## 关于命名

ID和类名不要以数字开头，数字开头的在 Mozilla/Firefox 浏览器中不起作用。

## 关于格式化

不要在属性值与单位之间留有空格。假如你使用 "margin-left: 20 px" 而不是 "margin-left: 20px" ，它仅在 IE 6 中有效，但是在 Mozilla/Firefox 或 Netscape 中却无法正常工作。

## 样式优先级关系(由低到高)

浏览器缺省设置

外部样式表

内部样式表（位于 <head> 标签内部）

内联样式（在 HTML 元素内部）

## 颜色的定义格式

十六进制 - 如："#ff0000"

RGB - 如："rgb(255,0,0)"

颜色名称 - 如："red"

## CSS背景

五大属性:

background-color 颜色

background-image 图像，需要路径

background-repeat 重复

background-attachment 粘附方式

background-position 位置

标准写法:

body

{

background-image:url('img.png');

background-repeat:no-repeat;

background-position:right top;

}

简略写法:(属性的顺序依照前面五大属性的顺序书写)

body {background:#ffffff url('img.png') no-repeat right top;}

背景图片不随着页面滚动:

background-attachment:fixed;

渐变的背景颜色：

linear-gradient(to bottom,#009dd7 0,#0079b1 50%,#00538b 100%)

## CSS文字

常用属性：

color 颜色

text-align 对齐方式

text-decoration 修饰，主要用于去掉下划线

text-indent 第一行缩进

text-transform 可以指定文字大小写转换

direction 文字方向

letter-spacing 文字间距

line-height 行高

vertical-align 垂直对齐

text-shadow 设置文字阴影

标准写法：

p

{

color:#778899;

text-align:center;

text-decoration:underline;

text-transform:capitalize;

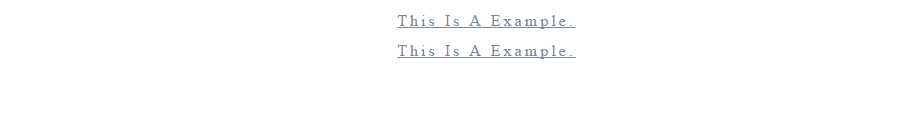
text-indent:50px;

letter-spacing:3px;

line-height:90%;

}

效果：



每一行被展开为宽度相等，左，右外边距是对齐（如杂志和报纸）。

text-align:justify;

删除链接的下划线

text-decoration:none;

## CSS字体

五大属性：

font-style 字体样式

font-variant 属性设置小型大写字母的字体显示文本，这意味着所有的小写字母均会被转换为大写，但是所有使用小型大写字体的字母与其余文本相比，其字体尺寸更小

font-weight 字体粗细

font-size/line-height 字体大小

font-family 字体系列

标准写法：

p

{

font-family: Georgia, serif;

font-style:italic;

font-size:30px;

}

简略写法：(属性的顺序依照前面五大属性的顺序书写)

p{font:italic 30px Georgia, serif;}

效果：



设置字体系列需注意：

* 多个字体系列是用一个逗号分隔指明。
* 如果字体系列的名称超过一个字，它必须用引号，如Font Family："宋体"。
* 防止一些兼容性问题，可以参考：Web安全字体组合。

http://www.runoob.com/cssref/css-websafe-fonts.html

三种方法设置字体大小：

设置文字的大小与像素

p {font-size:16px;}

用em来设置字体大小，1em的默认大小是16px。公式 em：px/16=em

p {font-size:1em;}

一个比较好看的现代英文字体：

font:26px 'Segoe UI Light',Helvetica,Arial,'Sans-Serif';

## 链接样式

链接的样式，可以用任何CSS属性（如颜色，字体，背景等）。

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

设置需按照以上顺序进行。

## 列表样式

三大属性：

list-style-type 设置列表项标志的类型。

list-style-position 设置列表中列表项标志的位置。

list-style-image 将图象设置为列表项标志。

标准写法：

Ul

{

list-style-type:circle;

list-style-position:inside;

}

简略写法：ul{list-style: circle inside;}

效果：



## 表格样式

border 边框

border-collapse 边框折叠

width/height 表格的宽度和高度

text-align 表格文字对齐

vertical-align 表格文字垂直对齐

padding 表格内容内边距

background-color 表格背景色

color 文本色

标准写法:

#customers{

font-family:"Trebuchet MS", Arial, Helvetica, sans-serif;

width:100%;

border-collapse:collapse;

}

#customers td, #customers th {

font-size:1em;

border:1px solid #000021;

padding:3px 7px 2px 7px;

