1:

字体：body{font-family:"Microsoft Yahei";}，位于<style>内。

字号与颜色：body{font-size:12px;color:#666}；当你设置的颜色是16进制的色彩值时，如果每两位的值相同，可以缩写一半。#666原为#666666。有字体颜色（color）、背景颜色（background-color）、边框颜色（border）；

粗体：p span{font-weight:bold;}

斜体：p a{font-style:italic;}

下划线：p a{text-decoration:underline;}

删除线：.oldPrice{text-decoration:line-through;}

段落缩进：p{text-indent:2em;}，2em的意思就是文字的2倍大小。

行间距：p{line-height:2em;}

文字或字母间隔：h1{

letter-spacing:50px;

}

...

<h1>了不起的盖茨比</h1>

单词间隔：h1{

word-spacing:50px;

}

...

<h1>welcome to imooc!</h1>

居中（左，右）显示：h1{

text-align:center;或left，right

}

<h1>了不起的盖茨比</h1>

2：html中的标签元素大体被分为三种不同的类型：块状元素、内联元素(又叫行内元素)和内联块状元素。

常用的块状元素有：

<div>、<p>、<h1>...<h6>、<ol>、<ul>、<dl>、<table>、<address>、<blockquote> 、<form>...设置display:block就是将元素显示为块级元素。 a{display:block;}将内联元素a转换为块状元素，从而使a元素具有块状元素特点。块级元素特点：

1、每个块级元素都从新的一行开始，并且其后的元素也另起一行。（真霸道，一个块级元素独占一行）

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%（和父元素的宽度一致），除非设定一个宽度。

常用的内联元素有：

<a>、<span>、<br>、<i>、<em>、<strong>、<label>、<q>、<var>、<cite>、<code>

当然块状元素也可以通过代码display:inline将元素设置为内联元素，

div{

display:inline;

}

......

<div>我要变成内联元素</div> 就是将块状元素div转换为内联元素，从而使 div 元素具有内联元素特点。内联元素特点：

1、和其他元素都在一行上；

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。

常用的内联块状元素有：

<img>、<input>内联块状元素（inline-block）就是同时具备内联元素、块状元素的特点，代码display:inline-block就是将元素设置为内联块状元素。inline-block 元素特点：

1、和其他元素都在一行上；

2、元素的高度、宽度、行高以及顶和底边距都可设置。

3：

盒模型边框：盒子模型的边框就是围绕着内容及补白的线，这条线你可以设置它的粗细、样式和颜色(边框三个属性)。 div 来设置边框粗细为 2px、样式为实心的、颜色为红色的边框：

div{

border:2px solid red;

}上面是 border 代码的缩写形式，可以分开写：

div{

border-width:2px;

border-style:solid;

border-color:red;

}

1、border-style（边框样式）常见样式有：

dashed（虚线）| dotted（点线）| solid（实线）。

2、border-color（边框颜色）中的颜色可设置为十六进制颜色，如:

border-color:#888;//前面的井号不要忘掉。

只为一个方向的边框设置样式：div{border-bottom:1px solid red;}

css内定义的宽（width）和高（height），指的是填充以里的内容范围。

因此一个元素实际宽度（盒子的宽度）=左边界+左边框+左填充+内容宽度+右填充+右边框+右边界。元素的高度也是同理。

元素内容与边框之间是可以设置距离的，称之为“填充”。填充也可分为上、右、下、左(顺时针)。如div{padding:20px 10px 15px 30px;}依次为上，右，下，左。如果上、右、下、左的填充都为10px;可以这么写div{padding:10px;}，如果上下填充一样为10px，左右一样为20px，可以这么写：div{padding:10px 20px;}。

元素与其它元素之间的距离可以使用边界（margin）来设置。边界也是可分为上、右、下、左。如代码：div{margin:20px 10px 15px 30px;}依次为上，右，下，左；如果上右下左的边界都为10px;可以这么写：div{ margin:10px;}；如果上下边界一样为10px，左右一样为20px，可以这么写：div{ margin:10px 20px;}；如果left和right的值相同为20px，可以这么写：margin:10px 20px 30px;。

padding和margin的区别，padding在边框里，margin在边框外。

4：

布局模型：需位于<style>内。

如果说css布局模型是本，那么 CSS 布局模板就是末了，是外在的表现形式。

CSS包含3种基本的布局模型，用英文概括为：Flow、Layer 和 Float。  
在网页中，元素有三种布局模型：  
1、流动模型（Flow）：是默认的网页布局模式。也就是说网页在默认状态下的 HTML 网页元素都是根据流动模型来分布网页内容的。流动布局模型具有2个比较典型的特征：

第一点，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，因为在默认状态下，块状元素的宽度都为100%。实际上，块状元素都会以行的形式占据位置。

第二点，在流动模型下，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。（内联元素可不像块状元素这么霸道独占一行）。

