

零基础CSS入门教程

乘风者系列图书

CHENWENYANG
陈文阳

乘风者计划专家博主
高校软件技术专业教师



CSS零基础入门级教学
实践与知识内容相结合
CSS常用语法汇总与详解

包含详细的CSS入门知识与实践，程序员快速入门CSS的必备手册



关注【程序员大阳】

学习更多零基础教程



阿里云开发者“藏经阁”

海量电子手册免费下载

目录

零基础 CSS 入门教程(1)--初识 CSS.....	5
零基础 CSS 入门教程(2)--第一段 CSS 代码.....	6
零基础 CSS 入门教程(3)--文本样式.....	10
零基础 CSS 入门教程(4)--字体样式.....	16
零基础 CSS 入门教程(5)--背景样式.....	23
零基础 CSS 入门教程(6)--元素选择器.....	30
零基础 CSS 入门教程(7)--id 选择器.....	35
零基础 CSS 入门教程(8)--class 选择器.....	39
零基础 CSS 入门教程(9)--注释.....	43
零基础 CSS 入门教程(10)--列表样式.....	45
零基础 CSS 入门教程(11)--边框样式.....	51
零基础 CSS 入门教程(12)--分组选择器.....	56
零基础 CSS 入门教程(13)--嵌套选择器.....	60
零基础 CSS 入门教程(14)--尺寸.....	64
零基础 CSS 入门教程(15)--外边距.....	68
零基础 CSS 入门教程(16)--内边距.....	71
零基础 CSS 入门教程(17)--表格样式.....	75
零基础 CSS 入门教程(18)--颜色值.....	85

零基础 CSS 入门教程(19)--盒子模型.....	89
零基础 CSS 入门教程(20)--显示.....	92
零基础 CSS 入门教程(21)--透明.....	97
零基础 CSS 入门教程(22)--溢出.....	99
零基础 CSS 入门教程(23)--伪类.....	104
零基础 CSS 入门教程(24)--外联选择器.....	106
零基础 CSS 入门教程(25)--选择器优先级.....	111
零基础 CSS 入门教程(26)--按钮实例.....	114
零基础 CSS 入门教程(27)--图片实例.....	120
零基础 CSS 入门教程(28)--导航栏实例.....	124
零基础 CSS 入门教程(29)--下拉菜单实例.....	130
零基础 CSS 入门教程(30)--布局实例.....	136
零基础 CSS 入门教程(31)--代码格式化.....	145

零基础 CSS 入门教程(1)--初识 CSS

1. 前言

学习 CSS 是离不开 HTML 的，HTML 是 CSS 的基础，如果还没学过 HTML 的，可以先去学习下我的《零基础 HTML 入门教程》。

2. HTML 与 CSS 区别

HTML 用来描述网页的内容，而 CSS 用于给内容添加样式。

如果把网页设计比作房屋装修，HTML 内容就是房子里有哪些家具——沙发、茶几、床、餐桌，都是内容。

而 CSS 是对内容的进一步描述——沙发是几座的？茶几的大小？床的颜色？餐桌的高度？沙发和茶几的距离？

也就是说，网页上有什么东西，是 HTML 决定的，但是这些东西的外观样式，是 CSS 决定的。

3. CSS 有哪些功能

CSS 可以描述字体、颜色、背景色、尺寸、距离、边框、布局等方面的内容，功能还是比较多的。

4. 如何学习 CSS

学习 CSS 的方法，很简单，多学多练、不懂就问。

相对于 HTML 来说，CSS 要更加复杂一点，更加抽象一点。但是总体上 CSS 还是一门设计语言，比较直观，比较易懂的。

零基础 CSS 入门教程(2)--第一段 CSS 代码

1. 前言

上一篇介绍了 CSS 的作用，就是用来设置网页的样式的。本篇就用一个最基础的入门实例，来展示下 CSS 是如何将样式设置给网页的。

2. 样式的应用对象

首先明白第一点，CSS 样式的应用对象是什么。

CSS 是用来设置网页样式的，它是给内容添加装饰的，所以 CSS 样式需要设置到网页内容——也就是要设置到 HTML 元素上的。

一个网页可能有很多元素，例如有 table、form、div、span、img 等等，我们在设置样式的时候，需要选择设置到具体哪个元素上。

例如下面一段代码，我们希望修改春晓 作者：孟浩然的文本颜色为绿色，那么我们就需要将样式应用到第一个 p 标签。

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS 第一个实例</title>
    <meta charset="utf-8">
</head>
<body>
```

```
<p>
    春晓 作者：孟浩然
</p>
<p>
    春苗不觉晓
</p>
<p>
    处处闻啼鸟
</p>
<p>
    夜来风雨声
</p>
<p>
    花落知多少
</p>
</body>
</html>
```

3. 样式的应用方式

好的，现在我们想将绿色这个样式应用到上面的第一个标签，也就是下面这段代码上。

```
<p>
    春晓 作者：孟浩然
</p>
```

那么如何应用呢，其实可以通过 HTML 的 `style` 属性来实现。也就是说，`style` 属性的值就是我们要编写的 CSS 代码部分。

通过 `style` 属性，我们可以将 CSS 样式代码应用到指定的标签上。

看下面的代码，我们给 p 标签设置了 style 样式，style 属性的值 color:green; 就是 CSS 代码。

```
<p style="color:green;">  
    春晓 作者：孟浩然  
</p>
```

4. CSS 代码解析

我们已经对 HTML 很熟悉了，HTML 是标签式语言，它的格式如下代码，其中 `labelName` 为标签名，`content` 为标签内容，`propertyName` 为属性名，`propertyValue` 为属性值。

```
<labelName propertyName="propertyValue">content</labelName>
```

那么 CSS 的格式时怎么样的呢，我们分析下，如下代码中，`color` 表示样式名，`:` 用于区分前面名和后面的值，`green` 表示值，`;` 表示结束符号。

```
color:green;
```

这一段代码解释一下，就是将标签内文本颜色（`color`）设置为绿色（`green`）。

我们现在打开网页预览，效果如下。



5. 小结

从上面的例子其实可以发现，CSS 通过 style 属性将 CSS 代码应用与标签。

我们只需要记住一些 CSS 样式的名称和取值，就能掌握 CSS 了。

大家可以试一下将 green 修改为其他颜色，例如 red、black，能否有效？

零基础 CSS 入门教程(3)--文本样式

1. 前言

上一篇我们已经讲解了如何将 CSS 样式设置到标签上，然后也演示了 CSS 语言格式特点。

所以之后的学习就很简单了，我们学习下新的样式名称，取值和对应的效果，然后自己多写一些代码练习熟练，就可以了。

本篇，我们就来学习下文本相关的样式。

2. 文本颜色

可以使用 `color:xxx;` 来设置文本颜色，其中 `xxx` 是颜色的名称如 `red/green/yellow`。示例如下：

```
<body>
  <div>
    文本颜色演示：
    <p style="color:red;">
      春眠不觉晓
    </p>
    <p style="color:green;">
      处处闻啼鸟
    </p>
    <p style="color:blue;">
      夜来风雨声
    </p>
    <p style="color:yellow;">
      花落知多少
    </p>
  </div>
</body>
```

```
</p>
</div>
</body>
```

我们呢设置了 4 种颜色，所以效果如下：

文本颜色演示：

春苗不觉晓

处处闻啼鸟

夜来风雨声

处处闻啼鸟

3. 文本对齐

什么叫做对齐呢，就好比我们军训的时候参加队列练习，教官会说“向左看齐”，其含义就是左对齐。

对齐的意思就是向左对齐、向右对齐。还有一种叫做居中对齐，就是不管有几个人，都在中间呆着。

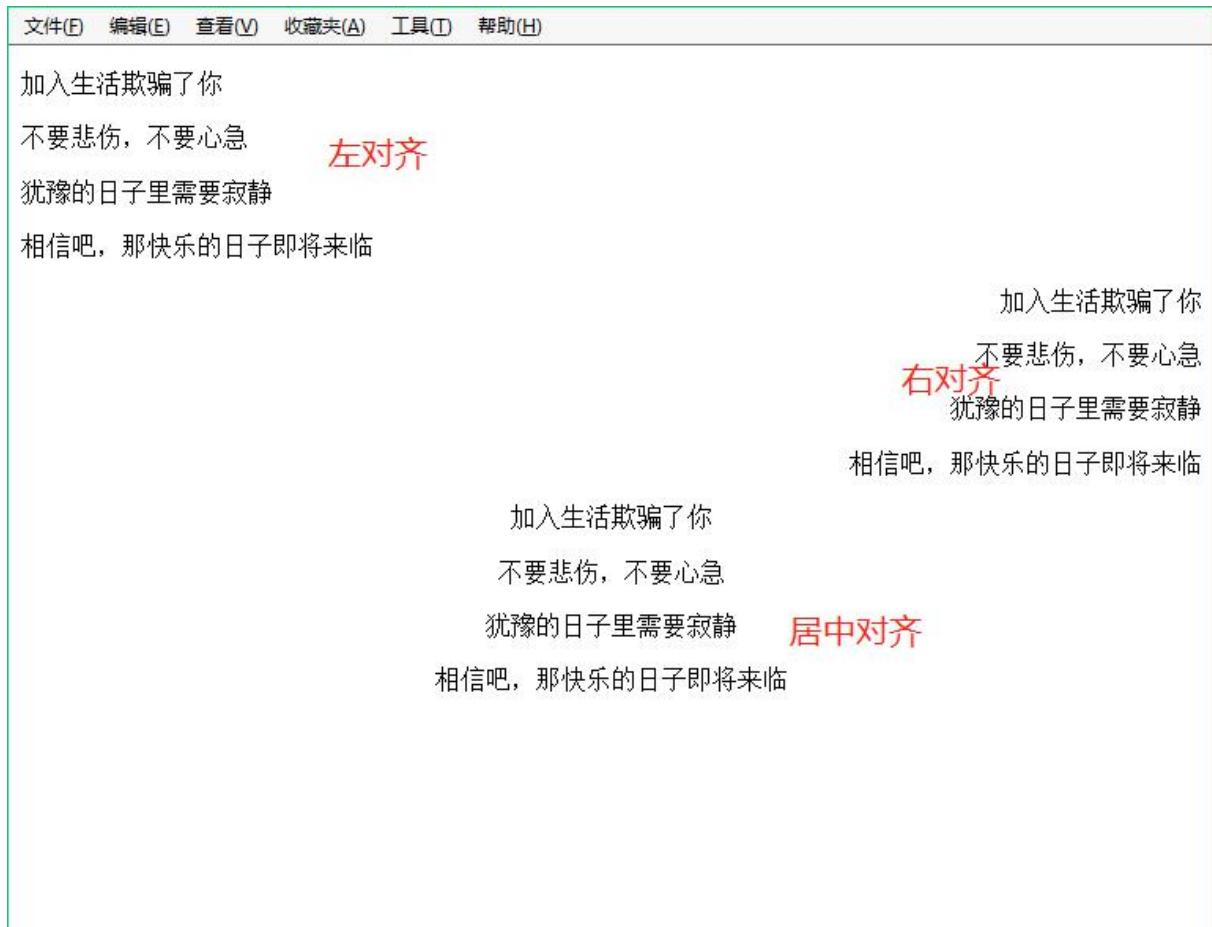
接下来我们用代码更直观的展现文本对齐：

```
<!-- 左对齐 -->
<div style="text-align: left;">
  <p>
    加入生活欺骗了你
  </p>
  <p>
    不要悲伤，不要心急
  </p>
```

```
<p>
    犹豫的日子里需要寂静
</p>
<p>
    相信吧，那快乐的日子即将来临
</p>
</div>
<!-- 右对齐 --&gt;
&lt;div style="text-align: right;"&gt;
    &lt;p&gt;
        加入生活欺骗了你
    &lt;/p&gt;
    &lt;p&gt;
        不要悲伤，不要心急
    &lt;/p&gt;
    &lt;p&gt;
        犹豫的日子里需要寂静
    &lt;/p&gt;
    &lt;p&gt;
        相信吧，那快乐的日子即将来临
    &lt;/p&gt;
&lt;/div&gt;
<!-- 居中对齐 --&gt;
&lt;div style="text-align: center;"&gt;
    &lt;p&gt;
        加入生活欺骗了你
    &lt;/p&gt;
    &lt;p&gt;
        不要悲伤，不要心急
    &lt;/p&gt;
    &lt;p&gt;
        犹豫的日子里需要寂静
    &lt;/p&gt;
&lt;/div&gt;</pre>
```

```
</p>
<p>
    相信吧，那快乐的日子即将来临
</p>
</div>
```

在上面的例子中，我们分别对三个 div 设置了左对齐，右对齐，居中对齐，所以 div 中的段落就会采用相应的对齐方式，所以效果如下：



4. 文本装饰线

提到文本装饰线，这个名字就有点高端了，但是下划线大家想必很清楚吧。

CSS 的文本装饰线不仅支持下划线，还支持上划线，还有穿越线。上划线是在文本上方的横线，而穿越线是穿过文字中间的直线。

我们来看下代码：

```
<div>

    <!-- 没有装饰线 -->
    <p style="text-decoration:none;">
        加入生活欺骗了你
    </p>

    <!-- 下划线 -->
    <p style="text-decoration:underline;">
        相信吧，那快乐的日子即将来临
    </p>

    <!-- 上划线 -->
    <p style="text-decoration:overline;">
        不要悲伤，不要心急
    </p>

    <!-- 穿越线 -->
    <p style="text-decoration:line-through;">
        犹豫的日子里需要寂静
    </p>

</div>
```

对应效果如下。



上面的效果中，下划线常用于网站的超级链接中，很多时候超级链接喜欢用下划线标志出来。.

5. 小结

到这里大家可能已经发现了，其实 CSS 语法都是一样的，只是需要记住那些样式的写法，包括名称和值两部分的写法。

或者记在电子文档里面，知道有这么一个样式，当我们需要写的时候能够想起来。

其实写的多了，自然就记住了。

零基础 CSS 入门教程(4)--字体样式

1. 文本与字体区别

上一篇我们学习了设置文本的样式，包括颜色、对齐方式、装饰线，这些实际上都是针对整个文本内容来说的。

而字体这块，则是针对一个个的文字了，主要是设置文字的字体类型、字体风格、字体粗细、字体大小。

2. 字体类型

什么是字体类型呢，就是我们常说的宋体、黑体、楷体、微软雅黑等类型区分，在使用 WORD 编写文档时经常会遇到，如下图：



在网页中，我们同样可以指定字体，通过 `font-family` 样式名来指定即可。windows 系统常见的内置字体有：

宋体 SimSun

黑体 SimHei

微软雅黑 Microsoft YaHei

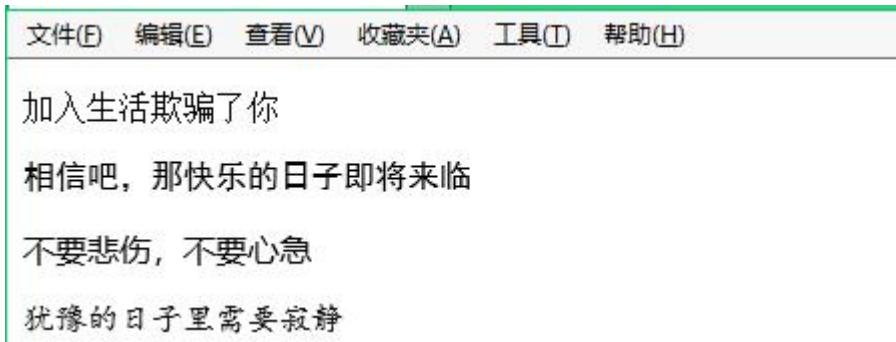
仿宋 FangSong

楷体 KaiTi

我们尝试修改字体，代码如下：