}

#customers th {

font-size:1.1em;

text-align:left;

padding-top:5px;

padding-bottom:4px;

background-color:#8FBC8F;

color:#ffffff;

}

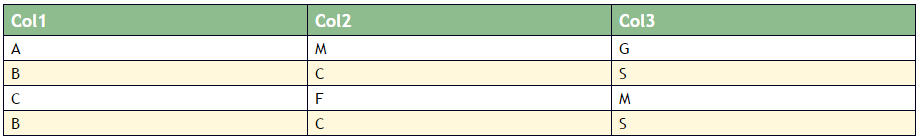
#customers tr.alt td {

color:#000000;

background-color:#FFF8DC;

}

效果:



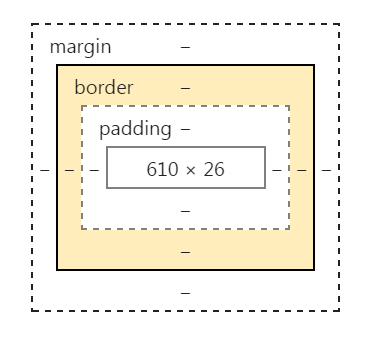
## 盒子模型

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。



元素的高度和宽度计算公式：

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

块和内联元素(重要)

块（block）元素是一个元素，占用了全部宽度，在前后都是换行符。

块元素的例子：

<h1>

<p>

<div>

内联（inline）元素只需要必要的宽度，不强制换行。

内联元素的例子：

<span>

<a>

可以更改内联元素和块元素，反之亦然，可以使页面看起来是以一种特定的方式组合，并仍然遵循web标准。

li {display:inline;}

## 边框

border-(top, right, bottom, left)-(color, style, width) 逐个边框的上右下左分别设置

简写:

border: (width style color) 直接全局设置边框上右下左的宽度,样式,颜色

style值可选项:

dotted: dotted:定义一个点线框

dashed: 定义一个虚线框

solid: 定义实线边界

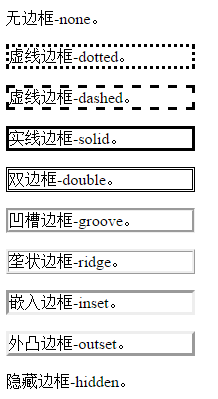
double: 定义两个边界。 两个边界的宽度和border-width的值相同

groove: 定义3D沟槽边界。效果取决于边界的颜色值

ridge: 定义3D脊边界。效果取决于边界的颜色值

inset:定义一个3D的嵌入边框。效果取决于边界的颜色值

outset: 定义一个3D突出边框。 效果取决于边界的颜色值



width值可选项：

直接设置数值（px, em）或者使用关键字“thick”，“medium”，“thin”，默认为medium

注意： border-color单独使用是不起作用的，必须得先使用border-style来设置边框样式。

outline

轮廓（outline）是绘制于元素周围的一条线，位于边框边缘的外围，可起到突出元素的作用。它跟border是一样是元素的边框，但是它又有别于border：outline 它的大小不会计算到元素上，不会影响布局，而border是会的。

它的语法与border一致，将上面的border改为outline即可。

画圆形边框：

<http://www.jb51.net/css/42638.html>

<http://www.kuqin.com/shuoit/20141014/342620.html>

## 外边距和内边距

内外边距（margin, padding）的区别参考盒子模型

margin可选的值：auto,数值（px）,%

padding可选的值：数值（px）,%

(margin, padding)-(top, bottom, right, left) ：+ 可选值

简写：

(margin, padding)：(top, bottom, right, left)可选值

注意：当元素的 Padding（填充）（内边距）被清除时，所"释放"的区域将会受到元素背景颜色的填充。

## 元素的尺寸

|  |  |  |
| --- | --- | --- |
| 属性 | 描述 | 可选值 |
| height | 设置元素的高度。 | auto，数值（px），%，inherit |
| width | 设置元素的宽度。 | auto，数值（px），%，inherit |
| line-height | 设置行高。 | normal，数值（px），%，inherit，  Number(此数字会与当前的字体尺寸相乘来设置行间距)。 |
| max-height | 设置元素的最大高度。 | none，数值（px），%，inherit |
| max-width | 设置元素的最大宽度。 | none，数值（px），%，inherit |
| min-height | 设置元素的最小高度。 | 数值（px），%，inherit |
| min-width | 设置元素的最小宽度。 | 数值（px），%，inherit |

