个单元格，一行中包含几对<td>就有几列

⑤ <th>...</th>：表格的头部的一个单元格，表格表头

⑥ <caption>...</caption>：表格标题

1. 超链接

<a href=”#” tilte=” “>...</a>。 在新窗口打开加入属性 target=”\_blank”

1. 图片

<img src="图片地址" alt="下载失败时的替换文本" title = "提示文本">

1. 表单提交

<form method="post" action="save.php">...</form>

1. 文本区域

<textarea **rows="行数"** **cols="列数"**>文本</textarea>

1. 单选框、复选框

<input type="radio/checkbox" value="值" name="名称" checked="checked"/>

1. 使用下拉框

单选：<select><option value=”” selected=”selected>...</option></select>

多选：<select multiple=”multiple”><option ...>...</option></select>

1. label标签：当用户单击选中该label标签时，浏览器就会自动将焦点转到和标签相关的表单控件上，用for属性关联

# css

1. 外联式写法：

<link href="**base.css**" rel="stylesheet" type="text/css" />

1. 类选择器与ID选择器：
   1. ID选择器只能在文档中使用一次，类选择器可以使用多次
   2. 可以使用类选择器词列表方法为一个元素同时设置多个样式，而ID不行

class=“stress bigsize”

1. 子选择器：使用”>”表示，用于选择指定标签元素的第一代子元素，如： .food>li{...}
2. 包含选择器：使用空格，选择指定标签元素下的子元素，如： .first span{...}
3. 通用选择器： \*{...}
4. 伪类选择符：比较常用的就是 a:hover{...},鼠标滑过的状态
5. 分组选择器：为html多个标签设置同一个样式，可使用分组选择符(,) 如 h1,span{...}
6. css选择器的权值计算规则：标签的权值为1；类选择符的权值为10；ID选择符的权值最高位100；继承权值最低。
7. 为某些权值设置具有最高权值使用 !important 要写在分号前面
8. CSS格式化排版
   1. 设置字体： font-family:”Microsoft Yahei”;
   2. 字号颜色： font-size:12px;color:#666;
   3. 粗体： font-weight:bold;
   4. 斜体： font-style:italic;
   5. 下划线： text-decoration:underline;
   6. 删除线： text-decoration:line-through;
   7. 缩进： text-indent:2em;
   8. 行间距： line-height:1.5em;
   9. 中文字间距、字母间距：lettr-spacing:50px; 单词间距设置：word-spacing:50px;
   10. 对齐： text-align:center (left/right);
9. 元素分类
   1. 块状元素有：div、p、h1...h6、ol、ul、dl、table、address、blockquote、form
   2. 内联元素有：a、span、br、i、em、strong、label、q、var、cite、code
   3. 内联块状元素有：img、input
10. 块级元素特点：(块状元素内联 display:inline;)
    1. 每个块级元素都是从新的一行开始，并且其后的元素也另起一行
    2. 元素的高度、宽度、行高以及顶和底边距都可设置
    3. 元素宽度在不设置的情况下，是它本身父容器的100%
11. 内联元素特点：
    1. 和其他元素都在一行
    2. 元素的高度、宽度及顶部和底部边距不可设置
    3. 元素的宽度就是它包含的文字或图片的宽度，不可改变
12. 内联块状元素 (设置为内联块状元素：display:inline-block;)
    1. 和其他元素在一行
    2. 元素的高度、宽度、行高以及顶和底边距都可以设置
13. 盒子模型
    1. 盒子模型的边框就是围绕着内容及补白的线，这条线可是设置其粗细、样式和颜色。