```
<div>
  <!-- 宋体 -->
  <p style="font-family: SimSun;">
    加入生活欺骗了你
  </p>
  <!-- 黑体 -->
  <p style="font-family: SimHei;">
    相信吧，那快乐的日子即将来临
  </p>
  <!-- 微软雅黑 -->
  <p style="font-family: Microsoft YaHei;">
    不要悲伤，不要心急
  </p>
  <!-- 楷体 -->
  <p style="font-family: KaiTi;">
    犹豫的日子里需要寂静
  </p>
</div>
```

对应效果如下，可见不同字体的样式差别还是比较大的。



3. 字体风格

什么是字体风格呢，其实就是一个 `font-style` 样式。

`font-style` 常用取值有 `normal`、`italic` 两种，其实也就是正常字体、斜体两种风格，代码如下：

```
<div>
  <p style="font-style: normal;">
    加入生活欺骗了你
  </p>
  <p style="font-style: italic;">
    相信吧，那快乐的日子即将来临
  </p>
  <p>
    不要悲伤，不要心急
  </p>
  <p>
    犹豫的日子里需要寂静
  </p>
</div>
```

效果如下，斜体有时候会用来突出显示某部分文字。

文件(F) 编辑(E) 查看(V) 收藏夹(A) 工具(I) 帮助(H)

加入生活欺骗了你

相信吧，那快乐的日子即将来临

不要悲伤，不要心急

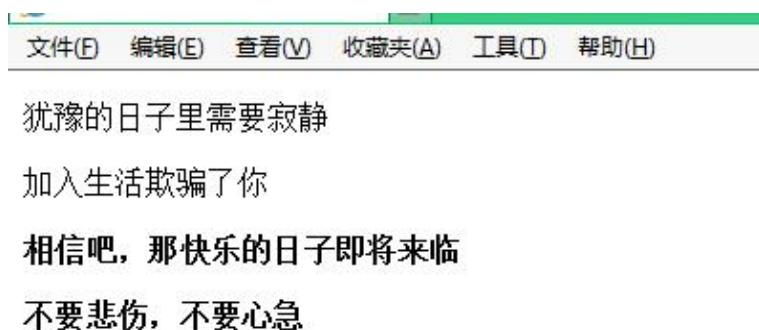
犹豫的日子里需要寂静

4. 字体粗细

这个比较好理解，就是字体的笔画的粗细程度，用 font-weight 描述，代码：

```
<p style="font-weight:lighter;">  
    犹豫的日子里需要寂静  
</p>  
<p style="font-weight:normal;">  
    加入生活欺骗了你  
</p>  
<p style="font-weight:bold;">  
    相信吧，那快乐的日子即将来临  
</p>  
<p style="font-weight:bolder;">  
    不要悲伤，不要心急  
</p>
```

其中，lighter 表示比正常字体细一点，normal 表示正常字体，bold 表示粗体，bolder 表示比粗体更粗一点，效果如下：



是不是发现效果并不算明显，所以实际使用过程中，font-weight 用的也不多。

5. 字体大小

字体大小就用的非常多了，字体大小使用 font-size 描述，而且有好几种描述单位。

5.1 使用像素单位

像素单位是 px，默认是 16px，我们可以在此基础上调大或者调下字体。例如：

```
<div>
  <p style="font-size:12px;">
    这里是 12px 字体
  </p>
  <p style="font-size:16px;">
    这里是 16px 字体
  </p>
  <p style="font-size:20px;">
    这里是 20px 字体
  </p>
  <p style="font-size:24px;">
    这里是 24px 字体
  </p>
</div>
```

效果如下：



5.2 使用 em 单位

em 是相对单位， 也就是说 1em 就是 16px 大小。那么 0.5em 就是 8px 了，2em 就是 32px 大小了，依次类推。代码：

```
<div>
  <p style="font-size:0.5em;">
    这里是 0.5em 字体
  </p>
  <p style="font-size:1em;">
    这里是 1em 字体
  </p>
  <p style="font-size:1.5em;">
    这里是 1.5em 字体
  </p>
  <p style="font-size:2.0em;">
    这里是 2em 字体
  </p>
</div>
```

效果如下：

这是0.5em字体

这里是1em字体

这里是1.5em字体

这里是2em字体

5.3 其他单位

还可以使用%号作为单位，这种用法比较少，此处不再讲解。

6. 小结

CSS 用法比较多，但是大家也无需死记硬背，只要掌握了有哪些样式，脑子中有印象，而且编写过例子实践过。

以后真正写网页用到的时候，参考下之前的例子，或者从网上搜一下，就能写出相应的效果了。

零基础 CSS 入门教程(5)--背景样式

1. 前言

之前讲了文本和字体，用来设置文字相关的样式。

本篇讲述下背景的用法，包括背景色、背景图像、背景图像的平铺方式、背景图像的定位，以及整个背景样式的简写方式。

2. 设置背景色

我们可以通过 `background-color` 给标签设置背景颜色，例如：

```
<div style="background-color: yellow;">  
  <p>  
    春眠不觉晓  
  </p>  
  <p>  
    处处闻啼鸟  
  </p>  
  <p>  
    夜来风雨声  
  </p>  
  <p>  
    花落知多少  
  </p>  
</div>
```

上述代码我们就为 div 区域设置了黄色的背景色，所以效果如下：



3. 设置背景图片

如果觉得背景色不够好看，还可以找一张图片作为背景图，就想我们电脑或者手机的壁纸一般。

我们在 images 文件夹下放置一张 2.jpg，然后通过 background-image 来设置背景图。

```
<div style="background-image:url(images/2.jpg);>  
  <p>  
    春眠不觉晓  
  </p>  
  <p>  
    处处闻啼鸟  
  </p>  
  <p>  
    夜来风雨声  
  </p>  
  <p>  
    花落知多少  
  </p>  
</div>
```

效果如下，可以发现虽然背景图显示出来了，但是重复显示了很多张，并不好看。



4. 背景图平铺

我们可以通过 `background-repeat` 属性来设置背景图的平铺方式。

什么叫平铺呢，就是图像在窗口上是如何铺设的，是横向一张一张的铺，还是不平铺只展示一张图片，还是纵向的一张一张的铺。

4.1 不平铺

可以选择让背景图不平铺，使用 `background-repeat: no-repeat` 即可。

```
<div style="background-image:url(images/2.jpg);background-repeat: no-repeat;">
  <p>
    春眠不觉晓
  </p>
  <p>
    处处闻啼鸟
  </p>
  <p>
    夜来风雨声
  </p>
  <p>
    花落知多少
  </p>
</div>
```

效果如下，此时背景图只显示一张，不会重复。



4.2 水平平铺

使用 `background-repeat:repeat-x;`，可以设置水平平铺，效果如下：



4.3 垂直平铺

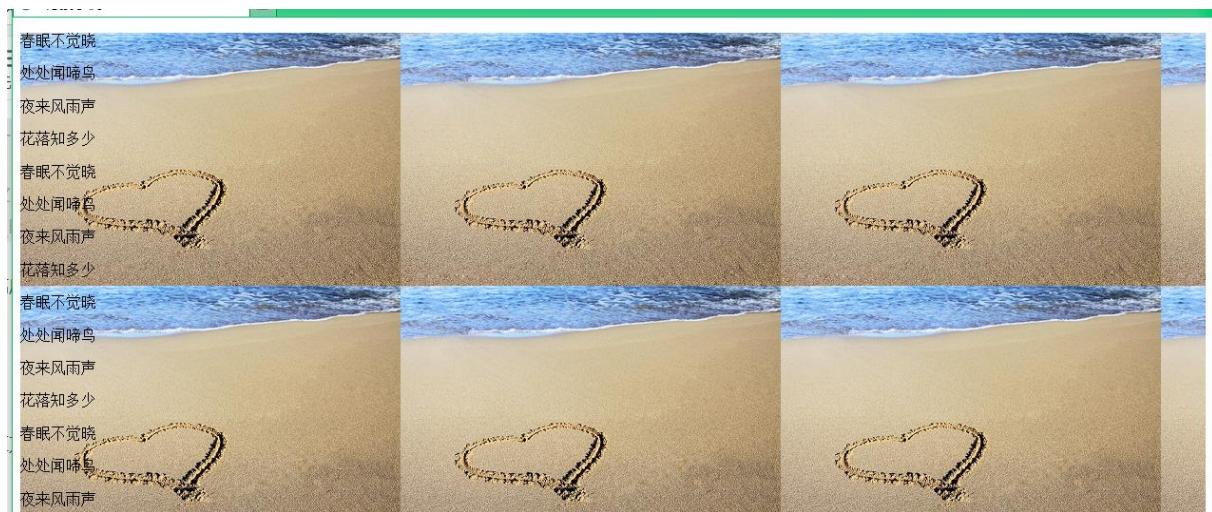
使用 `background-repeat:repeat-y;`，可以设置垂直平铺，效果如下：



注意垂直平铺，得当垂直方向的高度超出背景图高度时，才需要重复，所以我们多些了一些诗句，将高度撑的更高一些。

4.4 两个方向平铺

如果是默认不写，则是水平、垂直方向都会平铺，效果如下：



5. 背景定位

可以使用 `background-position` 改变图像在背景中的位置，例如：

```
<div style="background-image:url(images/2.jpg);background-repeat: no-repeat;background-position: right top;">
  <p>
    春眠不觉晓
  </p>
  <p>
    处处闻啼鸟
  </p>
  <p>
    夜来风雨声
  </p>
  <p>
    </p>
```

```
花落知多少  
</p>  
</div>
```

将图像位置设置为了右侧、上方，所以效果如下：



6. 背景简写

背景相关的样式比较多，如果逐一写下来，还是挺费劲的，所以还有一种简写方法，如下：

```
<div style="background:yellow url('images/2.jpg') no-repeat right top;">  
<p>  
    春眠不觉晓  
</p>  
<p>  
    处处闻啼鸟  
</p>  
<p>  
    夜来风雨声  
</p>  
<p>  
    花落知多少  
</p>  
</div>
```

一个 `background` 样式名，同时设置了背景色、背景图、平铺方式、定位。

这种写法更简单，也更常见！

7. 小结

背景图经常会用在登录界面，选择一张大图片，当做整个界面的背景(为 body 标签设置背景即可)。

大家可以尝试下使用背景图设计一个好看的登录界面。

零基础 CSS 入门教程(6)--元素选择器

1. 前言

在之前学习 CSS 的过程中，大家想必也发现了一些问题。

通过 style 属性编写 CSS 代码，带来的第一个问题，就是代码比较杂乱，HTML 中混杂着 CSS 的部分，看起来就不愉快。

第二个问题，就是每个标签的 CSS 都得手动编写，哪怕有几个标签我们想给他们设置一样的样式，也得完全从头编写，非常繁琐。

本篇就来介绍下 CSS 中的基本选择器，可以非常好的解决此类问题。

而在实际项目开发过程中，CSS 一般也是通过选择器应用到标签元素上的。

2. 什么是选择器

我们知道，一组 CSS 代码，就是一组样式规则，之前我们通过 style 属性，将 CSS 样式规则设置到标签元素上。

选择器就是标签元素的筛选器，同样是 CSS 样式规则，我们可以通过选择器，来选择应用到某个或者某些标签元素上。也就是说，**选择器是用来选择标签元素的。**

3. 使用元素选择器

如果我们想给四个段落设置文本颜色为蓝色，使用 style 属性设置，则代码如下：

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
</head>
<body>
    <p style="color:blue;">黄河远上白云间</p>
    <p style="color:blue;">一片孤城万仞山</p>
    <p style="color:blue;">羌笛何须怨杨柳</p>
    <p style="color:blue;">春风不度玉门关</p>
</body>
</html>
```

而如果我们使用元素选择器，则改为如下操作：

第一步，在 head 区域添加`<style>`标签，head 区域可设置应用于整个网页的规则，例如 charset。此处编写的 style 标签，其内部的规则也是应用于整个网页的。

```
<head>
    <meta charset="utf-8">
    <style>
    </style>
</head>
```

第二步，在`<style>`中编写选择器，也就是说我们想选择哪些 HTML 标签元素，来修改他们的样式。此处我们想修改的是网页上所有 p 元素的样式，所以我们写一个 p，代表我们选择 p 元素。

```
<head>
  <meta charset="utf-8">
  <style>
    p
  </style>
</head>
```

第三步，为选择器添加样式。我们已经通过选择器选择了所有 p 元素，此时可以为其设定样式了，代码如下：

```
<head>
  <meta charset="utf-8">
  <style>
    p {
      color: blue;
    }
  </style>
</head>
```

注意，p 后面的大括号，表示 p 的样式规则集。也就是说大括号内的样式代码，都会应用于 p 这个选择器。大括号内就写 CSS 代码就行了，还是原来 `cssName:cssValue;` 这种格式。

4. 网页整体解析

我们来看下改完之后的网页代码：

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
```

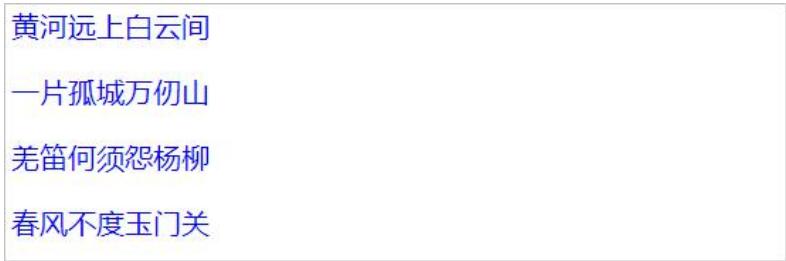
```
<style>
  p {
    color: blue;
  }
</style>

