

文字、段落、背景的常用样

一、文字样式

(一)、字体颜色: **color**

- 字体颜色
- #FF0000
- rgb(255,0,0)
- red;

(二)、文字大小: **font-size**

- 1em = 16px; 0.75em=12px;
- xx-small 9px
- x-small 11px
- small 13px
- medium 16px
- large 19px
- x-large 23px
- xx-large 27px

【附加:】字体单位: px、em、rem。

(三)、字体库系列: **font-family**: Arial, "Times New Roman", "宋体","黑体","微软雅黑"; 当字体是中文字体时需要加双引号, 当英文字体中有空格时需要加双引号。一般常设置三个值, 一个一个匹配, 当计算机操作系统中没有找到该字体库, 则使用默认字体。

CSS 定义了 5 种通用字体系列:

- Serif 字体
- Sans-serif 字体
- Monospace 字体
- Cursive 字体
- Fantasy 字体

(四)、字体粗细: **font-weight**:

- 数值范围: 100-900
- 100 对应最细的字体变形, 900 对应最粗的字体变形
- 数字 400 等价于 normal
- 而 700 等价于 bold
- bolder/lighter

(五)、字体风格: **font-style**

- 默认: normal
- italic: 斜体
- oblique: 倾斜

font-style: normal/italic

斜体 (italic) 是一种简单的字体风格, 对每个字母的结构有一些小改动, 来反映变化的外观。与此不同, 倾斜 (oblique) 文本则是正常竖直文本的一个倾斜版本。通常情况下, italic 和 oblique 文本在 web 浏览器中看上去几乎完全一样。

(六)、font字体简写或复合写法:

- font: font-size/line-height font-family (两组数值: 字体大小/文本行高 字体库)
- font: style weight size/line-height family (四组数值: 字体风格 字体粗细 字体大小/文本行高 字体库)

例如:

```
p {  
    font: italic bold 12px/20px arial,sans-serif;  
}
```

直接描述字体的所有属性。

(七)、自定义字体库:

- 1、将字体库文件引入工程;
- 2、自定义字体库名称:

```
@font-face {  
    font-family: Audiowide-Regular;
```

```
src: url(fonts/Audiowide-Regular.ttf);  
}
```

3、调用自定义字体库：

```
div {  
    font-family: Audiowide-Regular;  
}
```

二、文本段落样式

1、line-height（行高）

- normal：默认行间距
- 数字：高度，此数字会与当前字体尺寸相乘来设置行间距
- length：固定行间距
- %：当前字体尺寸的百分比间距
- 当单行文本的行高等于父容器高度时，可实现单行文本在容器中垂直方向居中对齐的效果。

2、text-align（文本水平对齐方式）

- left：左
- right：右
- center：居中

3、letter-spacing（字母间距）

- normal：默认间距
- length：固定字母间距（允许使用负值）

4、text-decoration（文本修饰）

- none：默认，标准文本(取消文字下划线，常用于超链接地址)
- underline：下划线
- overline：上划线
- line-through：穿过文本的线条

5、text-indent（文本首行缩进）

- length: 默认是0; 固定缩进
- %, 相对父级元素宽度的百分比缩进

6、text-transform (控制文本的大小写)

- none: 默认标准文本
- capitalize: 每个单词以大写字母开头
- uppercase: 全大写字母
- lowercase: 全小写字母

7、direction (设置文本方向)

- ltr: 默认从左到右
- rtl: 文本方向从右向左

8、white-space (设置元素中空白的处理方式)

- normal: 默认空白被浏览器忽略
- nowrap: 强制文本不换行, 直到遇到br标签

9、text-overflow (设置文字溢出)

- clip: 裁切
- ellipsis: 省略号

【备注:】实现文字过长隐藏, 用省略号代替效果需要同时设置以下:

- white-space: nowrap (强制文本不换行)
- 设置容器宽度width
- 设置溢出隐藏overflow: hidden
- 设置溢出显示方式为省略号显示方式