2、浮动模型 (Float)：任何元素在默认情况下是不能浮动的，但可以用 CSS 定义为浮动，如 div、p、table、img 等元素都可以被定义为浮动。如下代码可以实现两个 div 元素一行显示。

div{

width:200px;

height:200px;

border:2px red solid;

float:left;

}

<div id="div1"></div>

<div id="div2"></div> 两个元素一左一右可以实现一行显示

div{

width:200px;

height:200px;

border:2px red solid;

}

#div1{float:left;}

#div2{float:right;}

1. 层模型（Layer）：层模型有三种形式：绝对定位(position: absolute)；相对定位(position: relative)；固定定位(position: fixed)；

想为元素设置层模型中的绝对定位，需要设置position:absolute(表示绝对定位)，这条语句的作用将元素从文档流中拖出来，然后使用left、right、top、bottom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。如果不存在这样的包含块，则相对于body元素，即相对于浏览器窗口；可以随窗口上下滑动而相对于视图变化。div元素相对于浏览器窗口向右移动100px，向下移动50px。

div{

width:200px;

height:200px;

border:2px red solid;

position:absolute;

left:100px;

top:50px;

}

<div id="div1"></div>

想为元素设置层模型中的相对定位，需要设置position:relative（表示相对定位），它通过left、right、top、bottom属性确定元素在正常文档流中的偏移位置。相对定位完成的过程是首先按static(float)方式生成一个元素(并且元素像层一样浮动了起来)，然后相对于以前的位置移动，移动的方向和幅度由left、right、top、bottom属性确定，偏移前的位置保留不动。相对于以前位置向下移动50px，向右移动100px;

#div1{

width:200px;

height:200px;

border:2px red solid;

position:relative;

left:100px;

top:50px;

}

<div id="div1"></div>

body>

<div id="div1"></div><span>偏移前的位置还保留不动，覆盖不了前面的div没有偏移前的位置</span>

</body>

虽然div元素相对于以前的位置产生了偏移，但是div元素以前的位置还是保留着，所以后面的span元素是显示在了div元素以前位置的后面。

fixed：表示固定定位，与absolute定位类型类似，但它的相对移动的坐标是视图（屏幕内的网页窗口）本身。由于视图本身是固定的，它不会随浏览器窗口的滚动条滚动而变化，除非你在屏幕中移动浏览器窗口的屏幕位置，或改变浏览器窗口的显示大小，因此固定定位的元素会始终位于浏览器窗口内视图的某个位置，不会受文档流动影响，这与background-attachment:fixed;属性功能相同。

以相对于其它元素进行定位使用position:relative；参照定位的元素必须是相对定位元素的前辈元素：参照定位的元素必须加入position:relative;定位元素加入position:absolute，便可以使用top、bottom、left、right来进行偏移定位了。

5：长度单位总结一下，目前比较常用到px（像素）、em、% 百分比，要注意其实这三种单位都是相对单位。像素指的是显示器上的小点（CSS规范中假设“90像素=1英寸”）。实际情况是浏览器会使用显示器的实际像素值有关，在目前大多数的设计者都倾向于使用像素（px）作为单位。

em就是本元素给定字体的 font-size 值，如果元素的 font-size 为 14px ，那么 1em = 14px；如果 font-size 为 18px，那么 1em = 18px。如下代码：

p{font-size:12px;text-indent:2em;}就是可以实现段落首行缩进 24px（也就是两个字体大小的距离）。但当给 font-size 设置单位为 em 时，此时计算的标准以 p 的父元素的 font-size 为基础。

html:

<p>以这个<span>例子</span>为例。</p>

css:

p{font-size:14px}

span{font-size:0.8em;}

结果 span 中的字体“例子”字体大小就为 11.2px（14 \* 0.8 = 11.2px）。

百分比：p{font-size:12px;line-height:130%}设置行高（行间距）为字体的130%（12 \* 1.3 = 15.6px）。

6：

如果被设置元素为文本、图片等行内元素时，水平居中是通过给父元素设置text-align:center 来实现的。

当被设置元素为 [块状元素](http://www.imooc.com/code/2048" \o "忘了的小伙伴，让我们点击链接复习一下吧! ) 时用 text-align：center 就不起作用了，这时也分两种情况：定宽块状元素和不定宽块状元素。

满足定宽和块状两个条件的元素是可以通过设置“左右margin”值为“auto”来实现居中的。

<style>

div{

border:1px solid red;/\*为了显示居中效果明显为 div 设置了边框\*/

width:200px;/\*定宽，为所要居中的内容定宽\*/

margin:20px auto;/\* margin-left 与 margin-right 设置为 auto \*/

}

</style>

(不定宽块状元素：块状元素的宽度width不固定。)

不定宽度的块状元素有三种方法居中（这三种方法目前使用的都很多）：

1：加入 [table](http://www.imooc.com/code/292" \o "忘了的小伙伴，让我们点击链接复习一下吧! ) 标签。利用table标签的长度自适应性---即不定义其长度也不默认父元素body的长度（table其长度根据其内文本长度决定），因此可以看做一个定宽度块元素，然后再利用定宽度块状居中的margin的方法，使其水平居中。