[example] div{border:2px solid red;}

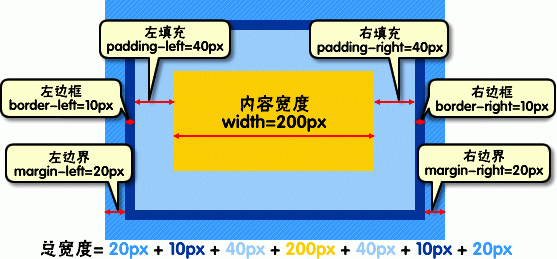
分开写就是: dashed(虚线) dotted(点线) solid(实线)

div{border-widtd:2px;border-style:solid;border-color:red;}

* 1. 盒子模型单独设置一个边框：

div{border-bottom|border-top|border-right|border-left:1px solid #ccc;}

* 1. 盒模型的宽度和高度



* 1. 填充

填充可以 分为上、右、下、左（顺时针）

[example] div{padding:20px 10px 15px 30px}

也可以分开写成：div{padding-top:20px;padding-right:10px;padding-bottom:15px;padding-lefg:30px;}

如果上下左右都一样可以写成: div{padding:10px;}

1. 流动模型（默认的网页布局模式）
   1. 块状元素都会在所处的包含元素内自上而下按顺序垂直延伸分布，默认状态下，块状元素的宽度都为100%。即块状元素都会以行的形式占据位置。
   2. 内联元素都会在所处的包含元素内从左到右水平分布显示
2. 浮动模型

让两个块状元素并排显示，可以浮动

1. 层模型
   1. 绝对定位：[position:absolute],使用top等属性相对于其最接近的一个具有定位属性的父包含块进行绝对定位。
   2. 相对定位：[position:relative],相对定位完成的过程首先按static方式生成一个元素，然后相对于以前的位置移动，移动的方向和幅度由top等属性确定，偏移前的位置保留不动。
   3. 固定定位：[position:fixed],固定定位的元素会是种位于浏览器窗口内视图的某个位置。
   4. Relative与Absolute组合使用：参照定位的元素必须是相对定位元素的前辈元素；参照定位元素必须加入position:relative；定位元素加入position:absolute，便可以使用top等来进行偏移定位
2. 代码简写
   1. 盒模型代码简写：top、right、bottom、left值相同，可缩写为：margin:10px;
   2. 如果top和bottom值相同，left和right值相同，可缩写为：margin:10px 20px;
   3. 如果left和right值相同，可缩写为: margin:10px 20px 30px;
   4. 颜色值缩写：设置的颜色是16进制的色彩值时，如果每两位的值相同，可以缩写一半。
   5. 字体缩写：使用字体简写至少要指定font-size和font-family属性；在font-size和line-height中间要加入”/” [ font:12px/1.5em “宋体”，sans-serif;],缩写顺序：font-style font-variant font-weight font-size font-family
3. 长度： 1em = font-size
4. 水平居中设置
   1. 定宽块状元素：设置margin:auto; width也要设置
   2. 不定宽块状元素：三种方法：加入table标签；设置display:inline；设置position:relative和left:50%
5. 垂直居中
   1. **父元素高度确定的单行文本**的竖直居中的方法是通过设置父元素的 height 和 line-height 高度一致来实现的
   2. 父元素高度确定的多行文本、图片、块状元素：使用插入table（包括tbody、tr、td）标签，同时设置vertical-align:middle（vertical-align只有在父元素是td或th才有效）； display:table-cell /\*IE8以上及Chrome、Firefox\*/、vertical-align：middle /\* IE8以上及Chrome、Firefox \*/

