

更多嵌入式 Linux 学习资料, 请关注: 一口 Linux 回复关键字:1024



一、什么是 CSS?

样式表 (stylesheets) 的技术诞生于 1996 年底, 全称是层叠样式表

(Cascading Stylesheets—简称 CSS), 他可以对布局、字体、颜色、背景和其它文图效果实现更加精确的控制。

使用样式表只通过修改一个文件就改变大量的网页的外观和格式, 使网页的风格统一。

使用样式表的具体优势表现在以下几个方面:

1. 格式和结构分离 HTML 没有具体控制网页的格式或外观, 浏览器自己决定应该如何显示。层叠样式表通过将定义结构的部分和定义格式的部分分离, 使我们能够对页面的布局施加更多的控制。

HTML 仍可以保持简单明了的初衷。CSS 代码独立出来从另一角度控制页面外观。更好的控制页面全局的能力使用层叠样式表我们可以调整字号, 字体, 控制边距等等, 对页面的全局信息更精确的控制。

2. 可以制作出体积更小下载更快的网页 样式表只是简单的文本, 就象 HTML 那样。它不需要图象, 不需要执行程序, 不需要插件。它就象 HTML 指令那样快。

使用串接样式表可以减少表格标签及其它加大 HTML 体积的代码, 减少图象用量从而减少文件尺寸。

更快更容易地维护及更新大量的网页没有样式表时, 如果想更新整个站点中所有主体文本的字体, 必须一页一页地修改每张网页。 样式表的主旨就是将格式

和结构分离, 利用样式表, 我可以将站点上所有的网页都指向单一的一个 CSS 文件, 我只要修改 CSS 文件中某一行, 那么整个站点都会随之发生变动。

1. 建立第一张样式表

```
<html>
<head>
  <STYLE TYPE="text/css">
    <!--
      H4 { color: green; font-size: 37px; font-family: impact }
      P { text-indent: 1cm; background: yellow; font-family: courier }
    -->
  </STYLE>
  <title>建立第一张样式表</title>
</head>
<body>
  <h4>建立第一张样式表</h4>
  <p>知识创造财富,学习改变人生!</p>
</body>
</html>
```

该例子解释如下:

```
H4{ color: green }
```

它告诉浏览器将所有 `<H4></H4>` 包围的文字以绿色显示, 每一条规则包含一个选择及说明。

H4 就是选择, 它是一个附带样式功能的 HTML 标签。

说明用于定义实际的样式, 包括两部分: 属性 (本例中即 color) 和参数 (green)。

任何 HTML 标签都可用作标签。所以你可以将样式表的信息附加到任何要素。

从通常的 `<P>` 到 `<CODE>` 及 `<TABLE>` 内容。

甚至可以通过将样式表用于 `` 将层叠样式表的属性用于图象。

继承性

样式表的规则可从母体延续到子体

下面是一个例子:

```
B { color: blue }
```

这项规则告诉浏览器将所有之内的文字用蓝色显示。

但是在下列情况下, 浏览器该如何处理呢?

```
<B>我的邮箱是:<I>21689991@qq.com</I></B>
```

对于 `标签` 并没有设定样式, 但因为位于之中, 所以它将继承母体设定的样式, 也以蓝色显示.

2. 把样式表加入到网页中

方法 1: `<LINK>` 标签:

```
<html>
  <head>
    <title>建立第一张样式表</title>
    <LINK REL=stylesheet HREF="mystyles.css" TYPE="text/css">
  </head>
  <body>
    <h4>建立第一张样式表</h4>
    <p>知识创造财富, 学习改变人生!</p>
  </body>
</html>
```

使用链接的样式表时, 你无须使用注释标签, 现在还要生成一个额外的样式表文件, 我们暂时命名为 `mystyles.css`

文件内容如下:

```
H4 { color: green; font-size: 37px; font-family: impact }
P { text-indent: 1cm; background: yellow; font-family: courier }
```

- 在 `HREF` 属性中你可以选择使用绝对或相对 URL.

显示结果

建立第一张样式表

知识创造财富, 学习改变人生!