</head>

<body>
  <p>黄河远上白云间</p>
  <p>一片孤城万仞山</p>
  <p>羌笛何须怨杨柳</p>
  <p>春风不度玉门关</p>
</body>
</html>
```

在 head 区域有 style 标签，所以其中的样式规则会应用于整个网页。

p 是元素选择器，所以后面大括号的的样式，会应用于网页上所有的 p 元素，也就是所蓝色这个文本效果，会在所有 p 标签生效，所以效果如下：



黄河远上白云间
一片孤城万仞山
羌笛何须怨杨柳
春风不度玉门关

5. 元素选择器的优点

由上面的例子可见，我们不用元素选择器，也能实现所有 p 标签文本蓝色的效果，那么使用元素选择器后，到底有什么好处呢。

使用元素选择器的好处有三点：

1. 代码变少了。之前要为每个 p 标签编写 CSS 代码，现在只需要在头部写一行。
2. 更好维护了。之前如果要修改颜色，需要把每个 p 标签 style 内的代码进行修改，现在只需要修改头部这一处。
3. 代码更清晰。之前 CSS 和 HTML 代码其实是混在一起的，不好区分。使用选择器后，CSS 代码全部集中于头部，很明显更好区分，更加清晰了。

6. 小结

元素选择器可以一次性将样式应用于某种标签，很好很强大。

零基础 CSS 入门教程(7)--id 选择器

1. 前言

上一篇介绍了元素选择的使用，本篇来介绍一种新的选择器：id 选择器。

我们先来说一下使用场景，如下面的代码中，所有诗句颜色都是蓝色。

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <style>
    p {
      color: blue;
    }
  </style>
</head>
<body>
  <p>黄河远上白云间</p>
  <p>一片孤城万仞山</p>
  <p>羌笛何须怨杨柳</p>
  <p>春风不度玉门关</p>
</body>
</html>
```

此时我们如果觉得第三句“羌笛何须怨杨柳”，是最精彩的一句，我们希望让该句加粗显示，那么如何做到呢。

2. id 选择器

id 选择器，就是应用于选择一个指定元素的。在上面的场景中，我们希望针对性的选择第三个 p 标签，所以我们就可以给第三个 p 标签指定一个 id，然后使用 id 选择器选择这个标签。

那么什么是 id 呢。就好比每个人都有一个唯一的身份证号，每个学生都有唯一的一个学号。id 就是网页标签的唯一标识符，我们可以根据需要给标签添加 id。

3. 使用 id 选择器

接下来，我们使用 id 选择器，给第三句诗设置样式。

首先，给第三句诗所在标签设置 id，如下：

```
<p>黄河远上白云间</p>
<p>一片孤城万仞山</p>
<p id="bolder-line">羌笛何须怨杨柳</p>
<p>春风不度玉门关</p>
```

然后，在 style 区域编写 id 选择器，注意 id 选择器的格式为#xxx，其中#表示按 id 选择，xxx 为被选择的 id，所以此处为：

```
<style>
  p {
    color: blue;
  }

  #bolder-line
</style>
```

最后，为 id 选择器设置样式规则，同样使用大括号包裹：

```
<style>
p {
    color: blue;
}

#bolder-line {
    font-weight: bolder;
}
</style>
```

所以 `font-weight: bolder;` 就被应用到 id 等于 `bolder-line` 的标签上了。

4. 整体网页解析

整体网页代码如下：

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <style>
        p {
            color: blue;
        }

        #bolder-line {
            font-weight: bolder;
        }
    </style>
```

```
</head>
<body>
  <p>黄河远上白云间</p>
  <p>一片孤城万仞山</p>
  <p id="bolder-line">羌笛何须怨杨柳</p>
  <p>春风不度玉门关</p>
</body>
</html>
```

首先是 p 选择器，其样式 `color:blue;` 会应用于所有 p 标签，所以所有段落都是蓝色的。

然后是#bolder-line 选择器，其样式 `font-weight: bolder;` 会应用于 id 等于 bolder-line 的标签，即第三个段落，所以该段落文字加粗。

最终效果如下：



黄河远上白云间
一片孤城万仞山
羌笛何须怨杨柳
春风不度玉门关

5. 小结

元素选择器应用于某类元素的所有标签，id 选择器应用于指定 id 的一个标签。

这两种标签写法的区别是，元素选择器直接写元素名，而 id 选择器写#后面跟着 id。

零基础 CSS 入门教程(8)--class 选择器

1. 前言

前 2 片分别介绍了元素选择器、id 选择器，其实这两种选择器还不是最常用的选择器。

最常用、使用频率最高、最受前端程序员欢迎的选择器，毫无疑问是 class 选择器。

为啥呢？因为元素选择器选择元素过于宽松，而 id 选择器过于严谨，class 选择器面向某一类元素进行选择，实现了灵活度和广泛性的良好结合。

本篇就带大家伙了解下 class 选择器。

2. 场景

考虑考这么一个场景，还是那首《凉州词》，如果我们想实现奇数行蓝色、偶数行绿色，咋处理呢。

此时，如果使用元素选择器，则所有元素只能采用一种颜色。

如果使用 id 选择器，则需要为四个元素分别设置 id，然后为四个 id 分别设置样式。

但是明明只有两种样式啊！

此时就可以适用 class 选择器了。

3. 使用方法

HTML 中专门有一个设置样式的属性，即 class 属性，通过该属性可以设置当前标签应用的样式类。

所以，第一步，我们为四个段落设置两种样式类：

```
<body>
  <p class="text-blue">黄河远上白云间</p>
  <p class="text-green">一片孤城万仞山</p>
  <p class="text-blue">羌笛何须怨杨柳</p>
  <p class="text-green">春风不度玉门关</p>
</body>
```

第二步，在头部 style 标签中设置 class 选择器，注意 class 选择器使用.xxx 格式，.表示此选择器是 class 选择器，xxx 表示选择器应用的标签 class 属性值。

```
<style>
  .text-blue {
  }

  .text-green {
  }
</style>
```

第三步，为 class 选择器设置样式，在大括号内编写选择器对应的样式：

```
<style>
  .text-blue {
    color: blue;
```

```
    }
    .text-green {
        color: green;
    }
</style>
```

通过上面的代码，class 值为 text-blue 的标签，会应用 color:blue 样式；而 class 值为 text-green 的标签，会应用 color:green 样式。

4. 整体解析

看完整代码：

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <style>
        .text-blue {
            color: blue;
        }
        .text-green {
            color: green;
        }
    </style>
</head>
<body>
    <p class="text-blue">黄河远上白云间</p>
    <p class="text-green">一片孤城万仞山</p>
    <p class="text-blue">羌笛何须怨杨柳</p>
    <p class="text-green">春风不度玉门关</p>
```

```
</body>  
</html>
```

首先看头部的 style 标签内，有两个选择器，这两个选择是以.`开头`，所以是 class 选择器。

选择器针对的 class 分别为 `text-blue`、`text-green`，所以网页上 class 值为 `text-blue`、`text-green` 的元素会应用对应大括号内的样式。

5. class 选择器的优点

本篇开头也说了，class 选择器是最常用的选择器，那么它到底有何优点呢。

试想这么一个场景，我们正在开发一个网站，其中按钮的风格肯定是统一的。也就是说，按钮的样式是遵循一个风格，是类似的。

我们通过一系列 CSS 代码定义了一个好看的按钮，就可以通过 class 选择器，将其应用于其他功能类似的按钮。

所以 class 选择器机制，可以让我们自定义一个样式库，然后随时应用于我们想要设置该样式的标签元素上。

甚至别人写好的 class 样式，我们也可以拿过来应用。

这就是说，可以利用经验，加快编程！

6. 小结

class 选择器作为最常用的选择器，在日常 CSS 样式编写的过程中，也是强烈推荐优先使用的！

零基础 CSS 入门教程(9)--注释

1. 前言

之前在讲解 HTML 的时候，已经说过注释这个事情了。理解注释，掌握两点就足够了：

1. 注释对程序代码的运行没有实际作用，也就是说，写了注释，相当于没写。
2. 注释对程序员起到一个提示作用，也就是说，注释可以帮助程序员理解程序。

CSS 的注释，也是这个道理，写了 CSS 注释并不会影响 CSS 的样式效果。

2. CSS 注释的格式

CSS 注释格式如下：

```
/* 注释内容 */
```

所以我们给一段 CSS 代码添加上注释后，代码如下：

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <style>
    /* 蓝色文本 */
    .text-blue {
      color: blue;
    }

    /* 绿色文本 */
    .text-green {
      color: green;
    }
  </style>
```

```
</head>
<body>
  <p class="text-blue">黄河远上白云间</p>
  <p class="text-green">一片孤城万仞山</p>
  <p class="text-blue">羌笛何须怨杨柳</p>
  <p class="text-green">春风不度玉门关</p>
</body>
</html>
```

添加注释后，很明显 CSS 代码部分更加容易读懂了，而且对代码运行效果没有影响。

3. 小结

注释挺重要，养成良好的添加注释的习惯，会大有裨益。

零基础 CSS 入门教程(10)--列表样式

1. 前言

本篇来介绍下列表的 CSS 样式编写，在学习之前，我们先回顾下 HTML 中的有序列表和无序列表。

```
<span>无序列表-我的爱好: </span>
<ul>
    <li>篮球</li>
    <li>足球</li>
    <li>乒乓球</li>
</ul>
<span>有序列表-成绩排行: </span>
<ol>
    <li>张三</li>
    <li>李四</li>
    <li>赵五</li>
</ol>
```

效果如下，无序列表没有顺序之分，有序列表是有先后顺序的。

无序列表-我的爱好:

- 篮球
- 足球
- 乒乓球

有序列表-成绩排行:

1. 张三
2. 李四
3. 赵五

2. 无序列表样式

可以通过 `list-style-type` 修改无序列表的样式，代码如下：

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <style>
        .ul-none {
            list-style-type: none;
        }
        .ul-circle {
            list-style-type: circle;
        }
        .ul-square {
            list-style-type: square;
        }
    </style>
</head>
<body>
    <ul>
        <li>篮球</li>
        <li>足球</li>
        <li>乒乓球</li>
    </ul>
    <ul class="ul-none">
        <li>篮球</li>
        <li>足球</li>
        <li>乒乓球</li>
    </ul>
</body>
```

```
<ul class="ul-circle">
  <li>篮球</li>
  <li>足球</li>
  <li>乒乓球</li>
</ul>
<ul class="ul-square">
  <li>篮球</li>
  <li>足球</li>
  <li>乒乓球</li>
</ul>
</body>
</html>
```

上述四个无序列表，分别通过 class 选择器，设置了默认、无、圆圈、方块四种列表样式，所以效果如下：

- 篮球
- 足球
- 乒乓球

篮球
足球
乒乓球

- 篮球
- 足球
- 乒乓球

- 篮球
- 足球
- 乒乓球

3. 有序列表样式

有序列表可以通过 `list-style-type`，将序号部分修改为英文字母，罗马字母等格式。HTML 部分代码：

```
<ol>
  <li>张三</li>
  <li>李四</li>
  <li>赵五</li>
</ol>
<ol class="ol-lower-alpha">
  <li>张三</li>
  <li>李四</li>
  <li>赵五</li>
</ol>
<ol class="ol-lower-roman">
  <li>张三</li>
  <li>李四</li>
  <li>赵五</li>
</ol>
```

CSS 部分：

```
.ol-lower-alpha {
  list-style-type: lower-alpha;
}

.ol-lower-roman {
  list-style-type: lower-roman;
}
```

上面的 HTML 元素，分别是默认效果、小写英文字母、小写罗马字母样式，所以效果如下：

1. 张三
2. 李四
3. 赵五

a. 张三
b. 李四
c. 赵五

i. 张三
ii. 李四
iii. 赵五

可见效果一般情况，所以平时列表样式用的也不多。

4. 修改列表标记图像

还可以通过 `list-style-image`，将列表左侧的标记修改为一个图像：

```
<ol class="list-image-mark">
  <li>张三</li>
  <li>李四</li>
  <li>赵五</li>
</ol>
```

CSS 代码：

```
.list-image-mark {
  list-style-image: url('logo.jpg');
}
```

注意 `logo.jpg` 要放在当前网页同目录下，便于 CSS 代码去加载图片，上述代码效果如下：



可见为列表使用图像标记，效果还是比较好看的。

5. 小结

列表样式在实际开发过程中，使用的并不算多，但是也是需要大家了解的，一旦遇到相关代码能读懂即可。

零基础 CSS 入门教程(11)--边框样式

1. 前言

边框是使用频率很高的样式，图片可以加边框，表格基本都是带边框的，所以边框的学习是很重要的。

本篇就来介绍下 CSS 边框的用法。

2. 边框的样式

可以通过 border-style 设置边框样式，常用的有 solid 实线、dotted 点线、dashed 虚线三种。

```
<p class="border-solid">  
    演示实线边框  
</p>  
<p class="border-dotted">  
    演示点线边框  
</p>  
<p class="border-dashed">  
    演示虚线边框  
</p>
```

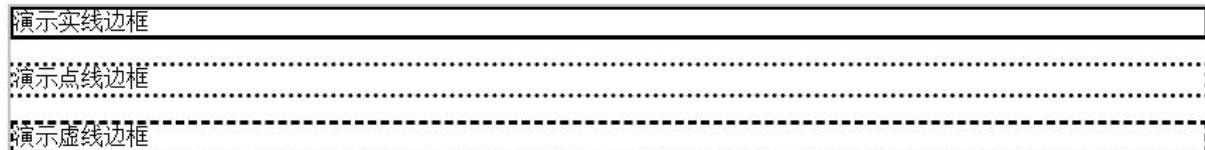
CSS 代码：

```
.border-solid {  
    border-style: solid;  
}
```

```
.border-dotted {  
    border-style: dotted;  
}
```

```
.border-dashed {  
    border-style: dashed;  
}
```

效果如下：



3. 边框宽度

可以通过 border-width 调整边框的宽度，单位一般使用 px 像素。

```
<p class="border-thick">  
    演示粗边框  
</p>  
<p class="border-thin">  
    演示细边框  
</p>
```

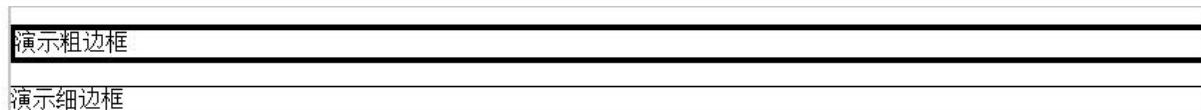
CSS 代码，当日具体的宽度，可以根据实际情况调整，我此处仅仅为了演示。

```
.border-thick {  
    border-style: solid;  
    border-width: 4px;
```

```
}

.border-thin {
    border-style: solid;
    border-width: 1px;
}
```

效果如下：



4. 边框颜色

可以使用 border-color 指定边框的颜色，HTML 代码：

```
<p class="border-red">
    演示红色边框
</p>
<p class="border-green">
    演示绿色边框
</p>
```

CSS 代码：

```
.border-red {
    border-style: solid;
    border-color: red;
}
```

```
.border-green {  
    border-style: solid;  
    border-color: green;  
}
```

效果如下：



5. 简写属性

大家想必发现了，我们在设定边框样式时，一般都会同时设定边框样式、宽度、颜色，如果逐一定义，比较麻烦。

所以 CSS 还提供了简写属性：

```
<p class="border-simple">  
    使用简写属性  
</p>
```

CSS 代码：

```
/* 使用简写属性，边框样式为实线、宽度 1px、颜色蓝色 */  
.border-simple {  
    border: solid 1px blue;  
}
```

效果如下：

使用简写属性

一般情况下，边框都是使用简写属性来编写，因为这样写比较简单。

6. 单独设定各边样式

还支持单独设定上边框、下边框、左边框、右边框的样式：

```
<p class="border-each">  
    单独设定各边框样式  
</p>
```

