

第 6 章



移动 Web 界面样式

本章主要介绍可以在移动 Web 应用中使用的 CSS3 新特性，同时还介绍如何使用 Media Queries 样式模块在传统网站的样式结构下增加移动 Web 版本网站。

6.1 CSS3

CSS2.1 发布至今已经有 7 年的历史，在这 7 年里，互联网的发展已经发生了翻天覆地的变化。CSS2.1 有时候难以满足快速提高性能、提升用户体验的 Web 应用的需求。CSS3 标准的出现就是增强 CSS2.1 的功能，减少图片的使用次数以及解决 HTML 页面上的特殊效果。

在 HTML5 逐渐成为 IT 界最热门话题的同时，CSS3 也开始慢慢地普及起来。目前，很多浏览器都开始支持 CSS3 部分特性，特别是基于 Webkit 内核的浏览器，其支持力度非常大。在 Android 和 iOS 等移动平台下，正是由于 Apple 和 Google 两家公司大力推广 HTML5 以及各自的 Web 浏览器的迅速发展，CSS3 在移动 Web 浏览器下都能到很好的支持和应用。

CSS3 作为在 HTML 页面担任页面布局和页面装饰的技术，可以更加有效地对页面布局、字体、颜色、背景或其他动画效果实现精确的控制。

目前，CSS3 是移动 Web 开发的主要技术之一，它在界面修饰方面占有重要的地位。由于移动设备的 Web 浏览器都支持 CSS3，对于不同浏览器之间的兼容性问题，它们之间的差异非常小。不过对于移动 Web 浏览器的某些 CSS 特性，仍然需要做一些兼容性的工作。

当前，CSS3 技术最适合在移动 Web 开发中使用的特性包括：

- 增强的选择器

- 阴影
- 强大的背景设置
- 圆角边框

接下来的章节我们将会重点介绍如何使用这些 CSS3 特性来实现移动 Web 界面。

6.2 选择器

选择器是 CSS3 中一个重要的部分，通过使用 CSS3 的选择器，可以提高开发人员的工作效率。在本节中，我们将为读者介绍属性选择器和伪类选择器的基本用法。

6.2.1 属性选择器

在 CSS3 中，我们可以使用 HTML 元素的属性名称选择性地定义 CSS 样式。其实，属性选择器早在 CSS2 中就被引入了，其主要作用就是为带有指定属性的 HTML 元素设置样式。例如，通过指定 div 元素的 id 属性，设定相关样式。

属性选择器一共分为 4 种匹配模式选择器：

- 完全匹配属性选择器
- 包含匹配选择器
- 首字符匹配选择器
- 尾字符匹配选择器

1. 完全匹配属性选择器

其含义就是完全匹配字符串。当 div 元素的 id 属性值为 test 时，利用完全匹配选择器选择任何 id 值为 test 的元素都使用该样式。如下代码通过指定 id 值将属性设定为红色字体：

```
<div id="article">测试完全匹配属性选择器</div>  
<style type="text/css">  
[id=article]{  
    color:red;  
}  
</style>
```

HTML5 移动 Web 开发指南

2. 包含匹配选择器

包含匹配比完全匹配范围更广。只要元素中的属性包含有指定的字符串，元素就使用该样式。

其语法是：[attribute*=value]。其中 attribute 指的是属性名，value 指的是属性值，包含匹配采用“*”符号。

例如下面三个 div 元素都符合匹配选择器的选择，并将 div 元素内的字体设置为红色字体：

```
<div id="article">测试完全匹配属性选择器</div>
<div id="subarticle">测试完全匹配属性选择器</div>
<div id="article1">测试完全匹配属性选择器</div>
<style type="text/css">
[id*=article]{
    color:red;
}
</style>
```