第一步：为需要设置的居中的元素外面加入一个 table 标签，第二步：为这个 table 设置“左右 margin 居中”。

<style>table{

border:1px solid;

margin:0 auto;

}

</style>

2：设置 [display: inline](http://www.imooc.com/code/2049" \o "忘了的小伙伴，让我们点击链接复习一下吧! ) 方法：与第一种类似，显示类型设为 行内元素，进行不定宽元素的属性设置。改变块级元素的 display 为 inline 类型（设置为 [行内元素](http://www.imooc.com/code/2049" \o "忘了的小伙伴，让我们点击链接复习一下吧! ) 显示），然后使用 text-align:center 来实现居中效果。css代码：

<style>

.container{

text-align:center;

}/\* margin:0;padding:0（消除文本与div边框之间的间隙）\*/

.container ul{

list-style:none;

margin:0;

padding:0;

display:inline;

}/\* margin-right:8px（设置li文本之间的间隔）\*/

.container li{

margin-right:8px;

display:inline;

}

</style> html代码：

<body>

<div class="container">

<ul>

<li><a href="#">1</a></li>

<li><a href="#">2</a></li>

<li><a href="#">3</a></li>

</ul>

</div>

</body>

3：通过给父元素设置[float](http://www.imooc.com/code/2071" \o "忘了的小伙伴，让我们点击链接复习一下吧! )，然后给父元素设置 [position:relative](http://www.imooc.com/code/2074" \o "忘了的小伙伴，让我们点击链接复习一下吧! ) 和 left:50%，子元素设置 position:relative 和 left: -50% 来实现水平居中。我们可以这样理解：假想ul层的父层（即下面例子中的div层）中间有条平分线将ul层的父层（div层）平均分为两份，ul层的css代码是将ul层的最左端与ul层的父层（div层）的平分线对齐；而li层的css代码则是将li层的平分线与ul层的最左端（也是div层的平分线）对齐，从而实现li层的居中。Css代码：

<style>

.container{

float:left;

position:relative;

left:50%

}

.container ul{

list-style:none;

margin:0;

padding:0;

position:relative;

left:-50%;

}

.container li{float:left;display:inline;margin-right:8px;}

</style> html代码：

<body>

<div class="container">

<ul>

<li><a href="#">1</a></li>

<li><a href="#">2</a></li>

<li><a href="#">3</a></li>

</ul>

</div>

</body>

7：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](http://www.imooc.com/code/2083" \o "忘了的小伙伴，让我们点击链接复习一下吧!)高度一致来实现的。(height: 该元素的高度，line-height: 顾名思义，行高（行间距），指在文本中，行与行之间的 基线间的距离 )。

line-height 与 font-size 的计算值之差，在 CSS 中成为“行间距”。分为两半，分别加到一个文本行内容的顶部和底部。

这种文字行高与块高一致带来了一个弊端：当文字内容的长度大于块的宽时，就有内容脱离了块。

<style>

.container{

height:100px;

line-height:100px;

background:#999;

}

</style>

2：父元素高度确定的多行文本、图片等的竖直居中的方法：使用插入 [table](http://www.imooc.com/code/292" \o "忘了的小伙伴，让我们点击链接复习一下吧! )  (包括tbody、tr、td)标签，同时设置 vertical-align：middle。

css 中有一个用于竖直居中的属性 vertical-align，在父元素设置此样式时，会对inline-block类型的子元素都有用。

css代码：

table td{height:500px;background:#ccc}

因为 td 标签默认情况下就默认设置了 vertical-align 为 middle，所以我们不需要显式地设置了。

html代码：

<body><table><tbody><tr><td class="wrap">

<div>

<p>看我是否可以居中。</p>

</div></td></tr></tbody></table>

</body>

8：

为元素设置1. [position : absolute](http://www.imooc.com/code/2073" \o "忘了的小伙伴，让我们点击链接复习一下吧! ) 2. float : left 或 [float:right](http://www.imooc.com/code/2071" \o "忘了的小伙伴，让我们点击链接复习一下吧! ) 语句。

元素的display显示类型就会自动变为以 display:inline-block（[块状元素](http://www.imooc.com/code/2048" \o "忘了的小伙伴，让我们点击链接复习一下吧! )）的方式显示，当然就可以设置元素的 width 和 height 了，且默认宽度不占满父元素。 a 标签是 [行内元素](http://www.imooc.com/code/2049" \o "忘了的小伙伴，让我们点击链接复习一下吧! ) ，所以设置它的 width 是 没有效果的，但是设置为 position:absolute 以后，就可以了。

css代码

<style>

.container a{

position:absolute;

width:200px;

background:#ccc;

}

</style> html代码：

<div class="container">

<a href="#" title="">进入课程请单击这里</a>

</div>