CSS 代码：

```
.border-each {  
    border-top: solid 1px blue;  
    border-bottom: solid 1px green;  
    border-left: dashed 1px grey;  
    border-right: dashed 1px grey;  
}
```

效果如下：

单独设定各边框样式

7. 小结

边框一般掌握简写方法就够了。

零基础 CSS 入门教程(12)--分组选择器

1. 前言

有时候，我们希望把样式应用到不止一个元素上，如果按照之前的方式，我们只能通过 id 选择器 /class 选择器/元素选择器逐一设置元素的样式。

本篇我们介绍下分组选择器，通过分组选择器，我们可以一次性设置多种选择器对应元素的样式。

2. 不使用分组选择器

考虑如下代码：

```
<body>
  <h1>凉州词</h1>
  <h4>作者：王之涣</h4>
  <p>
    黄河远上白云间
  </p>
  <p>
    一片孤城万仞山
  </p>
  <p>
    羌笛何须怨杨柳
  </p>
  <p>
    春风不度玉门关
  </p>
</body>
```

如果我们希望，标题、作者、诗词都使用蓝色文本，那么 CSS 代码如下：

```
h1 {  
    color: blue;  
}  
  
h4 {  
    color: blue;  
}  
  
p {  
    color: blue;  
}
```

写起来，还是挺麻烦的。

3. 使用分组选择器

如果使用分组选择器，我们可以使用逗号分割开选择器，然后统一应用一些样式代码。

```
h1,h4,p {  
    color: blue;  
}
```

如上代码，表示将`[color: blue;]`同时应用于三个元素选择器，所以效果如下：

凉州词

作者：王之涣

黄河远上白云间

一片孤城万仞山

羌笛何须怨杨柳

春风不度玉门关

4. 混合使用分组选择器

分组选择器可以同时将 id、元素选择器、class 选择器一并分组，例如下面的例子：

```
<head>
  <meta charset="utf-8">
  <style>
    h1,
    h4,
    .blue-text,
    #blue-id {
      color: blue;
    }
  </style>
</head>
<body>
  <h1>凉州词</h1>
  <h4>作者：王之涣</h4>
  <p class="blue-text">
```

```
黄河远上白云间
</p>
<p>
    一片孤城万仞山
</p>
<p id="blue-id">
    羌笛何须怨杨柳
</p>
<p>
    春风不度玉门关
</p>
</body>
```

我们的分组选择器，同时应用于 h1 元素选择器、h4 元素选择器、.blue-textclass 选择器、#blue-id 选择器，所以效果如下：



凉州词

作者: 王之涣

黄河远上白云间

一片孤城万仞山

羌笛何须怨杨柳

春风不度玉门关

5. 小结

通过分组选择器，我们编写 CSS 代码，可以非常灵活的应用于我们指定的多个选择器，这样能够更加的灵活！

零基础 CSS 入门教程(13)--嵌套选择器

1. 前言

上一篇，我们介绍了分组选择器，实际上是可以将样式应用于多个选择器。

本篇介绍另外一种使用频率很高的选择器——嵌套选择器，用来选择选择器内部的选择器。

2. 使用场景

如下代码：

```
<body>
  <div>
    <h1>凉州词</h1>
    <p>
      黄河远上白云间
    </p>
    <p>
      一片孤城万仞山
    </p>
    <p>
      羌笛何须怨杨柳
    </p>
    <p>
      春风不度玉门关
    </p>
  </div>
<div>
```

```
<h1>假如生活欺骗了你</h1>
<p>
    假如生活欺骗了你
</p>
<p>
    不要悲伤，不要心急
</p>
<p>
    忧郁的日子里需要寂静
</p>
<p>
    相信吧，那快乐的日子即将来临
</p>
</div>
</body>
```

上面的代码中，有两首诗词，一首是中国的古典诗，一首诗俄罗斯的现代诗。

如果我们希望中国诗的内容用绿色表示，而俄罗斯诗的内容用黄色表示，利用之前的选择器都是比较难实现的。

但是我们可以发现，中国诗和俄罗斯诗歌分处于不同的 div 下的 p 标签中，所以此时可以利用嵌套选择器。

3. 使用嵌套选择器

代码如下：

```
<head>
    <meta charset="utf-8">
    <style>
```

```
.chinese-poem p {  
    color: green;  
}  
.russia-poem p{  
    color: blue;  
}  
</style>  
</head>  
<body>  
    <div class="chinese-poem">  
        <h1>凉州词</h1>  
        <p>  
            黄河远上白云间  
        </p>  
        <p>  
            一片孤城万仞山  
        </p>  
        <p>  
            羌笛何须怨杨柳  
        </p>  
        <p>  
            春风不度玉门关  
        </p>  
    </div>  
    <div class="russia-poem">  
        <h1>假如生活欺骗了你</h1>  
        <p>  
            假如生活欺骗了你  
        </p>  
        <p>  
            不要悲伤，不要心急  
        </p>
```

```
<p>
    忧郁的日子里需要寂静
</p>
<p>
    相信吧，那快乐的日子即将来临
</p>
</div>
</body>
```

解释下，`.chinese-poem p` 是一个嵌套选择器，中间的空格，表示选择`.chinese-poem` 选择器对应的元素内部的 `p`，所以该嵌套选择器，选中的元素是所有`.chinese-poem` 对应元素内部的 `p` 标签。

同理，`.russia-poem p` 表示选择`.russia-poem` 内部所有的 `p` 标签。

所以，效果如下：

凉州词

黄河远上白云间

一片孤城万仞山

羌笛何须怨杨柳

春风不度玉门关

假如生活欺骗了你

假如生活欺骗了你

不要悲伤，不要心急

忧郁的日子里需要寂静

相信吧，那快乐的日子即将来临

4. 小结

嵌套选择器使用频率非常高，原因是经常需要制定某个区域内部元素的统一样式。

也就是不同区域，应用不同的样式这种情况很常见，而这种情况对应的技术手段，就是嵌套选择器！

零基础 CSS 入门教程(14)--尺寸

1. 前言

尺寸这个词，想必是比较容易理解的，因为买衣服的时候，都得讲究一个尺寸。

具体到网页设计上，尺寸就是指的元素的大小，再说的明白一点，就是指的元素的宽度和高度。

通过 CSS，可以设定元素的宽度、高度，本篇就来具体介绍下。

2. 默认尺寸

为了能体现元素到底占有多大地方，需要给元素设定一个边框，这样才能看到边界。所以代码如下：

```
.box-border {  
    border: 1px solid black;  
}
```

当元素具备 box-border 样式类时，该元素就有边界了，我们能看到它所占的宽度、高度。

接下来，我们看一下，元素默认会占有多大的尺寸：

```
<p class="box-border">  
    默认尺寸  
</p>
```

上述代码，为 p 段落设置了边框，但是并未明确指定宽度、高度，其效果如下：



通过上述效果我们可以看出，默认情况下，元素宽度会占满上级元素的宽度，而元素高度会根据内容的高度决定。

3. 设定元素的宽度和高度

我们可以通过 width/height 设定元素的宽度、高度：

```
<p class="box-border fixed-size">  
    设置宽度和高度  
</p>
```

对应 CSS 代码：

```
.fixed-size {  
    width: 200px;  
    height: 40px;  
}
```

上述代码，明确制订了.fixed-size 选择的元素的宽度为 200px，高度为 40px，所以该区域的高度和宽度是明确确定的，不会随窗口大小变化。



4. 指定最大、最小宽度

可以通过 max-width、min-width 设定元素的宽度范围，此时元素宽度可以变化，但是只能在指定

的范围内变化。如下：

```
<p class="box-border max-min-width">  
    设置最大、最小宽度  
</p>
```

CSS 代码：

```
.max-min-width {  
    max-width: 100px;  
    min-width: 80px;  
}
```

效果如下：



在这个例子中，由于我们设定了最大宽度 100px，而全部文字宽度超出了 100px，所以出现了换行，保证元素占有的宽度最大只能到 100px。同样的，如果窗口缩到很小，该元素也会至少占有 80px 的宽度空间。

5. 指定最大、最小高度

可以通过 max-height/min-height 设置元素的最大、最小高度：

```
<p class="box-border max-min-height">  
    设置最大、最小高度;设置最大、最小高度;设置最大、最小高度;设置最大、  
    最小高度  
</p>
```

CSS 代码：

```
.max-min-height {  
    max-height: 30px;  
    min-height: 10px;  
}
```

对应效果如下：

设置最大、最小高度;设置最大、最小高度;设置最大、最小高度;设置最大、最小高度;设置最大、
最小高度

由于最大高度，已经设置为 30px，所以文字内容再多，也不会改变元素的高度，最终导致文字溢出了 p 元素占有的空间。

6. 小结

一般情况下，大家能理解 max/min/max-width/min-width/max-height/min-height 这些样式代码的含义即可。

入门阶段，能够自己编写 max/min 指定元素大小，就差不多了。

零基础 CSS 入门教程(15)--外边距

1. 前言

要理解外边距，先来说一下外边。

物体的外边，指的就是外围的边界，对网页元素来说，外边就是元素最外面的边界。

那么外边距，就是一个物体的外边，与另一个物体的外边的距离。

CSS 之中，通过 margin 来设定元素的外边距，本篇就来说明下用法。

2. 默认情况

先看代码：

```
<body>
  <input type="button" value="按钮 1">
  <input type="button" value="按钮 2">
  <br>
  <input type="button" value="按钮 3">
  <input type="button" value="按钮 4">
</body>
```

效果如下：



从上述效果可以看出，默认情况下，元素之间的边距很小，排列非常紧凑，这也会导致从界面效果上看密密麻麻，不够美观。

3. 设置外边距

可以通过 margin-left/margin-right;margin-top/margin-bottom 分别设定元素的左、右、上、下边距。

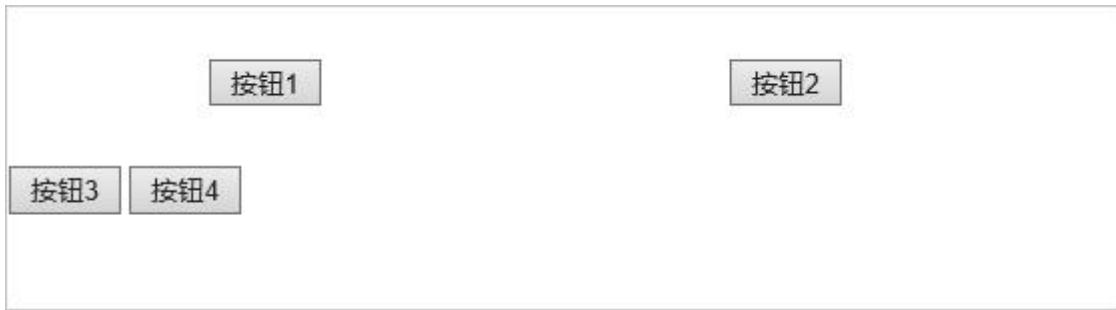
我们为按钮 1 设置一个 id:

```
<body>
  <input type="button" value="按钮 1" id="button1">
  <input type="button" value="按钮 2">
  <br>
  <input type="button" value="按钮 3">
  <input type="button" value="按钮 4">
</body>
```

然后通过 id 选择器选中该元素，为其设置 CSS 样式：

```
#button1 {
  margin-left: 100px;
  margin-right: 200px;
  margin-top: 30px;
  margin-bottom: 30px;
}
```

上述代码，将按钮 1 的左边距设为了 100px，右边距设为了 200px，上下边距设为了 30px，所以效果如下：



我们可以看到，按钮 1 距离左外边距变大了 100px，右外边距变大了 300px，上下外边距变大了 30px。

4. 简写属性

可以通过以下方式，简化外边距的写法：

```
/* 上下左右边距均设置为 10px */
margin: 10px;

/* 上下 10px 左右 20px */
margin: 10px 20px;

/* 上、右、下、左分别为 1px 2px 3px 4px */
margin: 1px 2px 3px 4px;
```

也就是说，当只写一个值时，代表同时为 4 个边设置同样的边距。

当写两个值时，第一个值代表上下边距，第二个值代表左右边距。

当同时写了 4 个值时，分别代表上、右、下、左四个边距，也就是从上开始，顺时针的顺序。

5. 小结

在网页设计开发过程中，外边距的使用相当频繁，所以本节内容还是需要在理解的基础上多加练习。

零基础 CSS 入门教程(16)--内边距

1. 前言

上一篇介绍了外边距，也就是元素跟相邻元素的距离。

本篇来介绍内边距，顾名思义，内边距是指的元素内部的内容，与元素的边的距离。

CSS 通过 padding 来控制内边距。

2. 默认情况

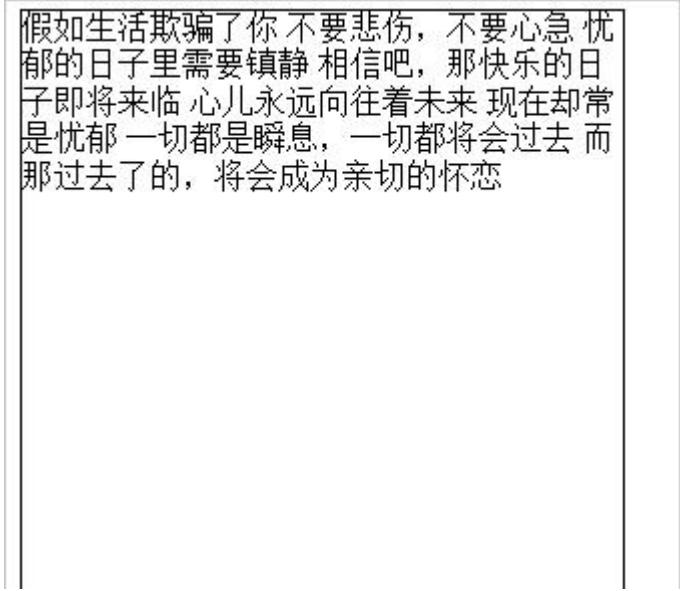
看如下代码：

```
<body>
  <div class="box">
    假如生活欺骗了你
    不要悲伤，不要心急
    忧郁的日子里需要镇静
    相信吧，那快乐的日子即将来临
    心儿永远向往着未来
    现在却常是忧郁
    一切都是瞬息，一切都将会过去
    而那过去了的，将会成为亲切的怀恋
  </div>
</body>
```

我们为 box 类设置宽度、高度，并添加一个边框：

```
.box {  
    width: 300px;  
    height: 300px;  
    border: 1px solid black;  
}
```

效果如下：



假如生活欺骗了你 不要悲伤，不要心急 忧郁的日子里需要镇静 相信吧，那快乐的日子即将来临 心儿永远向往着未来 现在却常是忧郁 一切都是瞬息，一切都将会过去 而那过去了的，将会成为亲切的怀恋