方法 1:@import

其中 css 文件的导入还可以用下面这种方法

```
<html>
<head>
  <title>建立第一张样式表</title>
  <style>
    @import url(./css/importcss.css);
  </style>
</head>
<body>
  <h4>建立第一张样式表</h4>
  <p>知识创造财富, 学习改变人生!</p>
</body>
</html>
```

如果同时用 link 和 import, 如下所示

```
<html>
<head>
  <title>建立第一张样式表</title>
  <LINK REL=stylesheet HREF="./css/linkcss.css" TYPE="text/css">
  <style>
    @import url(./css/importcss.css);
  </style>
</head>
<body>
  <h4>建立第一张样式表</h4>
  <p>知识创造财富, 学习改变人生!</p>
</body>
</html>
```

其中 mystyles1.css 我们修改如下 (H4 颜色修改为 red 红色):

```
H4 { color: red; font-size: 37px; font-family: impact }
P { text-indent: 1cm; background: yellow; font-family: courier }
```

结果显示:

建立第一张样式表

知识创造财富, 学习改变人生!

调换下顺序:

```
<html><head>
<title>建立第一张样式表</title>
<style>
@import url(mystyles1.css);
</style>
<LINK REL=stylesheet HREF="mystyles.css" TYPE="text/css">
</head>
<body>
<h4>建立第一张样式表</h4>
<p>知识创造财富, 学习改变人生!</p>
</body>
</html>
```

可见后者会覆盖前者样式。

3. 分类引用

我们先前已经讲过所有的 HTML 标签都可以用作选择, 并附加样式表说明. 但是还有更复杂的情况, 比如,

你想让主体文字的 1 段用绿色显示, 第 2 段用紫色显示, 而第 3 段用灰色显示,

这种情况下可以将段落 P 分成 3 种不同的类别, 每一类应用不同的样式表说明. 这些规则将以以下方式显示:

```
P.first { color: green }
P.second { color: purple }
P.third { color: gray }
```

你的 HTML 代码可以这样使用样式表:

```
<P CLASS=first>第一段</P>
<P CLASS=second>第二段</P>
<P CLASS=third>第三段</P>
```

还可以生成不加任何 HTML 标签的分类:

```
.first { color: green }
```

这种方式更加灵活, 因为现在我们可以将 CLASS=first 用于任何 HTML 标签, 而设定的文字将以绿色显示。

实例:

```
<html><head>
<title>建立第一张样式表</title>
<LINK REL=stylesheet HREF="css/mystyles4.css" TYPE="text/css">
</head>
<body>
```

```
<h4>建立第一张样式表</h4>
<p class=first>知识创造财富,学习改变人生!</p>
<p CLASS=second>知识创造财富,学习改变人生!</p>
<p CLASS=third>知识创造财富,学习改变人生!</p>
<p CLASS=fourth>知识创造财富,学习改变人生!</p>
</body>
</html>
```

mystyles.css

```
H4 { color: green; font-size: 37px; font-family: impact }
P { text-indent: 1cm; background: yellow; font-family: courier }
P.first { color: green }
P.second { color: purple }
P.third { color: gray ; font-size: 37px}
.fourth { color: red ; font-size: 56px; background: blue;}
```

4. 字体控制 font-family

本节讲述如何控制字号、字体。

字体系列就是 CSS 中要使用的一系列字体。其基本语法如下:

```
H2 { font-family: helvetica,impact, sans-serif }
```

这时 Web 浏览器阐释样式表的规则: 首先在列表中寻找字体的名称 (helvetica), 如果在该计算机中安装了这种字体, 就使用它。如果没有安装, 则移向下一字体 (impact), 如果这种字体也没有安装, 则移向第 3 种字体 (sans-serif)。sans-serif 字体参数是浏览器可以依据的最后一种参数, 它告诉浏览器使用任何一种缺省 sans-serif 字体 (或许就是 Arial)。

如果一种字体的名称不只一个单词, 如 Gill Sans, 在你的 CSS 代码中用引号包含该字体的名称。

```
BODY { font-family: "gill sans", "new baskerville", serif }
```

行内加入的 CSS 规则, 使用单引号:

```
<P STYLE="font-family: 'gill sans', 'new baskerville', serif">这里是文本!</P>
```

5. 字号控制 font-size

在 `html` 中对字号的控制是很有限的, 你只能设定 7 种字号, 但是用样式表你可以更自由的设定字号. 使用 `font-size` 属性, 你可以对文字的尺寸进行无限的控制.