三、背景样式【重点及难点】

1、背景颜色: background-color: 颜色;

- 设置背景色
- 颜色名字: 例如red
- 颜色16进制数值: 例如#FF0000;
- rgb色值: rgb(255, 0, 0)、rgba(255, 0, 0, 0.5)
- transparent: 默认背景透明

2、背景图像：**background-image:url(图片地址);**

- url ()：指向背景图片
- none：无背景图像，默认

3、背景重复方式：**background-repeat**

- repeat：默认，垂直水平重复
- repeat-x：水平方向重复
- repeat-y：垂直方向重复
- no-repeat：不重复

4、背景图像位置：**background-position**：水平 垂直；（取值可以是：**关键字、百分比、绝对值**）

- 水平：left左、center中、right右
- 垂直：top上、center中、bottom下
- top left, top center, top right, center left, center center, center right, bottom left, bottom center, bottom right
- 如果只定义了第一个值，那么第二个必然是center
- 百分比 x%, y%：如果只定义了第一个值，另一个必然是50%
- 绝对值 x, y：**如果只定义了第一个，另一个必然是50%**

【备注：】

1)、css雪碧图实现背景就是基于背景定位（取值为绝对数值）。

2)、背景图片的显示原则：

- 当容器尺寸大于图片尺寸，背景图片将默认平铺，直至铺满元素；
- 当容器尺寸小于图片尺寸，只显示元素范围以内的背景区域。如果设置了位置，则按照位置的设置来显示；如果没有设置位置，则从背景图片左上角开始显示。

5、background复合写法

语法：

background: 背景颜色 背景图片 背景重复方式 背景位置 背景依附性；

例如：

```
background: #fff url(图片地址) no-repeat left center fixed;
```

属性值的顺序可调换，建议按照上述顺序书写。

复合写法中不支持背景尺寸。如果需要定义背景尺寸，需要单独定义background-size属性。

6、背景依附性：**background-attachment**

- 设置背景图像是否固定或者随着页面的其余部分滚动
- scroll: 默认值，背景图像会随着页面其余部分滚动而滚动
- fixed: 背景固定，不随滚动条滚动

7、背景尺寸 **background-size**: 宽度（水平） 高度（垂直）；（取值可以是：绝对值、百分比、关键字）

CSS3之前，背景图像的大小由图像实际大小决定，而CSS3可以指定背景图片的大小。可以设置成关键词、像素值或百分比大小（该百分比是相对于该背景所在元素的宽度和高度的百分比）。

- 20px 20px
- 50% 50%
- **auto** —— 设置背景宽度或高度后，另一个值可以使用auto自适应
- **contain** —— 背景长宽比例不变，将背景完整地显示在元素内，背景不会被裁切，元素的上下或者左右可能有留白。
- **cover** —— 背景长宽比例不变，将背景图像缩小或拉伸，铺满整个元素，元素上下和左右都不会留白，背景图片会有些部分被裁切。

8、同一元素中显示多个背景：

可以设置多个背景，用逗号隔开

- background-image:url(), url(), url();
- background-repeat:no-repeat , repeat-x , repeat-y;
- **background-position**:3% 98% , center , top;
- 多个背景的复合写法：
background:url("one.jpg") right bottom no-repeat , url("two.jpg") left top

repeat-x;

【以下三个跟背景相关的属性不常用，在学习到CSS3时会再次讲解】

9、background-origin （设置背景图片的定位区域，属性值有三个）

- border-box 背景从border区域就开始;
- padding-box 背景从padding区域开始;
- center-box 背景只从正文开始;


10、background-clip （设置背景颜色的绘制区域，属性值有两个）


- padding-box 背景颜色从padding区域开始;
- content-box 背景颜色只从正文开始;


11、背景颜色渐变gradient


四、CSS Sprites(CSS雪碧图)

(一)、概述:

 WORD文章标题

 Excel内容标题

 PDF内容标题

 文本文档标题

 主页

 空间

 网络

 其他

完成以上效果，多个小图标是使用img还是使用背景呢？

1. 背景图上可以显示文字、表格等内容；

2. img插入图片属于网页的**结构**，而背景图属于网页的**表现**，是修饰网页目的的；

3. img是通过src引入到当前文件的；背景是通过css的href引入到当前文件的；

1)、当浏览器解析到img元素指向到的图片文件时，**串行下载**，就会暂停后续其他资源的下载和处理，直到将该图片文件加载显示完毕。

2)、href是会**并行下载css**资源，**不会停止对当前HTML文档的处理**。

4. 网页制作中，能使用背景的地方就尽量少使用img。

(二)、CSS精灵（CSS雪碧图）概念：

- CSS Sprites，国内很多人叫CSS精灵或CSS雪碧图，是一种网页图片应用处理方式。通过图片整合技术，将一个页面涉及到的所有零星图片都包含到一张大图中去，当访问该页面时，载入的图片就不会像以前那样一幅一幅地慢慢显示出来，而是一次网络访问将整合到一起的所有小图标都获取到。
- CSS Sprites把网页中这些图片整合到一张图片文件中后，用的时候如何调用呢？其原理是利用CSS的“background-image”，“background-repeat”，“background-position”的组合进行背景定位，background-position可以用数字精确的定位出背景图片的位置。



上述页面效果的素材图标其实是：



(三)、CSS Sprites优缺点：

1、优点

- 利用CSS Sprites通过整合图片，能很好地减少对服务器的请求次数，从而加快页面加载速度，这也是CSS Sprites最大的优点，也是其被广泛传播和应用的主要原因。

- 加速的关键，不是降低质量，而是减少个数。传统切图讲究精细，图片规格越小越好，重量越小越好，其实规格大小无所谓，计算机统一都按byte计算。客户端每显示一张图片都会向服务器发送请求。所以，图片越多请求次数越多，造成延迟的可能性也就越大。

- CSS Sprites能减少图片的字节，多张图片合并成1张图片的字节总是小于原始的多张图片的字节总和。

- 解决了网页设计师在图片命名上的困扰，只需对一张集合的图片上命名就可以了，不需要对每一个小元素进行命名，从而提高了网页的制作效率。另外维护起来也更加方便。CSS Sprites非常值得学习和应用，特别是页面有一堆icon（图标）时。

2、缺点

诚然CSS Sprites是如此的强大，但是也存在一些不可忽视的缺点，如下：

- 在图片合并的时候，你要把多张图片有序的合理的合并成一张图片，还要留好足够的空间，防止板块内出现不必要的背景；这些还好，最痛苦的是在宽屏，高分辨率的屏幕下的自适应页面，你的图片如果不够宽，很容易出现背景断裂；

- CSS Sprites在开发的时候比较麻烦，你要通过photoshop或其他工具测量计算每一个背景单元的精确位置，这是针线活，没什么难度，但是很繁琐；建议使用类似CSS Sprites 样式生成工具，虽然还有一些使用上的不灵活，但是比photoshop测量来的方便多了，而且样式直接生成，复制即可。

（四）、CSS Sprite的实现步骤：

- 首先将小图片整合到一张大的图片上
- 然后根据具体图标在大图上的位置，给背景定位。background-position:-8px -95px;

例如：

```
.bg_sprite{background-image:url(/整图地址); background-repeat:no-repeat}
```

引用该类 .. 然后在元素中逐一定义背景坐标 .. 以下为关键属性 ..

#ico1 {width:容器宽度;height:容器高度;background-position:X坐标 Y坐标}

#ico2 {width:容器宽度;height:容器高度;background-position:X坐标 Y坐标}

#ico3 {width:容器宽度;height:容器高度;background-position:X坐标 Y坐标}

必须要限定容器大小符合背景图元素位置 .. 另外 .. XY轴是相对于整张图片的
左上角来算的 .. 所以取值一定要算清楚。