我们可以看到，默认情况下，元素内部的内容是紧贴着元素的边的，没有浪费空间。

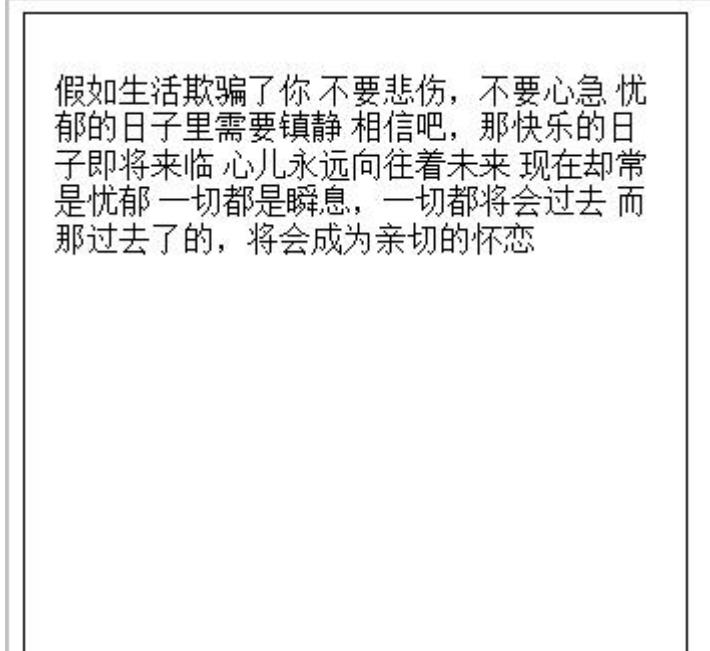
3. 设置内边距

我们可以通过 padding-top、padding-right、padding-bottom、padding-left、，分别设置元素的上、右、下、做外边距。

例如我们可以为 box 类添加如下样式代码，调整上方和左右两侧的内边距：

```
.box {  
    width: 300px;  
    height: 300px;  
    border: 1px solid black;  
    padding-top: 30px;  
    padding-left: 15px;  
    padding-right: 15px;  
}
```

通过上面的代码，我们将上边距设为了 30px，而左右边距设为了 15px，所以效果如下：



假如生活欺骗了你 不要悲伤，不要心急 忧郁的日子里需要镇静 相信吧，那快乐的日子即将来临 心儿永远向往着未来 现在却常是忧郁 一切都是瞬息，一切都将会过去 而那过去了的，将会成为亲切的怀恋

4. 简写属性

与外边距 margin 同样，padding 内边距也支持简写属性：

```
/* 上下左右边距均设置为 10px */  
padding: 10px;
```

```
/* 上下 10px 左右 20px */
padding: 10px 20px;
/* 上、右、下、左分别为 1px 2px 3px 4px */
padding: 1px 2px 3px 4px;
```

也就是说，当只写一个值时，代表同时为 4 个边设置同样的内边距。

当写两个值时，第一个值代表上下内边距，第二个值代表左右内边距。

当同时写了 4 个值时，分别代表上、右、下、左四个内边距，也就是从上开始，顺时针的顺序。

5. 小结

内边距、外边距的使用都很频繁，因为我们要控制距离，才能让网页显得错落有致。

零基础 CSS 入门教程(17)--表格样式

1. 前言

本篇来介绍下，表格常用的那些 CSS 样式编写方法，使用的 CSS 代码其实之前基本都已经接触过了，本篇也算是对之前的 CSS 章节的应用演示。

2. HTML 表格边框

在 HTML 教程部分，我们曾经使用 border 属性设定表格，代码如下：

```
<table border="1">
  <thead>
    <th>姓名</th>
    <th>性别</th>
  </thead>
  <tbody>
    <tr>
      <td>张三</td>
      <td>男</td>
    </tr>
    <tr>
      <td>李四</td>
      <td>女</td>
    </tr>
  </tbody>
</table>
```

其实这种方式，是不推荐使用的。因为网页设计遵循的原则，还是 HTML 表示内容，而 CSS 进行样式描述。

对于表格来说，表格的内容就是行列里的文字，而边框属于对表格外观的美化，属于样式部分，所以更适合用 CSS 描述。

3. CSS 设定表格边框

之前我们已经学习过 CSS 之边框的设定方法了，所以可以利用 CSS 的 border 来为表格设定边框，代码如下：

设定表格边框：

```
<table class="table-border">  
  <thead>  
    <th>姓名</th>  
    <th>性别</th>  
  </thead>  
  <tbody>  
    <tr>  
      <td>张三</td>  
      <td>男</td>  
    </tr>  
    <tr>  
      <td>李四</td>  
      <td>女</td>  
    </tr>  
  </tbody>  
</table>
```

对应 CSS 代码：

```
.table-border {  
  border: 1px solid black;  
}
```

我们为表格设定了宽度 1px、实线、黑色的边框，所以效果如下：

设定表格边框：	
姓名	性别
张三	男
李四	女

4. 设定列边框

在上面的例子中，表格外围是有了边框了，但是各列之间并未有边框区分。实际大多数情况下，我们希望列也有表格边框，可以如下设定：

同时设定列边框：

```
<table class="table-col-border">  
  <thead>  
    <th>姓名</th>  
    <th>性别</th>  
  </thead>  
  <tbody>  
    <tr>  
      <td>张三</td>  
      <td>男</td>  
    </tr>  
    <tr>  
      <td>李四</td>  
      <td>女</td>  
    </tr>  
  </tbody>  
</table>
```

CSS 代码：

```
.table-col-border th,  
.table-col-border td {  
    border: 1px solid black;  
}
```

我们利用嵌套选择器、分组选择器，设定了.table-col-border 内部 th 和 td 元素的 CSS 样式，所以效果如下：

同时设定列边框：

姓名	性别
张三	男
李四	女

5. 折叠边框

在上图中，表格具备双边边框，看起来奇奇怪怪的。这是因为 th、td 标签设置了独立的边框。

如果想让边框折叠为一条，可以使用 border-collapse 属性实现：

折叠边框：

```
<table class="table-collapse">  
    <thead>  
        <th>姓名</th>  
        <th>性别</th>  
    </thead>  
    <tbody>  
        <tr>  
            <td>张三</td>
```

```
<td>男</td>
</tr>
<tr>
    <td>李四</td>
    <td>女</td>
</tr>
</tbody>
</table>
```

CSS 代码：

```
.table-collapse {
    border-collapse: collapse;
}
.table-collapse th,
.table-collapse td {
    border: 1px solid black;
}
```

在上面的代码中，我们对表格应用了 `border-collapse: collapse;` 样式代码，所以表格被折叠，效果如下：

折叠边框：

姓名	性别
张三	男
李四	女

6. 表格尺寸

再看上面的表格，尺寸太小了，看着就憋屈。我们可以使用之前学习的 width、height 修改表格的尺寸。

表格尺寸：

```
<table class="table-size">
  <thead>
    <th>姓名</th>
    <th>性别</th>
  </thead>
  <tbody>
    <tr>
      <td>张三</td>
      <td>男</td>
    </tr>
    <tr>
      <td>李四</td>
      <td>女</td>
    </tr>
  </tbody>
</table>
```

CSS 代码：

```
.table-size {
  border-collapse: collapse;
  width: 400px;
  height: 100px;
}
```

```
.table-size th,  
.table-size td {  
    border: 1px solid black;  
}
```

指定高度、宽度后效果如下：

表格尺寸：

姓名	性别
张三	男
李四	女

7. 表格内边距

表格尺寸增大后，看起来舒服多了。但是行列的内容与边框靠的太近，还是不够美观，此时我们可以利用 padding 设定表格的内边距。

表格的内边距：

```
<table class="table-padding">  
    <thead>  
        <th>姓名</th>  
        <th>性别</th>  
    </thead>  
    <tbody>  
        <tr>  
            <td>张三</td>  
            <td>男</td>  
        </tr>  
        <tr>  
            <td>李四</td>
```

```
<td>女</td>
</tr>
</tbody>
</table>
```

CSS 代码：

```
.table-padding {
    border-collapse: collapse;
    width: 400px;
    height: 100px;
}

.table-padding th,
.table-padding td {
    width: 200px;
    border: 1px solid black;
    padding: 10px;
}
```

在上面的代码中，我们将 th、td 的内边距设为 10 个像素，所以表格行列里面的文字与行列边框之间，会有 10px 的距离，效果如下：

表格的内边距：

姓名	性别
张三	男
李四	女

8. 表格背景色

通常情况下，我们希望让表格的标题栏具备一个特殊的背景色，这样看起来会区分度更强。

表格背景色：

```
<table class="table-back">  
  <thead>  
    <th>姓名</th>  
    <th>性别</th>  
  </thead>  
  <tbody>  
    <tr>  
      <td>张三</td>  
      <td>男</td>  
    </tr>  
    <tr>  
      <td>李四</td>  
      <td>女</td>  
    </tr>  
  </tbody>  
</table>
```

CSS 代码：

```
.table-back {  
  border-collapse: collapse;  
  width: 400px;  
  height: 100px;  
}  
.table-back th,  
.table-back td {  
  width: 200px;  
  border: 1px solid black;  
  padding: 10px;  
}
```

```
.table-back th {  
    background-color: green;  
    color: white;  
}
```

我们为表格的标题栏，设定了绿色背景，同时将文字调为白色，所以效果如下：

表格背景色：

姓名	性别
张三	男
李四	女

此时表格已经比较美观了，大功告成！

9. 小结

本篇介绍了表格基本的、常用的样式描述方法。

设计使用过程中，可以根据网站整体风格，设计表格的具体样式。

零基础 CSS 入门教程(18)--颜色值

1. 前言

世界是五彩缤纷的，不论是皑皑的白雪、还是飘落的秋叶，世界因色彩而绚丽。

同样，网页也因为合理搭配色彩，而美观。

之前我们已经学习过设置文本的颜色，设置背景色，以及设置边框的颜色。

颜色的取值都是：red、blue、green 等代表颜色的单词。

但是，代表颜色的单词是十分有限的，而世界上的色彩千千万万，我们该如何使用 CSS 来代表特定的颜色值呢？

2. 三原色

实际上，在光学中，红、绿、蓝三种颜色被称为三原色，也就是说这三种光学颜色混合后，可以产生各种各样的颜色。

基于三原色的原理，我们也可以通过设置红、绿、蓝颜色的比例，来确定一个特定的颜色。

3. RGB 颜色值

我们可以通过 RGB，R 代表红色，G 代表绿色，B 代表蓝色，来确定颜色值。每种颜色取值范围是 0 -255,0 就表示这种颜色没啥影响力，而 255 表示把这种颜色用到最大力量了。

形如：

```
<head>
  <meta charset="utf-8">
  <style>
    .rgb-black {
      color: rgb(255, 0, 0);
    }
    .rgb-green {
      color: rgb(0, 255, 0);
    }
    .rgb-mix {
      color: rgb(120, 200, 0);
    }
  </style>
</head>
<body>
  <div class="rgb-black">
    rgb(0, 0, 0)的颜色
  </div>
  <div class="rgb-green">
    rgb(0, 255, 0)的颜色
  </div>
  <div class="rgb-mix">
    rgb(120, 200, 0)的颜色
  </div>
</body>
```

在上面的例子中，`rgb-black` 是红色部分 255，绿色蓝色部分 0，所以红色是最大值，绿色蓝色的部分则一点没有所以呈现红色。

rgb-green 的是绿色部分是最大值，红蓝没有，所以呈现绿色。

rgb-mix，红色 120 比较小，而绿色 200 比较多，所以呈现一种绿色较多红色有一点影响的整体效果。

最终效果如下：

rgb(0, 0, 0)的颜色
rgb(0, 255, 0)的颜色
rgb(120, 120, 0)的颜色

4. 16 进制颜色值

上面我们讲了，每种颜色取值是 0-255，正好可以用一个两位的 16 进制数表示，所以用一个六位的 16 进制数，就能表达红、黄、蓝三种颜色的取值。

例如：

```
<div class="hex-red">  
    #FF0000 的颜色  
</div>
```

CSS 代码：

```
.hex-red{  
    color: #FF0000;  
}
```

颜色取值为 #FF0000，其中#表示这是 16 进制颜色值，FF 代表红色，00 代表黄色，00 代表蓝色。FF 对应的 10 进制数即为 255，所以呈现为红色。

16 进制数如果大家不理解的话，也没关系，大家只要注意这种写法#000000，代表一种颜色就行了。

5. 取色

我们知道颜色值是如何表示了，但是我们也没法看到一种好看的颜色，就知道它的颜色值。

如果我们想要模仿一个网站中的好看的颜色，我们可以借助取色工具，例如 TakeColor，取色效果如下：



这种软件，可以显示我们选中的颜色对应的颜色值，我们拿到颜色值后，就可以在网页中使用该颜色了。

6. 小结

CSS 支持三种设置颜色值的方式：

1. 直接使用英文单词代表的颜色
2. 使用 RGB 表示颜色值
3. 使用 16 进制表示颜色值

零基础 CSS 入门教程(19)--盒子模型

1. 前言

之前我们讲过 CSS 的尺寸、边框、内边距、外边距。

其实这些都是占用网页空间，在 CSS 中，我们将网页元素占用的空间形容为一个盒子。

盒子模型是比较复杂的，在入门阶段大家做一个基础的理解就可以了。

2. 实例解析

我们直接通过一个实例来了解下盒子模型：

```
<head>
  <meta charset="utf-8">
  <style>
    .box {
      margin: 50px;
      border: 10px solid blue;
      padding: 20px;
      background-color: green;
    }
    .content {
      background-color: yellow;
    }
  </style>
</head>
<body>
```

```
<div class="box">
  <div class="content">
    黄河远上白云间<br>
    一片孤城万仞山<br>
    羌笛何须怨杨柳<br>
    春风不度玉门关<br>
  </div>
</div>
<hr>
</body>
```

效果如下：



我们具体介绍下，蓝色边框包围的部分，就是我们的盒子，我们使用.box 类描述这个盒子。

由于外边距设置为了 50px，所以这个盒子与上下左右的边距是 50 像素。

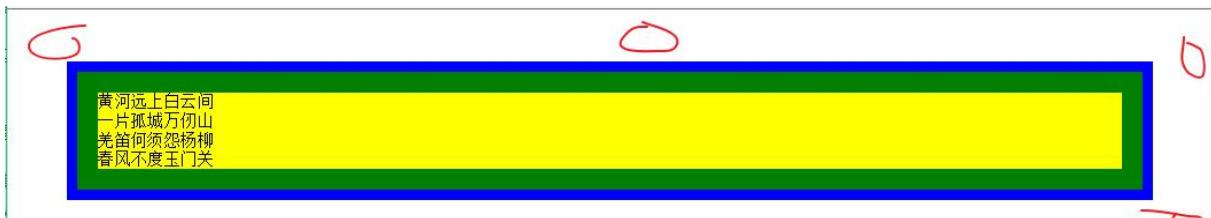
由于边框设置为了 10px，所以蓝色边框上下左右部分，宽度都是 10px。

最后由于内边距设置为了 20px，所以盒子内部内容和盒子边框的距离，上下左右都是 20px。

最后，里面是元素内容区域，我们指定其为黄色背景。

这就是盒子模型，一个元素占用的空间，其实是包含内容部分、内边距部分、边框部分、以及最容易遗漏的外边距部分。

看下图，各个红圈圈出来的部分，是 box 元素的外边距部分，也是被占用的空间（填不上其他东西，必须留出这些空间）。



3. 小结

大家一定要理解，一个元素占用的空间，其实是包含很多部分，不只是内容部分。

如果要精确描述的话：

元素的宽度=宽度(width)+左填充(padding-left)+右填充(padding-right)+左边框(border-left)+右边框(border-right)+左边距 margin-left)+右边距 margin-right)。