1)points

```
P { font-size: 16pt }
```

这条代码告诉浏览器以 16 points (点) 的尺寸显示`<P>`。Points 是确定文字尺寸非常好的单位, 因为它在所有的浏览器和操作平台上都适用。

2)px

```
P { font-size: 20px }
```

从网页设计的角度来说, pixel (象素) 是一个非常熟悉的单位, 它最大的优点就在于所有的操作平台都支持 pixel 单位

6. 字体风格

利用 CSS 加入斜体字非常简单, 而且我们对于加粗字体还有很多选择。

1) 倾斜字体 italic

```
H3 { font-style: italic }
```

当浏览器接到应用斜体字的指令后, 它将用户的机器中寻找安装的斜体字版本, 如果不存在斜体字, 浏览器将会把正常的字体倾斜显示。

2) 加粗字体 bold

```
P { font-weight: bold }
```

还可以用数值 100, 200, ... 900。正常的非加粗的字体值为 400, 900 是加粗字体的最大值。

3) 下划线 underline

```
B { text-decoration: underline }
```

还有其他的修饰内容比如:

`overLine` 给文字上划线

`line-through` 给文字划出删除线,

`none` 使得上述效果都不会发生。

`none` 参数非常有用, 你可以使链接的文字不以下划线的形式显示。

具体做法:

```
A:link { text-decoration: none }
```

```
A:active { text-decoration: none }
```

```
A:visited { text-decoration: none }
```

实例

```
<Html>
<head>
<title>cssStyle 字体控制</title>
<link rel=stylesheet href="css/mystyles.css" type="text/css">
</head>
<body>
<H3>css Style 控制字体</H3>
<p class="first">控制字号</p>
<p class="second">控制字号</p>
<p class="three">控制字号</p>
<p class="boldWord">这一行是粗体</p>
<p class="boldWord100">这一行是 100 号粗体</p>
<p class="italicWord">这一行是斜体</p>
<p class="underlineWord">使用下划线</p>
<p class="overlineWord">使用上划线</p>
<p class="delword">使用删除线</p>
</body>
</Html>
```

mystyles.css

```
H3 { font-family: helvetica, impact, sans-serif }
```

```
P.first { font-size: 8pt }
```

```
P.second { font-size: 16pt }
```

```
P.three { font-size: 32pt }
```

```
p.boldWord{font-weight:bold}
```

```
p.boldWord800{font-weight:100}
```

```
p.italicWord{font-style:italic}
```

```
p.underlineWord{text-decoration: underline }
```



```
p.overlineWord{text-decoration: overline }
```

```
p.delWord{text-decoration: line-through }
```

css Style 控制字体

控制字号

控制字号

控制字号

这一行是**粗体**

这一行是 100 号**粗体**

这一行是**斜体**

使用**下划线**

使用**上划线**

使用**删除线**

8. 文字的定位

本节讲述将学习文字和每行的内容之间是如何定位的。

1) 字间距 *word*、*letter-spacing*

利用字(词)间距属性, 你可以在字之间加入更大的距离:

```
H3 { word-spacing: 10px }
```

字母间距:

```
H3 { letter-spacing: 10px }
```

2) 行高 *line-height*

利用行高我们可以随心所欲地控制行与行之间的垂直距离。

```
B { line-height: 16pt }
```

当你用数字设定行高的时候, 浏览器将利用字号来确定行距: 它将字号乘以设定的参数值。所以, 在下面的例子中, 行高将是 24 点。

```
B { font-size: 12pt; line-height: 2 }
```

用比例设定行距

```
B { font-size: 10pt; line-height: 140% }
```

在本例中, 行距是长度 10points 的 140%, 即 14points。

3) 文字对齐 *vertical*、*text-align*

利用文字对齐属性, 你可以控制段落的水平对齐:

```
H4 { text-align: center }
```

这项属性只用于整块的内容, 如<P>、<H1>-<H6>、<BLOCKQUOTE>和。

以下为各种选项:

left 指将要素左对齐

right 指将要素右对齐

center 指将要素居中

控制垂直对齐

```
H4 { vertical-align: top }
```

垂直对齐可以使你控制文字或替代要素 (比如图象) 相对于其母体要素的垂直位置。例如, 如果你将一个 2x2 象素的 GIF 图象同其母体要素文字的顶部垂直对齐, 则该 GIF 图象将在 该行文字的顶部显示。