显示（display）与可见性（visibility）

visibility: visible /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。也就是说，该元素虽然被隐藏了，但仍然会影响布局。

display: none可以隐藏某个元素，且隐藏的元素不会占用任何空间。也就是说，该元素不但被隐藏了，而且该元素原本占用的空间也会从页面布局中消失。

## 定位

position可选值：

static：系统默认定位方式

fixed：元素的位置相对于浏览器界面窗口固定，主要用于浮动的导航栏

absolute: 和fixed定位类似，但是它是相对于最近的已定位的父元素，如果没有则相对于<html>

relative: 它的定位是相对于它本身正常的位置。有点类似于margin

z-index: 设置元素的堆叠顺序，可选值为：Number（数字越大，越靠前），auto，inherit

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。

clip: 用于裁剪元素

标准写法：

clip: rect(0px,60px,200px,0px);

使用滚动条来显示元素内溢出的内容

overflow 设置当元素的内容溢出其区域时发生的事情。

overflow-y 指定如何处理顶部/底部边缘的内容溢出元素的内容区域

overflow-x 指定如何处理右边/左边边缘的内容溢出元素的内容区域

可选的值：

visible 默认值。内容不会被修剪，会呈现在元素框之外。

hidden 内容会被修剪，并且其余内容是不可见的。

scroll 内容会被修剪，但是浏览器会显示滚动条以便查看其余的内容。

auto 如果内容被修剪，则浏览器会显示滚动条以便查看其余的内容。

inherit 规定应该从父元素继承 overflow 属性的值。

## 浮动与对齐

float: 一个浮动元素会尽量向左或向右移动，直到它的外边缘碰到包含框或另一个浮动框的边框为止。浮动元素之后的元素将围绕它，浮动元素之前的元素将不会受到影响。

使用clear可以清除浮动。

float可选值： left，right，both，none，inherit

clear可选值： left，right，none，inherit

margin实现居中

标准写法：（left或者right效果一致，如果宽度是100％，对齐是没有效果的）

.center

{

margin-left: auto;

width:70%;

}

position实现左右对齐

标准写法：

.right

{

position: absolute;

right:0px;

width:300px;

}

float也可以实现左右对齐。

## CSS的一些常用书写语法

后代选择器：匹配所有值得元素的后代元素，即所有div里的p

div p

{

background-color:yellow;

}

子元素选择器：与后代选择器相比，子元素选择器只能选择作为某元素子元素的元素

div>p

{

background-color:yellow;

}

伪类：

伪类的语法：

selector: pseudo-class {property : value;}

CSS类也可以使用伪类：

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

标准写法：

a:hover {color:#FF00FF;}

1. red:visited {color:#FF0000;}

一些常用的伪类：

:link 链接样式

:visited 链接访问过后的样式（只要访问历史记录存在）

:hover 鼠标悬停样式

:active 鼠标按下样式

Selector:focus 设置对象在成为输入焦点（该对象的onfocus事件发生）时的样式表属性。

一些常用的伪元素：

Selector:first-child 匹配作为其父元素的第一个子元素的任何元素。

Selector:first-letter 设置对象内第一个字符的样式

Selector:first-line 设置对象内第一行的样式

Selector:before,Selector:after 用来和 content 属性一起使用，设置在对象前(依据对象树的逻辑结构)发生的内容,也就是在对象前、后添加指定的文本。

@media 规则：@media 规则允许在相同样式表为不同媒体设置不同的样式。

标准写法：

@media only screen and (max-width: 500px)

{

body {

background-color: lightblue;

}

}

属性选择器

标准写法：

把包含标题（title）的所有元素变为蓝色

[title]

{

color:blue;

}

属性和值选择器

标准写法：

改变了标题title='runoob'元素的边框样式

[title=runoob]

{

border:5px solid green;

}

属性和值的选择器 - 多值

标准写法：

包含指定值的title属性的元素样式的例子，使用（~）分隔属性和值

[title~=hello] {

color:blue;

}

表单样式

属性选择器样式无需使用class或id的形式:

标准写法：

input[type="text"]

{

width:150px;

display:block;

margin-bottom:10px;

background-color:yellow;

}

分组与嵌套:

为了尽量减少代码，你可以使用分组选择器。每个选择器用逗号分隔.

h1

{

color:green;

}

h2

{

color:green;

}

p

{

color:green;

}

等效于

h1,h2,p

{

color:green;

}

为所有class="marked"元素内的p元素指定一个样式

.marked p

{

color:white;

}

## 其他

滚动条：适用于webkit浏览器 Google Chrome 和 Opera

<http://www.07net01.com/2016/10/1701847.html>

一些有趣的网站：

http://codepen.io/