元素的高度=高度+上填充+下填充+上边框+下边框+上边距+下边距。

这就是盒子模型，大家记住盒子占用的空间，包含很多部分就行。

零基础 CSS 入门教程(20)--显示

1. 前言

在 HTML 学习中，我们曾经讲过一个重要概念：块级元素 div、行级元素 span。它们的区别是：块级元素独占一块区域，所以后面换行；而行级元素只会占用当前行，所以后面不换行。

在上一篇文章中，我们又接触了 CSS 盒子模型，我们进一步了解了，在网页上，所有元素其实相当于一个盒子，会占据一定的空间。那么块级元素、行级元素，其实说的就是元素占用空间的方式，是占用一块，还是占用一行的区别。

元素占用空间的方式，用比较专业的名词来讲，就是**元素的盒类型**，或者说**显示类型**，仔细品味的话，就能发现这三种说法是一个意思。

实际上，在 CSS 中，我们可以通过 display 属性，指定元素的显示类型，直接让元素按我们想要的方式显示。

2. 块级元素 block

block 表示将元素设定为块级元素，也就是说单独的占用一块，这一块跟前后的元素在垂直方向是隔离的，相当于前后都有换行符。

看如下代码：

```
<head>
  <meta charset="utf-8">
  <style>
    .display-block {
      display: block;
```

```
background-color: grey;  
}  
</style>  
</head>  
  
<body>  
  <span>春眠不觉晓</span>  
  <span class="display-block">处处闻啼鸟</span>  
  <span>夜来风雨声</span>  
  <span>花落知多少</span>  
</body>
```

正常情况下，span 是行级元素，前后不会换行。但是通过 `display: block;` 将其盒类型修改为了块级元素，所以效果如下：



3. 行级元素 inline

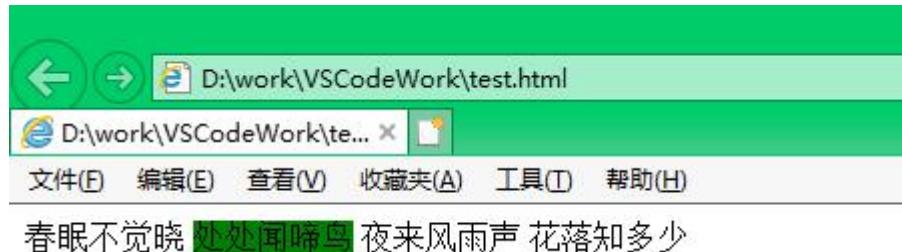
inline 表示将元素设置为行内元素，其实就是不换行，跟旁边的元素一个样子。

看如下例子：

```
<head>  
  <meta charset="utf-8">  
  <style>
```

```
.display-inline {  
    display: inline;  
    background-color: grey;  
}  
</style>  
</head>  
  
<body>  
    <span>春眠不觉晓</span>  
    <div class="display-inline">处处闻啼鸟</div>  
    <span>夜来风雨声</span>  
    <span>花落知多少</span>  
</body>
```

原本 div 是要前后换行的，此时将 display 设置为 inline，所以 div 区域变为普通的行级区域，不再换行，故效果如下：



4. 行内块级元素 inline-block

inline-block 就比较稀奇了，中文名称为行内-块级元素。

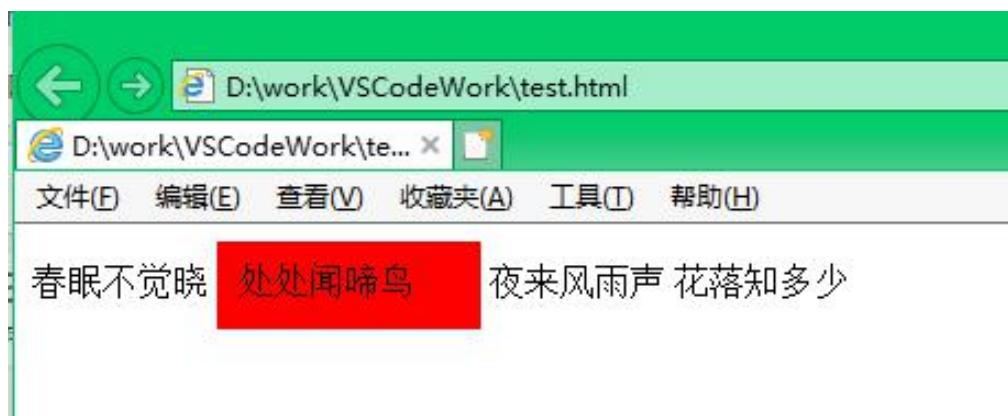
具体来说，inline-block 元素在整体上是一个行级元素，前后并不会换行。但是在盒子内部，却表现为一个块级元素，所以长度、高度、边距等属性可以应用到该元素上。

例如：

```
<head>
  <meta charset="utf-8">
  <style>
    .display-inline-block {
      display: inline-block;
      background-color: red;
      width: 100px;
      height: 20px;
      padding: 10px;
    }
  </style>
</head>

<body>
  <span>春眠不觉晓</span>
  <div class="display-inline-block">处处闻啼鸟</div>
  <span>夜来风雨声</span>
  <span>花落知多少</span>
</body>
```

前后并不换行，但是在行内是作为一个块的，效果如下：



5. 不显示 none

none 比较简单了，设置为 none 后元素没有盒子了，所以元素不在页面中占据空间，也不再显示。

例如：

```
<head>
  <meta charset="utf-8">
  <style>
    .display-none {
      display: none;
      background-color: grey;
    }
  </style>
</head>

<body>
  <span>春眠不觉晓</span>
  <span class="display-none">处处闻啼鸟</span>
  <span>夜来风雨声</span>
  <span>花落知多少</span>
</body>
```

效果如下：



6. 小结

display 属性的作用是设定元素的盒子类型，也就是调整元素在网页上占据的空间类型。

零基础 CSS 入门教程(21)--透明

1. 前言

前面讲的概念，算是比较抽象比较难一些了，本篇介绍的透明就非常好理解。CSS 可以通过 opacity 属性设置元素的透明程度，取值范围 0-1，当取值为 1 时是完全不透明，而取值为 0 是完全透明。

2. 实例解析

看如下代码：

```
<head>
  <meta charset="utf-8">
  <style>
    .opacity-none {
      opacity: 1;
    }
    .opacity-low {
      opacity: 0.7;
    }
    .opacity-high {
      opacity: 0.3;
    }
    .opacity-full {
      opacity: 0;
    }
  </style>
</head>
<body>
```

不透明效果:

```

```

低透明效果:

```

```

高透明效果:

```

```

完全透明效果:

```

```

```
</body>
```

opacity 取值越小，透明度越高，所以效果如下：



3. 小结

通过 opacity 属性可以轻松的设置元素的透明度。

零基础 CSS 入门教程(22)--溢出

1. 前言

之前我们讲过了，网页元素会占据一个盒子的空间。如果我们通过 width 或者 height 设置了元素的尺寸，那么就确定了元素内容的大小。

如果此时，元素内容太多，超出了我们设置的尺寸，会是什么情况呢。

这就是本节课介绍的溢出，指的就是内容太多，超出元素尺寸的情况。

CSS 提供了 overflow 属性设置溢出时元素的显示方式。

2. 默认显示溢出部分

如果我们不指定 overflow，则 overflow 默认取值为 visible，意思是溢出部分显示出来。例如：

```
<div class="box overflow-visible">
  <h3>凉州词</h3>
  黄河远上白云间<br>
  一片孤城万仞山<br>
  羌笛何须怨杨柳<br>
  春风不度玉门关<br>
</div>
```

对应 CSS：

```
.box {
  border: 1px solid blue;
```

```
height: 100px;  
}  
.overflow-visible {  
    overflow: visible;  
}
```

我们通过`.box` 类为元素设置了边框和高度，通过`.overflow-visible` 将元素溢出方式设置为 `visible`，注意默认情况也是该溢出方式。此时效果如下：



可见溢出的部分也显示了出来。

3. 隐藏溢出部分

通过将 `overflow` 属性值设置为 `hidden`，可以隐藏溢出部分。

```
<div class="box overflow-hidden">  
    <h3>凉州词</h3>  
    黄河远上白云间<br>  
    一片孤城万仞山<br>  
    羌笛何须怨杨柳<br>  
    春风不度玉门关<br>  
</div>
```

CSS:

```
.overflow-hidden {  
    overflow: hidden;  
}
```

效果如下：



凉州词
黄河远上白云间
一片孤城万仞山

4. 显示滚动条

上面两种方式的处理，明显都不够好。第一种内容都跑出来了，第二种直接不让你看溢出的内容了。

我们可以通过将 overflow 属性值设置为 scroll，从而显示滚动条以便查看溢出内容。

```
<div class="box overflow-scroll">  
    <h3>凉州词</h3>  
    黄河远上白云间<br>  
    一片孤城万仞山<br>  
    羌笛何须怨杨柳<br>  
    春风不度玉门关<br>  
</div>
```

CSS 代码：

```
.overflow-scroll {  
    overflow: scroll;  
}
```

此时效果如下，拖动右侧滚动条即可查看溢出内容。



5. 自动处理

在上面的例子中，内容的宽度并未超过元素宽度，其实没必要显示横向滚动条。此时可以直接将 overflow 设置为 auto，让浏览器自动处理去。

```
<div class="box overflow-auto">
  <h3>凉州词</h3>
  黄河远上白云间<br>
  一片孤城万仞山<br>
  羌笛何须怨杨柳<br>
  春风不度玉门关<br>
</div>
```

CSS 代码：

```
.overflow-auto {
  overflow: auto;
}
```

在 IE 浏览器下，处理结果如下：



使用 Chrome 浏览器看看，效果如下：



这两种浏览器都足够智能，只显示了垂直滚动条，但是从样式上看，好像 Chrome 浏览器更好看一点。

6. 小结

综合上面的实例，设置 auto 效果最好。

零基础 CSS 入门教程(23)--伪类

1. 前言

我们之前讲过了许多种选择器，比如元素选择器、id 选择器、class 选择器、分组选择器、嵌套选择器。

这些选择器，都是用来选择某个、或者某些元素的。

但是其实还有一些更高级的选择方式，例如选择处于某种状态的元素。比如超级链接，有未点击过的、有已点击过的、有鼠标悬停的、也有点击那一刻的。

我们可以通过伪类选择器，来选择这些特殊的状态的元素。至于为啥叫伪类，可能跟它们不是正儿八经的类，只是处于某种状态的类有关系。

2. 实例解析

我们以超级链接元素常用的伪类选择器为例进行演示：

```
<head>
  <meta charset="utf-8">
  <style>
    /* 未点击过的链接 */
    a:link {
      color: grey;
    }
    /* 已点击过的链接 */
    a:visited {
      color: red;
    }
```

```
}

/* 鼠标悬停的链接 */
a:hover {
    color: blue;
}

/* 点击那一刻的链接 */
a:active {
    color: green;
}

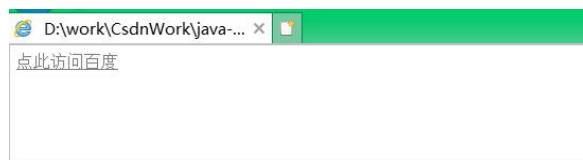
</style>

</head>

<body>
    <a href="http://www.baidu.com">点此访问百度</a>
</body>
```

我们通过伪类选择器，设定了不同状态下超级链接的样式。

所以未点击时，效果如下：



当鼠标悬停时，颜色会变为蓝色；当点击那一刻，颜色会变为绿色。再次返回该网页，因为已经点击过，颜色会变为红色。大家可以自己尝试下。

3. 小结

伪类选择器拓展了 CSS 选择器的功能，让我们可以把样式应用的更灵活更方便。

实际在项目开发过程中，大家可以接触到更多的伪类选择器。

零基础 CSS 入门教程(24)--外联选择器

1. 前言

本篇比较轻松，没有具体的 CSS 样式语法，只是介绍下 CSS 代码的存放位置。来吧，跟我了解下。

2. 内联样式

内联样式，通过元素的 style 属性，直接在元素上编写：

```
<!-- 内联样式 -->
<div style="color:blue;font-size:2em;">
    努力努力再努力!
</div>
```

这种写法的缺点是，样式只能应用于一个元素，如果想批量应用到元素上，就得一个一个去写，比较麻烦。

3. 内部样式表

我们可以将样式代码，集中到网页头部，这种方式就叫做内部样式表。

```
<head>
    <meta charset="utf-8">
    <!-- 此处即为内部样式表 -->
    <style>
        .my-text {
```

```
        color: red;  
        font-size: 1.5em;  
    }  
</style>  
</head>  
<body>  
    <!-- 内联样式 -->  
    <div style="color:blue;font-size:2em;">  
        努力努力再努力!  
    </div>  
    <!-- 使用内部样式表 -->  
    <div class="my-text">  
        加油加油再加油!  
    </div>  
</body>
```

使用内部样式表后，编写的样式可以应用到多个元素上，还有一个很大的好处：HTML 代码和 CSS 代码实现了分离，程序结构更加清晰明了，这个也是很重要的。

4. 外部样式表

如果我们的网站规模很大，有上百个网页，但是其实这些页面会共享一些样式。

此时我们可以将样式代码写到一个单独的文件中，然后在各个页面引用这个文件中的样式即可。

例如，我们编写 style.css 文件：

```
/* style.css 文件 */  
.my-div {  
    border: 1px solid grey;  
    color: green;
```

```
font-size: 2em;  
}
```

在 html 中引用该文件：

```
<!-- 引用外部样式表 -->  
<link rel="stylesheet" type="text/css" href="style.css">
```

然后代码中就可以使用该样式类了。

5. 完整实例

我们通过一个实例看下完整代码：

style.css 文件代码：

```
/* style.css 文件 */  
.my-div {  
    border: 1px solid grey;  
    color: green;  
    font-size: 2em;  
}
```

css-style-type.html 文件代码：

```
<!DOCTYPE html>  
<html>  
<head>  
    <meta charset="utf-8">  
    <!-- 此处即为内部样式表 -->
```

```
<style>
    .my-text {
        color: red;
        font-size: 1.5em;
    }
</style>
<!-- 引用外部样式表 --&gt;
&lt;link rel="stylesheet" type="text/css" href="style.css"&gt;
&lt;/head&gt;
&lt;body&gt;
    <!-- 内联样式 --&gt;
    &lt;div style="color:blue;font-size:2em;"&gt;
        努力努力再努力!
    &lt;/div&gt;
    <!-- 使用内部样式表 --&gt;
    &lt;div class="my-text"&gt;
        加油加油再加油!
    &lt;/div&gt;
    <!-- 使用外部样式表 --&gt;
    &lt;div class="my-div"&gt;
        好好学习、天天向上!
    &lt;/div&gt;
&lt;/body&gt;
&lt;/html&gt;</pre>
```