以下为各项参数:

top 将要素顶部同最高的母体要素对齐。

bottom 将要素的底部同最低的母体要素对齐。

text-top 将要素的顶部同母体要素文字的顶部对齐。

text-bottom 将要素的底部同母体要素文字的底部对齐。

baseline 将要素的基准线同母体要素的基准线对齐。

middle 将要素的中点同母体要素的中点对齐。

sub 将要素以下标的形式显示。

super 将要素以上标的形式显示。

4) 文字缩行 *text-indent*

```
P { text-indent: 40% }
```

该属性只应用于整块的内容。你可以使用任何一种熟悉的长度单位设定缩行。还可以使用比例参数。例如, 本例的缩行为 40% 的比例, 即第 1 行的起始

位置比其正常位置缩进了 40%, 如果将文字缩行赋予一个负值, 则文字第 1 行的起始位置比其正常位置还偏左。

5) 控制边距, 边框和空白填充 *margin*、*border*、*padding*

我们看一下边距, 边框, 空白填充的定义:

围绕要素的空白填充 (*padding*)

围绕空白填充的边框 (*border*)

围绕边框的边距 (空白—*margin*)

图示如下:



控制顶边距、底边距、左边距和右边距

这 4 项属性可以使你控制一个要素的四周的边距。

如下:

```
H4 { margin-top: 20px; margin-bottom: 5px; margin-left: 100px; margin-right: 55px }
```

设定边距参数值最明显的方法是利用长度位: *px* (象素)、*pt* (点数) 等。你也可以用比例值 设定边距。

控制顶空白填充, 底空白填充, 左空白填充和右空白填充 空白填充的作用类似于边距控制, 可以设定一个要素的前后左右的空白填充的尺寸。

```
H4 { padding-top: 20px; padding-bottom: 5px; padding-left: 100px; padding-right: 55px }
```

控制位于边距和空格填充之间的边框 控制顶边框宽度, 底边框宽度, 左边框宽度和右边框宽度 你可以控制整个边框的宽度, 也可以分别控制每一边的边框宽度:

```
H4 { border-top-width: 2px; border-bottom-width: 5px; border-left-width: 1px; border-right-width: 1px }
```

如果你想使边框四面应用相同的宽度, 很简单, 使用 `border-width` 标签就可一次完成, 例:

```
IMG { border-width: 2px }
```

设置边框颜色

```
P { border-color: green; border-width: 3px }
```

如果你想使边框的四边显示不同的颜色, 你可以在设置中分别列出各种颜色:

```
P { border-color: #666699 #FF0033 #000000 #FFFF99; border-width: 3px }
```

浏览器将第 1 种颜色理解为顶边框的颜色参数值, 第 2 种颜色为右边框, 然后是底边框, 然后是左边框。

实例

```
<html>
<head>
<title>cssStyle 字体定位</title>
<link rel=stylesheet href="myStyles.css" type="text/css">
</head>
<body>
<p class="wrodsp10">word spacing is 10</p>
<p class="wrodsp20">word spacing is 20</p>
<p class="Lettersp10">Letter spacing is 10</p>
<p class="Lettersp20">Letter spacing is 20</p>
<p class="lineH">生活的理想,<br>就是理想的生活!</p>
<h5 class="left">既然选择了远方,便只顾风雨兼程!</h5>
<h5 class="center">既然选择了远方,便只顾风雨兼程!</h5>
<h5 class="right">既然选择了远方,便只顾风雨兼程!</h5>
<p class="textindent">有一对兄弟,有一天他们出去爬山,然后一起回家。他们的家住在八十层楼,他们一人背着一大包的行李回家,却发现大楼停电了。于是哥哥就说:弟弟,我们一起爬楼梯上去吧。
于是他们就一起爬上去。
到了二十楼的时候,哥哥
又告诉弟弟:
包包太重了,我们把它放在二十楼,我们爬上去,明天再下来拿。
```

弟弟说: 好。

</p>

<p class="margin"> 于是他们就把他们的包包放在二十楼, 继续往上爬。到了四十楼, 弟弟开始抱怨, 于是就跟哥哥吵起来了。他们边吵边爬, 爬到了六十楼, 哥哥就对弟弟说: 只剩二十层楼了, 我们不要吵了, 默默的爬完它吧 !! 于是他们就各走各的, 终于到了家门口。哥哥就摆出了很帅的姿势: 弟弟开门。

弟弟就对哥哥说: 别闹了, 钥匙不是在你那儿。

结果, 他们把钥匙留在二十楼的包包里了。

</p>

<p class="border"> 于是他们就把他们的包包放在二十楼, 继续往上爬。到了四十楼, 弟弟开始抱怨, 于是就跟哥哥吵起来了。他们边吵边爬, 爬到了六十楼, 哥哥就对弟弟说: 只剩二十层楼了, 我们不要吵了, 默默的爬完它吧 !!