CSS动画

1. transition: transition:opacity 1s ease-in-out 0.5s/\*延时\*/, transform 1s ease-in-out
2. animation: animation
3. transform: 变换，改变元素的颜色位置大小
   1. translate/translateX/translateY(x [,y]): 沿着X和Y轴移动元素:[transform:translate(-10px, -10px);transform:translateX(10px);]
   2. rotate(angle): 旋转元素对象，值为一个度数值:[transform: rotate(45deg);]
   3. scale/scaleX/scaleY(angle): 缩放，改变元素的宽度和高度，取值包括正数、负数和小数:[ transform: scale(0.8, 0.8); transform: scaleX(0.8);]
   4. skew/skewX/skewY(angle [, angle])：倾斜元素对象，取值为一个度数值:[ transform: skew(15deg, 15deg); transform: skewX(15deg);]
   5. transform-origin(x-axis[,y-axis,z-axis]): x-axis (定义视图被置于x轴的何处，可能值是left\center\right\bottom\length\%); y-axis(定义视图被置于Y轴的何处); z-axis; [transform-origin:left top;]
4. transition 动画
   1. transition: 简写属性，用于在一个属性中设置四个过渡属性: [transition: all 4s linear;]
   2. transition-property none|all|property: 规定应用过渡的CSS属性的名称
   3. transition-duration time: 定义过渡花费的时间，默认是0
   4. transition-timing-function: 定义过渡效果的时间曲线，默认是 "ease"

* linear: 规定以相同的速度开始至结束的过渡效果
* ease : 规定以慢速开始，然后变快，然后慢速结束
* ease-in : 规定以慢速开始的过渡效果
* ease-out: 规定以慢速结束的过渡效果
* cubic-bezier(n,n,n,n): 在cubic-bezier中定义自己的值，可以是0到1之间
* transition-delay time: 规定过渡效果何时开始

1. @keyframes 规则是创建动画。@keyframes规则内指定一个CSS样式和动画将逐步从目前的样式更改为新的样式。用百分比来规定变化发生的时间，或用关键字“from” 和 “to” 等同于0%和100%；始终定义0%和100%。

@-webkit-keyframes running {  
 0% {  
 *background-position*: 0, 0;  
 }  
 14.3% {  
 *background-position*: -180**px**, 0;  
 }  
 28.6% {  
 *background-position*: -360**px**, 0;  
 }  
 42.9% {  
 *background-position*: -540**px**, 0;  
 }  
 57.2% {  
 *background-position*: -720**px**, 0;  
 }  
 71.5% {  
 *background-position*: -900**px**, 0;  
 }  
 85.8% {  
 *background-position*: -1080**px**, 0;  
 }  
 100% {  
 *background-position*: 0, 0;  
 }  
}

1. animations 动画属性

*-webkit-animation-name*: **running**;  
*-webkit-animation-duration*: 950**ms**;  
*-webkit-animation-iteration-count*: **infinite**;  
*-webkit-animation-timing-function*: **step-end**;

* 1. animation-name : keyframename|none: 规定 @keyframes 动画的名称
  2. animation-duration : time : 规定动画完成一个周期所花费的秒或毫秒，默认是0
  3. animation-timing-function : ease|linear|step-start|step-end : 规定动画的速度曲线，默认是 "ease"
  4. animation-delay :time : 规定动画何时开始，默认是0
  5. animation-iteration-count：value|infinite: 规定动画被播放的次数，默认是1
  6. animation-direction : 是否循环交替反向播放动画
     + normal: 默认值，动画按正常播放
     + reverse: 动画反向播放
     + alternate: 动画在奇数次正向播放，偶数次反向播放
     + alternate-reverse: 与alternate相反
     + initial: 设置该属性为他的默认值
  7. animation-play-state ： 规定动画是否正在运行或暂停，默认是”running”
  8. animation： 所有动画属性的简写属性，除了play-state

1. CSS3 定义了两种类型的渐变： 线性渐变、径向渐变

*background*: linear-gradient(**left**, **red**, **green**);

* 1. linear-gradient([<point> || <angle>]?<stop>,<stop>[,<stop>]\*)
     + 第一个参数表示线性渐变的方向（角度）top是从上到下，left是从左到右， left top是从左上角到右下角
     + 第二个和第三个参数分别是起点颜色和终点颜色，可以在它们之间插入更多的参数，表示多种颜色的渐变
  2. CSS3 径向渐变

*background*: radial-gradient(60% 55%, **closest-side**, **blue**, **green**, **yellow**, **black**);

radial-gradient([bg-position||<angle>]?[<shape>||<size>,]?<color-stop>,<color-stop>[,<color-stop>]\*)