效果如下，可见三种样式都应用成功了。



6. 小结

在学习过程中，为了方便，大家可能更习惯编写内联样式，或者使用内部样式表。

实际项目开发过程中，使用外部样式表的情况更多，因为可以更好的复用 CSS 代码。

零基础 CSS 入门教程(25)--选择器优先级

1. 前言

在上一篇文章中，我们介绍了内联样式、内部样式表、外部样式表，这三种样式都会影响到网页上的元素。

那么如果三种方式同时作用于同一个元素，元素该按什么规则显示呢。

这就涉及到 CSS 选择器优先级问题，本篇就此进行介绍。

2. 就近原则

网页元素在应用样式代码时，遵循就近原则，也是就是，谁跟网页元素关系近，就用谁的规则。

所以，内联样式直接就在元素标签上，优先级最高；内部样式表就在本网页内，优先级次之；外部样式表单独文件存储，优先级最低。

3. 实例验证

我们通过一个实例来验证下：

首先编写外部样式文件 style.css，代码如下：

```
.out-style {  
    color: red;  
}
```

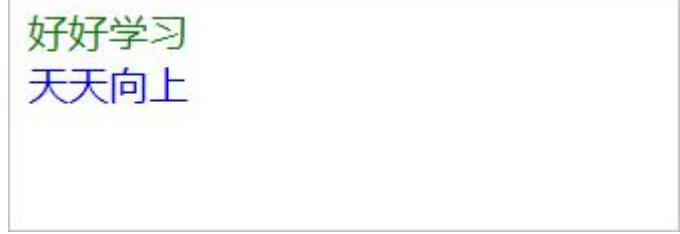
然后编写网页，并引用外部样式文件，且定义内部样式表：

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <!-- 引用外部样式表 -->
  <link rel="stylesheet" type="text/css" href="style.css">
  <style>
    .in-style {
      color: green;
    }
  </style>
</head>
<body>
  <div class="out-style in-style">
    好好学习
  </div>
  <div class="out-style in-style" style="color:blue">
    天天向上
  </div>
</body>
</html>
```

对于“好好学习”部分，同时应用了外部样式表的 out-style 和内部样式表的 in-style，因为内部样式表优先级高，所以会显示绿色字体。

对于“天天向上”部分，同时应用了外部样式表、内部样式表的样式，而且还编写了内联的样式，因为内联的优先级最高，所以会显示蓝色字体。

最终效果如下：



好好学习
天天向上

4. 小结

当我们引用别人编写的样式文件后，如果对效果不满意，可以利用优先级规则，编写我们自己想要的 CSS 代码。

如果我们自己编写的 CSS 代码优先级更高，就能覆盖引用样式文件的效果，从而实现我们自己的目标。

零基础 CSS 入门教程(26)--按钮实例

1. 前言

通过前面的学习，我们已经基本掌握了 CSS 常用的知识点。当然，不是全部的知识点都讲解了，其他的知识点大家可以再以后的学习实践过程中，逐步掌握。

从本篇开始，我们会讲解几个 CSS 的应用实例，让大家更好的把之前所学的知识融会贯通。对于计算机编程这样的应用型学科来说，学以致用是我们的目标，在使用中也更加能强化我们对知识的理解。

本篇，我们就来讲一下使用 CSS，做一个好看的按钮。

2. 目标分析

在开始阶段，我们自己也可能也不知道要做一个什么样的按钮，所以我们可以找一个参考目标。我选择了大家都非常熟悉的百度网的搜索按钮，如下图：



该按钮样式简单大气，尺寸适宜，颜色也低调朴实，所以我们决定就仿它了。

3. 外观设计

3.1 普通按钮

首先我们开发一个按钮，并为其设置样式类 `button-info`，后续通过该样式类，对按钮的外观进行设计。

```
<body>
  <input type="button" value="百度一下" class="button-info">
</body>
```

此时效果如下，毫无疑问，不如百度那个好看。



3.2 尺寸仿照

在百度网，右键点击按钮，选择检查，可以看到按钮的尺寸。如下图：



当然这是利用了 Chrome 浏览器的开发者工具，得出该按钮尺寸是 108*44。如果不会用开发者工具也没啥，反复尝试修改几次宽度、高度，差不多跟百度按钮一样大就行。

所以此时，我们将按钮尺寸设为：

```
.button-info {  
    width: 108px;  
    height: 44px;  
}
```

此时效果如下：



3.3 修改颜色与背景色

通过观察，我们发现文字颜色是白色，而背景色是一种蓝色，可以通过取色器(TakeColor 等软件)，捕获到该蓝色值为 #4569FF，所以我们进一步完善样式类如下：

```
.button-info {  
    width: 108px;  
    height: 44px;  
    background-color: #4569FF;  
    color: white;  
}
```

此时效果如下：



3.4 边框处理

通过观察百度的按钮，我们发现它应该是没有边框的，且边框有一个圆角弧度，我们可以利用 border 和 border-radius 属性来实现。

```
.button-info {  
    width: 108px;  
    height: 44px;  
    background-color: #4569FF;  
    color: white;  
    border: none;  
    border-radius: 8px;  
}
```

此时效果如下：

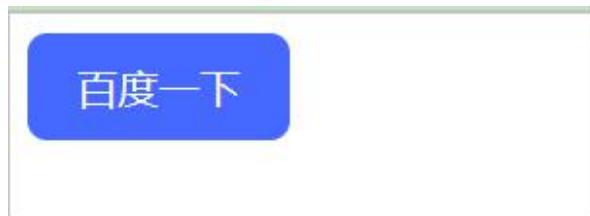


3.5 字体大小

再次观察，发现我们这个按钮字体偏小了，我们调大字体，直到跟百度字体字体差不多大。

```
.button-info {  
    width: 108px;  
    height: 44px;  
    background-color: #4569FF;  
    color: white;  
    border: none;  
    border-radius: 8px;  
    font-size: 1.05em;  
}
```

此时效果如下，已经比较接近百度的效果了。



4. 效果设计

我们还可以利用伪类选择器，为按钮增加一些“动态”的效果。代码如下：

```
.button-info:hover {  
    border: 2px solid lightblue;  
}  
  
.button-info:active {  
    border: 2px solid blue;  
}
```

上述代码编写后，当按钮悬停时，会出现一个淡蓝色的边框，效果如下：



而当按钮点击时，会出现一个蓝色的边框，效果如下：



大家可以去实际体会下，加了效果设计后，明显感觉更 nice 了。

5. 小结

本节介绍了如何利用 CSS 开发一个比较美观的按钮，大家可以参考本节内容，设计更多好看的按钮。

零基础 CSS 入门教程(27)--图片实例

1. 前言

上一篇我们详细讲解了如何利用 CSS，来制作一个好看的按钮。

本篇我们来研究下如何用 CSS 美化图片。

2. 普通图片

普通情况下，我们给图片设置个宽度和高度即可。

```
普通图片：<br>
<br>
```

对应 CSS：

```
.img {
    width: 400px;
    height: 300px;
}
```

此时效果如下，比较生硬，难言美观：

普通图片：



3. 圆角图片

我们可以通过为图片添加圆角，让图片变得不那么生硬，虽然是一个小的改变，但是效果不错：

```
圆角图片：<br>
<br>
```

对应 CSS：

```
.img-round {
    border-radius: 16px;
}
```

效果如下：

圆角图片：



4. 缩略图效果

还可以借助边框、内边距，实现一种好看且常用的缩略图效果，如下：

```
缩略图：<br>
<br>
```

对应 CSS：

```
.img-thumbnail {
    border: 1px solid grey;
    border-radius: 16px;
    padding: 4px;
}
```

此时效果如下，很明显缩略图的方式，看起来还是比较大气的。

缩略图：



5.小结

本节简单的讲解了如何使用 CSS 美化图片效果，虽然简单，但比较实用。

零基础 CSS 入门教程(28)--导航栏实例

1. 前言

几乎所有网站都有导航栏，导航栏可以快速的定位到网站特殊内容部分。

本篇我们就来使用 CSS，制作一个比较好看的导航栏。

2. 目标效果

我看了下最常用的几个网站的导航栏，发现腾讯首页导航栏效果还不错，如下：



我们就来做一个效果差不多的导航栏。

3. 开发过程

3.1 定义列表

首先我们导航栏是由多个菜单组成的，这种包含多个子项目的 HTML 元素，我们可以选用无序列表来实现。(当然也可以选择其他元素，不过列表的语义跟菜单比较相似，推荐使用)。

编写无序列表代码如下：

```
<body>
  <ul class="nav">
    <li><a href="#">新闻</a></li>
    <li><a href="#">视频</a></li>
    <li><a href="#">图片</a></li>
    <li><a href="#">军事</a></li>
  </ul>
</body>
```

我们为列表设置了 nav 样式类，后续通过该类设置列表的样式。此时效果如下：

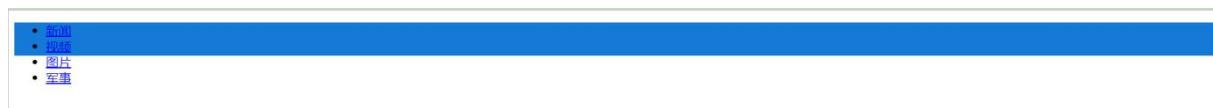
- [新闻](#)
- [视频](#)
- [图片](#)
- [军事](#)

3.2 调整高度与背景色

首先为 nav 设置高度与背景色，通过取色器取到颜色值为#1479D7，然后可以通过开发者工具，看到高度为 44px，所以 CSS 修改为：

```
.nav {
  height: 44px;
  background-color: #1479D7;
}
```

此时效果如下：



3.3 调整内外边距

通过观察，我们发现导航栏距离浏览器左右边有一定距离，可以通过 margin 调整外边距，另外列表默认样式中含有内边距部分，会影响我们的布局，索性直接把内边距设为 0。代码如下：

```
.nav {  
    height: 44px;  
    background-color: #1479D7;  
    margin: 0 100px;  
    padding: 0;  
}
```

此时效果如下：

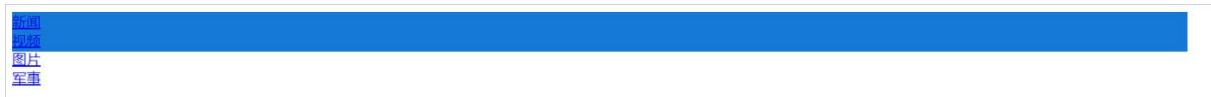


3.4 去掉项目符号

原来的效果中是没有项目符号的，我们通过 list-style-type: none; 去掉它，CSS 修改为：

```
.nav {  
    height: 44px;  
    background-color: #1479D7;  
    margin: 0 100px;  
    padding: 0;  
    list-style-type: none;  
}
```

此时效果如下：



3.5 设置列表项高度和位置

此时我们继续观察，目前我们的列表项是垂直排列的，但是实际上列表项需要水平排列。此处可以通过 `float:left;` 来实现列表项从左到右水平排列。`float` 是浮动的意思，在之前的教程中没有介绍，此处大家理解下使用 `float` 后，元素从左到右浮动排列就行了。另外我们将列表项高度也设置为 44px，跟导航栏高度保持一致。

```
.nav li {  
    height: 44px;  
    float: left;  
}
```

此时效果如下：



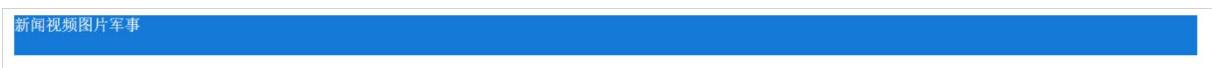
3.6 设置字体样式

通过开发者工具，我们能看到原来的字体样式和大小，此处直接借鉴过来，另外字体颜色毫无疑问是白色，然后还需要去掉超链接默认自带的下划线，所以将超级链接样式设置如下：

```
.nav li a {  
    font-family: SimSun, Arial, "Arial Narrow", HELVETICA;  
    font-size: 1.1em;  
    color: white;
```

```
text-decoration: none;  
}
```

此时效果如下：

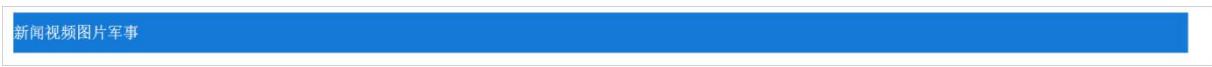


3.7 让文字垂直居中

我们知道导航栏高度为 44px，那么如何让文字垂直居中呢？此处有一个非常常用的技术，就是设置元素行高度 line-height，如果我们将元素行高度也设置为 44px，意思是文本也占据 44px 的高度，那么文本自然就会处于 44px 高度的中间了。这是一种非常常用垂直居中技术。如下：

```
.nav li a {  
    font-family: SimSun, Arial, "Arial Narrow", HELVETICA;  
    font-size: 1.1em;  
    color: white;  
    text-decoration: none;  
    line-height: 44px;  
}
```

此时效果如下：



3.8 设置列表项边距

目前列表项之间过于紧凑，我们可以通过 margin 增大边距，使其分散：

```
.nav li a {  
    font-family: SimSun, Arial, "Arial Narrow", HELVETICA;  
    font-size: 1.1em;  
    color: white;  
    text-decoration: none;  
    line-height: 44px;  
    margin: 0 18px;  
}
```

此时效果如下，可以感觉比较相似了：



3.9 添加悬停效果

当鼠标悬停时，文字颜色和背景色会变化，我们可以通过取色器获取悬停的颜色值。如下：

```
.nav li a:hover {  
    color: #14539A;  
    background-color: #CBE1ED;  
}
```

此时当鼠标经过时，效果如下：



4. 小结

至此，我们设计的导航栏跟腾讯首页的导航栏已经高度相似了，我觉得效果还不错。

大家可以去看看各网站的导航栏，好看的尝试用 CSS 模仿一下。

零基础 CSS 入门教程(29)--下拉菜单实例

1. 前言

下拉菜单也是使用频率很高的网页控件，例如百度首页上方的【设置】按钮就是一个下拉菜单，当我们的鼠标指向该按钮时，会弹出下拉菜单。



本篇我们就通过 CSS 来一步一步编写一个简单的下拉菜单。

2. 开发过程

2.1 HTML 元素组成

首先我们确定下 HTML 元素部分，有一个按钮 button，然后按钮下面是菜单区域 div，菜单区域包含多个链接 a。然后下拉菜单作为一个整体，我们用一个 div 包裹起来。所以 HTML 代码如下：