</p>

<p class="double"> 于是他们就把他们的包包放在二十楼, 继续往上爬。到了四十楼, 弟弟开始抱怨, 于是就跟哥哥吵起来了。他们边吵边爬, 爬到了六十楼, 哥哥就对弟弟说: 只剩二十层楼了, 我们不要吵了, 默默的爬完它吧 !!

</p>

</body>

</html>

myStyle.css

p.wrodsp10{word-spacing:10px}

p.wrodsp20{word-spacing:20pt}

p.lettersp10{letter-spacing:10pt}

p.lettersp20{letter-spacing:20pt}

p.lineH{line-height:40pt}

h5.left{text-align:left}

h5.right{text-align:right}

h5.center{text-align:center}

.textindent{text-indent:4%}

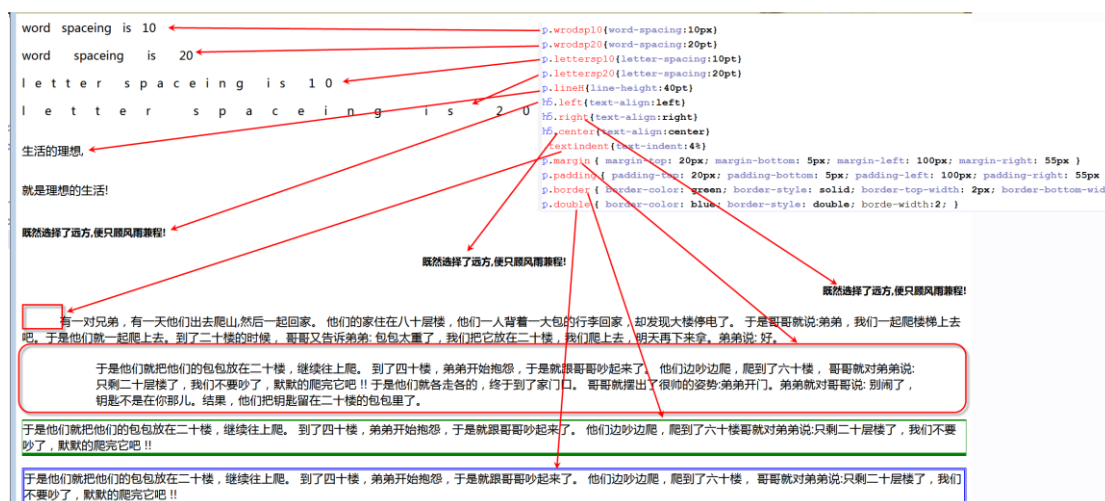
p.margin { margin-top: 20px; margin-bottom: 5px; margin-left: 100px; margin-right: 55px }

p.padding { padding-top: 20px; padding-bottom: 5px; padding-left: 100px; padding-right: 55px }

p.border { border-color: green; border-style: solid; border-top-width: 2px; border-bottom-width: 5px; border-left-width: 1px; border-right-width: 1px }

p.double { border-color: blue; border-style: double; border-width: 2px }

显示效果:



9. 控制颜色和背景

利用 CSS 在网页要素后面加入固定的背景色及图象,你甚至可以将要素只加到某一个单词后面。

1) 背景色 `background-color`

```
P.yellow { background-color: #FFFF66 }
```

2) 背景图片 `background-image`

你可以很轻松地将 GIF 或 JPEG 图片设置成背景图片:

```
B { background-image: url(background.gif) }
```

该规则将背景图象加到整个段落之后。

可以将 GIF 图象采用和背景图象通常的设置类似的方法平铺到文字后面,你还可以将背景图象只添加到两个单词后面。

可以使用 URL 调用一幅图象,你既可以使用相对 URL,例如 `images/bg.gif` 或完整的 URL

3) 控制背景图片

不仅可以将图象放在要素后面,你还可以精确地控制背景图象的各项设置。

可以决定是否及如何将其平铺, 背景图象应该滚动还是保持固定, 及将其放在什么位置。

4) 背景重复(background-repeat)

```
P { background-repeat: no-repeat; background-image: url(background.jpg) }
```