```
<body>
  <div class="dropdown">
    <button class="dropdown-button">下拉菜单</button>
    <div class="dropdown-menu">
      <a href="#">菜单 1</a>
      <a href="#">菜单 2</a>
      <a href="#">菜单 3</a>
    </div>
  </div>
</body>
```

其中 `dropdown` 表示下拉菜单，`dropdown-button` 表示按钮，`dropdown-menu` 表示菜单区域。此时由于尚未编写任何 CSS 样式代码，效果如下：



2.2 设定按钮样式

我们一步一步来处理下拉菜单，首先编写按钮的样式，代码如下，注意看注释：

```
/* 按钮样式 */
.dropdown-button {
  /* 颜色采用之前用过的百度蓝 */
  background-color: #4569FF;
  /* 字体采用白色 */
  color: white;
  /* 不要边框线 */
  border: none;
  /* 使用圆角 */
  border-radius: 5px;
  padding: 5px 10px;
  font-size: 14px;
  font-weight: bold;
  text-decoration: none;
  outline: none;
  cursor: pointer;
}
```

```
border-radius: 8px;  
/* 将按钮撑的大一些 */  
padding: 8px;  
}
```

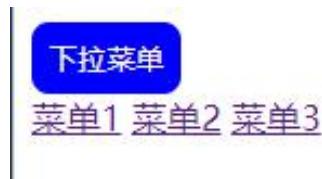
此时效果如下：



另外我们希望当鼠标悬停到下拉菜单区域时，按钮颜色能变化，这样有一个动态的效果，所以添加 CSS 代码：

```
/* 当鼠标悬停于下拉菜单上方时，按钮样式 */  
.dropdown:hover .dropdown-button {  
background-color: blue;  
}
```

此时当鼠标移到下拉菜单上方时，颜色变化为更深的蓝色。



2.3 设置菜单区域样式

首先菜单区域默认应该是不显示的，然后应设定一个宽度和边框，让下拉菜单区域能够有一个好看的边界：

```
/* 下拉菜单 */
.dropdown-menu {
    /* 默认不显示 */
    display: none;
    /* 设置宽度 */
    width: 160px;
    /* 设置边框 */
    border: 2px solid #E5F1FB;
}
```

此时效果如下：



当鼠标悬停到下拉菜单上方时，菜单区域应显示出来：

```
/* 鼠标悬停时，显示菜单区域 */
.dropdown:hover .dropdown-menu {
    display: block;
}
```

所以悬停时，效果如下：



2.4 设置菜单中链接样式

菜单中的超级链接应该是垂直排列，此时我们可以调整其盒模型为 block，这样就独占一块了。另外

链接字体设为黑色，并通过 padding 增大间距。

```
/* 链接样式 */
.dropdown-menu a {
    /* 换行 */
    display: block;
    /* 黑色字体 */
    color: black;
    /* 增大间距 */
    padding: 8px 12px;
    /* 去掉下划线 */
    text-decoration: none;
}
```

此时效果如下：



然后我们再给鼠标悬停的链接增加一个选中效果：

```
/* 链接悬停样式 */
.dropdown-menu a:hover {
    /* 颜色值为按钮默认颜色，有呼应效果 */
    background-color: #4569FF
}
```

悬停时，效果如下：



2.5 设置下拉菜单样式

还有一个非常重要的点，可能不容易发现，就是目前下拉菜单整体是一个 div，它会占用一个块，也就是说整行的空间都被其占用。表现为鼠标移到下拉菜单右侧时，也会引发菜单区域展开。

但是我们实际希望下拉菜单，只占用看到的那些区域就 OK，此时很简单，设置盒模型为 inline-block 即可，行内-块级元素，即占据空间，又不会引起换行。

```
/* 整个下拉菜单 */
.dropdown {
    display: inline-block;
}
```

此时鼠标只有在下拉菜单上方，才会触发菜单区域展开。

3. 小结

下拉菜单已经做出来了，其实还不够完善，要做一个完美的网页控件，可能需要大量的工作，尚需努力！

零基础 CSS 入门教程(30)--布局实例

1. 前言

本篇是 JavaWeb 学习之路，CSS 部分的最后一章。

如果大家一直在跟随我的脚步，想必获益匪浅吧，恭喜你们，在学习之路上又迈出了坚实的一步。

我也会继续努力，把后面的 JavaScript/jQuery/Java/MySQL 等内容讲好，从而培养真正掌握 Java Web 开发技能的技能人才。

2. 本章任务

本章介绍如何实现一个包含标题栏、导航栏、内容区域、底部版权区域，且内容区域左右分列的标准网站，如下图：



3. 开发过程

3.1 设定全局样式

通过 CSS 的通用选择器(*)设置所有元素内外边距为 0px，然后设置 body 区域背景色为白色。

```
/* 全局 */
* {
    margin: 0px;
    padding: 0px;
}

body {
    background: white;
}
```

3.2 头部标题栏样式

头部标题栏包含标题：

```
<div class="header">
    <h1>熊猫诗歌网</h1>
</div>
```

为标题设置背景色、文字水平居中、标题栏高度控制在 80px，且通过 line-height 高度等于标题栏高度使文字垂直居中。

```
/* 头部 */
.header {
    /* 标题栏高度 */
```

```
height: 80px;  
/* 垂直居中 */  
line-height: 80px;  
background: white;  
/* 水平居中 */  
text-align: center;  
}
```

此时效果如下：



熊猫诗歌网

3.3 导航栏样式

在 [JavaWeb 学习之路\(28\)-CSS 应用实例之导航栏](#) 中，我们详细介绍了如何实现一个导航栏，此处我们直接复用第 28 章中编写的导航栏。之前文章讲的很详细了，此处不再具体讲解如何实现导航栏。

注意我们修改了一个地方，就是去掉了 nav 的 margin，这样让导航栏宽度占满屏幕。

```
<ul class="nav">  
  <li><a href="#">唐诗</a></li>  
  <li><a href="#">宋词</a></li>  
  <li><a href="#">元曲</a></li>  
  <li><a href="#">现代诗</a></li>  
</ul>
```

CSS:

```
/* 导航 */
.nav {
    height: 44px;
    background-color: #1479D7;
    padding: 0;
    list-style-type: none;
}

.nav li {
    height: 44px;
    float: left;
}

.nav li a {
    font-family: SimSun, Arial, "Arial Narrow", HELVETICA;
    font-size: 1.1em;
    color: white;
    text-decoration: none;
    line-height: 44px;
    margin: 0 18px;
}

.nav li a:hover {
    color: #14539A;
    background-color: #CBE1ED;
}
```

此时效果如下：



3.4 内容区域

内容区域分为左右两部分，所以代码如下：

```
<div class="row">
  <div class="left-area">
    ...
  </div>
  <div class="right-area">
    ...
  </div>
</div>
```

我们通过 float，使左右两部分从左至右横向排列，然后设置左右侧比例分别为 70%、25%，注意加起来要比 100% 少一些，不然 margin 或 padding 占据的空间会使整个宽度超过 100%。

```
/* 左侧 */
.left-area {
  min-height: 750px;
  float: left;
  width: 70%;
  margin-left: 16px;
}

/* 右侧 */
.right-area {
  float: left;
  width: 25%;
  margin-left: 24px;
}
```

在左侧区域添加诗歌：

```
<div class="left-area">
  <div class="poem-box">
    <h2>凉州词</h2>
    <p>
      黄河远上白云间，一片孤城万仞山，羌笛何须怨杨柳，春风不度玉门关
    </p>
  </div>
  <div class="poem-box">
    <h2>将进酒</h2>
    <p>
      君不见黄河之水天上来，奔流到海不复回。
      君不见高堂明镜悲白发，朝如青丝暮成雪。
      人生得意须尽欢，莫使金樽空对月。
      天生我材必有用，千金散尽还复来。
      烹羊宰牛且为乐，会须一饮三百杯。
      岑夫子，丹丘生，将进酒，杯莫停。
      与君歌一曲，请君为我倾耳听。
      钟鼓馔玉不足贵，但愿长醉不愿醒。
      古来圣贤皆寂寞，惟有饮者留其名。
      陈王昔时宴平乐，斗酒十千恣欢谑。
      主人何为言少钱，径须沽取对君酌。
      五花马、千金裘，呼儿将出换美酒，与尔同销万古愁。
    </p>
  </div>
  <div class="poem-box">
    <h2>春江花月夜</h2>
    <p>
      春江潮水连海平，海上明月共潮生。
      潺滟随波千万里，何处春江无月明！
      江流宛转绕芳甸，月照花林皆似霰；
      空里流霜不觉飞，汀上白沙看不见。
      江天一色无纤尘，皎皎空中孤月轮。
    </p>
  </div>

```

```
江畔何人初见月？江月何年初照人？  
人生代代无穷已，江月年年望相似。
```

```
</p>  
</div>  
</div>
```

为诗歌添加简单的样式：

```
/* 诗歌 */  
.poem-box {  
    background-color: #E9EBEC;  
    padding: 10px;  
    margin: 24px 0px;  
}
```

为右侧区域添加作者：

```
<div class="right-area">  
    <div class="author-box">  
        <h3>诗人</h3>  
        <p>李白</p>  
        <p>杜甫</p>  
        <p>白居易</p>  
    </div>  
</div>
```

为作者区域添加简单的样式：

```
/* 作者 */  
.author-box {  
    background-color: #E9EBEC;
```

```

padding: 8px;
margin: 24px 0px;
}

/* 条目 */
.author-box p {
    padding: 16px 0px;
}

```

此时效果如下：

The screenshot shows a website layout with a header '熊猫诗歌网' (Panda Poetry and Song Network). Below the header is a blue navigation bar with tabs for '唐诗' (Tang Poems), '宋词' (Song Ci), '元曲' (Yuan Qu), and '现代诗' (Modern Poems). The main content area contains three poem cards:

- 凉州词**
黄河远上白云间，一片孤城万仞山，羌笛何须怨杨柳，春风不度玉门关
- 将进酒**
君不见黄河之水天上来，奔流到海不复回。君不见高堂明镜悲白发，朝如青丝暮成雪。人生得意须尽欢，莫使金樽空对月。天生我材必有用，千金散尽还复来。烹羊宰牛且为乐，会须一饮三百杯。岑夫子，丹丘生，将进酒，杯莫停。与君歌一曲，请君为我倾耳听。钟鼓馔玉不足贵，但愿长醉不复醒。古来圣贤皆寂寞，惟有饮者留其名。陈王昔时宴平乐，斗酒十千恣欢谑。主人何为言少钱，径须沽取对君酌。五花马、千金裘，呼儿将出换美酒，与尔同销万古愁。
- 春江花月夜**
春江潮水连海平，海上明月共潮生。滟滟随波千万里，何处春江无月明！江流宛转绕芳甸，月照花林皆似霰；空里流霜不觉飞，汀上白沙看不见。江天一色无纤尘，皎皎空中孤月轮。江畔何人初见月？江月何年初照人？人生代代无穷已，江月年年望相似。

On the right side, there is a sidebar with a '诗人' (Poet) section containing links to '李白' (Li Bai), '杜甫' (Du Fu), and '白居易' (Bai Juyi).

3.5 底部版权区域

在底部添加版权区域：

```

<div class="footer">
    版权所有：熊猫大哥大
</div>

```

为底部区域添加简单样式，需要注意的是由于内容区域使用了 float 浮动，到底部的时候需要通过 clear: both; 清除浮动，从而继续往下占据正常的空间。

```
/* 底部 */
.footer {
    clear: both;
    background: #5576BD;
    height: 48px;
    line-height: 48px;
    text-align: center;
}
```

最终效果如下：

熊猫诗歌网

唐诗 宋词 元曲 现代诗

凉州词
黄河远上白云间，一片孤城万仞山，羌笛何须怨杨柳，春风不度玉门关

将进酒
君不见黄河之水天上来，奔流到海不复回。君不见高堂明镜悲白发，朝如青丝暮成雪。人生得意须尽欢，莫使金樽空对月。天生我材必有用，千金散尽还复来。烹羊宰牛且为乐，会须一饮三百杯。岑夫子，丹丘生，将进酒，杯莫停。与君歌一曲，请君为我倾耳听。钟鼓馔玉不足贵，但愿长醉不复醒。古来圣贤皆寂寞，惟有饮者留其名。陈王昔时宴平乐，斗酒十千恣欢谑。主人何为言少钱，径须沽取对君酌。五花马、千金裘，呼儿将出换美酒，与尔同销万古愁。

春江花月夜
春江潮水连海平，海上明月共潮生。滟滟随波千万里，何处春江无月明！江流宛转绕芳甸，月照花林皆似霰；空里流霜不觉飞，汀上白沙看不见。江天一色无纤尘，皎皎空中孤月轮。江畔何人初见月？江月何年初照人？人生代代无穷已，江月年年望相似。

诗人

李白

杜甫

白居易

版权所有：熊猫大哥大

4. 小结

整体还是从上到下，依次编写。中间内容区域通过浮动分为两栏，整体还是比较简单的。

零基础 CSS 入门教程(31)--代码格式化

1. 背景

为什么单独拿出一章来讲代码格式化的问题，是因为我发现初学者，不理解为什么要进行代码格式化，以及如何进行代码格式化。

本篇就来说一下这方面的问题。

2. 为什么要进行代码格式化

先来看一段代码：

```
<!DOCTYPE html>
<html><head>
<meta charset="utf-8">
<style>      </style>
</head><body>
    <div>hello
</div></body></html>
```

这样的代码，虽然也没错，但是看起来非常费劲，让人难以理解。

我们对比下格式化后的代码：

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
```

```
<style>
</style>
</head>
<body>
  <div>
    hello
  </div>
</body>
</html>
```

是不是一目了然，干净清爽。这就是代码格式化的意义，让代码更加易读易懂易修改。代码格式化，就好比收拾屋子，只有把屋子收拾干净整齐了，你住起来才舒服。代码只有格式化了，你用起来才方便。

另外，在编程界，代码格式化是最基本的素养，如果你出去面试，代码格式化这块搞不好，面试官上来就会觉得你很不专业。

所以作为初学者，务必从一开始，就要注意代码格式化的问题。

3. 如何进行代码格式化

代码格式化是靠缩进来实现的，一般来说，子标签相对于父标签，要多缩进一个 tab 键的距离。

例如：

```
<body>
  <div>
    hello
  </div>
</body>
```

div 是 body 的子标签，所以 div 比 body 多缩进了 4 个空格（此处也是一个 tab 键）的距离。

hello 是 div 的内容（相当于孩子），所以 hello 比 div 多缩进了 4 个空格。

所以对一个 HTML 页面来说，最规范的缩进，应该是这样子的：

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <style>
    </style>
  </head>
  <body>
    <div>
      hello
    </div>
  </body>
</html>
```

但是由于 head、body 大家太熟悉了，也是非常底层的元素（离 html 很近，是 html 的亲生孩子），所以我们也可以把 head、body 放到跟 html 同级的位置：

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <style>
    </style>
  </head>
  <body>
```

```
<div>
  hello
</div>
</body>
</html>
```

这两种写法，都 OK。

4. 如何快捷的进行格式化

在 VSCode 中，可以按 Alt+Shift+F 键，代码就会自动格式化。

其他的开发工具或者代码编辑器，也都有类似的快捷键，大家百度一下即可。

5. 小结

代码格式化，很重要，要养成习惯。