上述规则应用于整个段落。

如果你只想让图象垂直或者水平方向平铺, 可以使用 `repeat-x` 将其水平平铺, 用 `repeat-y` 将其垂直平铺。

而 `repeat` 参数值则将图象从水平和垂直两个方向平铺。

5) 固定背景 background-attachment

在 HTML 中, 背景图象通常会随页面的滚动而一起滚动。但利用 CSS 的固定背景 (`background`) 属性, 你可以建立不滚动的背景图象, 页面滚动时, 背景图象可以保持固定。

```
BODY { background-attachment: fixed; background-image: url(background.jpg) }
```

这个例子可以看到页面滚动时, 背景仍保持固定。

其参数有两种选择:

- `scroll` 指背景图象随文字内容一起滚动, 即通常所见的方式。
- `fixed` 指文字滚动时, 背景图象保持固定。

6) 背景定位(background-position)

`background-position` 可以设定将背景插在什么位置显示。

```
P { background-position: center bottom; background-image: url (background.jpg) }
```

当上述 CSS 规则应用于本段文字时, 背景图象将在本段的中下部开始显示并向右平铺。

设定位置的 3 种方法:

- 1) 关键字参数 (Keyword values) 关键字设定方法简便易用:

`top` 将背景图象同前景要素的顶部对齐。

`bottom` 将其同前景要素的底部对齐。

`left` 将其同左边对齐。

right 将其同右边对齐。

center 将其水平居中 (如果使用在另一关键字前面) 或垂直居中 (如果用在另一关键字后面)。

- 2) 长度参数

长度参数可以使用你对背景图象的位置作出更精确的控制。你可以设定水平和垂直定位起点, 例:

```
P { background-position: 70px 10px; background-repeat: repeat-y; background-image: url (background.jpg) }
```

将 GIF 图象设置在从左上角起水平 70 象素, 垂直 10 象素的位置。由于设置了 **background-repeat: repeat-y**, 图象将在文字后垂直平铺。

- 3) 比例值 (Percentage values)

还可以使用比例值设定背景图象的位置, 例:

```
P { background-position: 70% 50%; background-image: url (background.jpg) }
```

当这条 CSS 规则应用于本段时, 背景图象从水平距离段落右端 70%, 垂直距离段落顶部 50% 的位置显示。

如果重新调整浏览器窗口的尺寸, 使得段落的显示尺寸变化, 背景图象的位置也相应发生变化。

综合实例 1

```
<html>
<head>
<title>cssStyle 字体定位</title>
<link rel=stylesheet href="cssStyle.css" type="text/css">
</head>
<body>
<b class="color">既然选择了远方,便只顾风雨兼程!</b>
<p class="yellow">江雪<br>
千山鸟飞绝,<br> 万径人踪
灭.<br> 孤舟蓑笠翁,<br> 独
钓寒江雪.</p>
<p class="image">江雪<br>
千山鸟飞绝,<br> 万径人踪
灭.<br> 孤舟蓑笠翁,<br> 独
钓寒江雪.</p>
<p class="imagenorep">江雪<br>
```



```
千山鸟飞绝,<br> 万径人踪  
灭.<br> 孤舟蓑笠翁,<br> 独钓寒  
江雪.</p>  
</body>  
</html>
```

cssStyle.css

```
b.color{color:blue}  
p.yellow{background-color:yellow}  
p.image{background-image:url(caomei.jpg)}  
p.imagenorep{background-repeat:no-repeat;background-image:url(caomei.jpg)}  
body.fiximage{background-attachment: fixed; background-image: url(caomei.jpg)}  
body.scrollimage{background-attachment: scroll; background-image: url(caomei.jpg)}
```

显示结果:

既然选择了远方,便只顾风雨兼程!

江雪
千山鸟飞绝,
万径人踪灭。
孤舟蓑笠翁,
独钓寒江雪。

江雪
千山鸟飞绝,
万径人踪灭。
孤舟蓑笠翁,
独钓寒江雪。

江雪
千山鸟飞绝,
万径人踪灭。
孤舟蓑笠翁,
独钓寒江雪。

综合实例 2

cssStyle.css 使用上例

```
<html>  
<head>  
<title>控制图片固定背景</title>  
<link rel=stylesheet href="cssStyle.css" type="text/css">  
</head>  
<body class="fiximage">  
<b class="color">既然选择了远方,便只顾风雨兼程!</b>  
<p>有一对兄弟,有一天他们出去爬山,然后一起回家。 他们的家住在八十层  
楼,他们一人背着一大包的行李回家,却发现大楼停电了。 于是哥哥就说:弟  
弟,我们一起爬楼梯上去吧。  
于是他们就一起爬上去。
```

```
到了二十楼的时候, 哥哥  
又告诉弟弟:  
包包太重了, 我们把它放在二十楼, 我们爬上去, 明天再下来拿。 弟  
弟说: 好。 于是他们就把他们的包包放在二十楼, 继续往上爬。 <br>  
到了四十楼, 弟弟开始抱怨, 于是就跟哥哥吵起来了。 他们边吵边  
爬, 爬到了六十楼, 哥哥就对弟弟说: 只剩二十层楼了, 我们不要吵  
了, 默默的爬完它吧 !! 于是他们就各走各的, 终于到了家门口。 哥哥  
就摆出了很帅的姿势: 弟弟开门。  
弟弟就对哥哥说: 别闹了, 钥匙不是在你那儿。  
结果, 他们把钥匙留在二十楼的包包里了。 <br>  
这个故事其实在反映我们的人生很多人在二十岁以前是活在家人的期望和老师的期许之下,  
背负着很多的压力; 在二十岁之后  
离开了众人的压力, 开始满腔的热  
血, 开始有很多的梦想要完成;  
可是工作了二十年后, 开始发觉工作不如意... 于是就开始抱怨老板、抱怨公  
司、抱怨社会、抱怨政府。 就在这抱怨中又渡过了二十年。 于是告诉自己,  
六十岁了没什么好抱怨的了, 就默默的走完自己的这些年吧。 到了八十岁快要  
死掉的前夕, 才想起自己好像有什么事还没完成 ..... 原来, 他二十岁的梦想  
还没有完成。 <br>  
所以, 希望现在的你 .... 充满着理想... 充满着热情 .... 去做自己想做的事。  
人生短短数十载, 也没有什么好担心的不是吗? 尤其是在经历那么多的遭  
遇, 更应该能体会, 把握现在。 <br>  
记得吗? 今天就是礼物! </p>  
</body>  
</html>
```

显示结果:



可以看到背景图片 repeat, 并且右侧有滚动条, 修改 background-

attachment 属性为 *fixed/scroll*, 可以分别看到背景图片固定/跟随滚动现象。

10. 要素的定位 *position*

1) 绝对定位

绝对定位使你能精确地定位要素在页面的独立位置, 而不考虑页面其它要素的定位设置。 定位 (左边和顶部)

```
H4 { position: absolute; left: 100px; top: 43px }
```

这项 CSS 规则让浏览器将<H4>的起始位置精确地定在距离浏览器左边 100 像素, 距离其顶部 43 像素的位置。

注意这里唯一设置的是左边和顶部, 也就是说, 文字将从左到右, 从上到下载入浏览窗口。

2) 相对定位

相对定位指你所定位的要素的位置相对于在文件中所分配的位置。

例:

```
I { position: relative; left: 40px; top: 10px }
```

相对定位的关键在于定位了的要素的位置是相对于它通常应在的位置进行定位。 如果停止使用相对定位, 则文字的显示位置将恢复正常。

3) 控制要素的宽度

```
DIV { position: absolute; left: 200px; top: 40px; width: 150px }
```

浏览器接到这项规则时, 它将文字按照规则规定的效果显示, 还将段落的最大水平尺寸限制在 150 像素。 可以通过宽度设置人为地拉宽或压缩图象。

4) 控制高度

高度应该和宽度的设置类似, 只不过是在垂直方向上:

```
DIV{position:absolute;left:200px;top:40px;height:150px}
```

5) 可视性

利用 CSS, 你可以隐藏要素, 使其在页面上看不见.

```
H4 { visibility: hidden }
```

- 选项

visible 使要素可以被看见

hidden 使要素被隐藏

inherit 指它将继承母体要素的可视性设置.

当一个要素被隐藏后, 它仍然要占据浏览器窗口中的原有空间。所以, 如果你将文字包围在一幅被隐藏的图象周围, 那么, 其显示效果将是文字包围着一块空白区域。