3. 首字符匹配选择器

首字符匹配就是匹配属性值开头字符，只要开头字符符合匹配，则元素使用该样式。

其语法是：[attribute^=value]。其中 attribute 指的是属性名，value 指的是属性值，首字符匹配采用“^=”符号。

例如下面三个 div 元素使用首字符匹配选择器后，只有 id 为 article 和 article1 的元素才被设置为红色字体。

```
<div id="article">测试完全匹配属性选择器</div>
<div id="subarticle">测试完全匹配属性选择器</div>
<div id="article1">测试完全匹配属性选择器</div>
<style type="text/css">
[id^=article]{
    color:red;
}
</style>
```

4. 尾字符匹配选择器

尾字符匹配跟首字符匹配原理一样。尾字符只匹配结尾的字符串，只要结尾字符串符合匹配，则元素使用该样式。

其语法是：[attribute\$=value]。其中 attribute 指的是属性名，value 指的是属性值，尾字符匹配采用“\$=”符号。

例如下面三个 div 元素使用尾字符匹配选择器时，只有 id 为 subarticle 的元素才被设置为红色字体。

```
<div id="article">测试完全匹配属性选择器</div>
<div id="subarticle">测试完全匹配属性选择器</div>
<div id="article1">测试完全匹配属性选择器</div>
<style type="text/css">
[id$=article]{
    color:red;
}
</style>
```

6.2.2 伪类选择器

在 CSS3 选择器中，伪类选择器种类非常多。然后在 CSS2.1 时代，伪类选择器就已经存在，例如超链接的四个状态选择器：a:link、a:visited、a:hover、a:active。

CSS3 增加了非常多的选择器，其中包括：

- first-line 伪元素选择器
- first-letter 伪元素选择器
- root 选择器
- not 选择器
- empty 选择器
- target 选择器

这些伪类选择器是 CSS3 新增的选择器，它们都能得到在 Android 和 iOS 平台下 Web 浏览器的支持。现在我们就为你介绍这部分的选择器。

1. before

before 伪类元素选择器主要的作用是在选择某个元素之前插入内容，一般用于清除浮动。

目前，before 选择器得到支持的浏览器包括：IE8+、Firefox、Chrome、Safari、Opera、Android Browser 和 iOS Safari。

before 选择器的语法是：

HTML5 移动 Web 开发指南

```
元素标签:before{
    content:"插入的内容"
}
```

例如，在 p 元素之前插入“文字”：

```
p:before{
    content:"文字"
}
```

2. after

after 伪类元素选择器和 before 伪类元素选择器原理一样，但 after 是在选择某个元素之后插入内容。

目前，before 选择器得到支持的浏览器包括：IE8+、Firefox、Chrome、Safari、Opera、Android Browser 和 iOS Safari。

after 选择器的语法是：

```
元素标签:after{
    content:"插入的内容"
}
```

3. first-child

指定元素列表中第一个元素的样式。语法如下：

```
li:first-child{
    color:red;
}
```

4. last-child

和 first-child 是同类型的选择器。last-child 指定元素列表中最后一个元素的样式。语法如下：

```
li:last-child{
    color:red;
}
```

5. nth-child 和 nth-last-child

nth-child 和 nth-last-child 可以指定某个元素的样式或从后数起某个元素的样式。

例如：

```
//指定第 2 个 li 元素  
li:nth-child(2){}  
//指定倒数第 2 个 li 元素  
li:nth-last-child{}  
//指定偶数个 li 元素  
li:nth-child(even){}  
//指定奇数个 li 元素  
li:nth-child(odd){}
```

本节我们只介绍了部分常用的 CSS 选择器，实际上选择器并不止这几种，其余的选择器不再详细介绍，有兴趣的读者可以阅读 CSS3 相关资料。

6.3 阴影

现在，CSS3 样式已经支持阴影样式效果。目前可以使用的阴影样式一共分成两种：一种是文本内容的阴影效果，另一种是元素阴影效果。下面我们就来为读者介绍这两种阴影样式。

6.3.1 box-shadow

CSS3 的 box-shadow 属性是让元素具有阴影效果，其语法是：

```
box-shadow:<length> <length> <length> || color;
```

其中，第一个 length 值是阴影水平偏移值；第二个 length 值是阴影垂直偏移值；第三个 length 值是阴影模糊值。水平和垂直偏移值都可取正负值，如 4px 或-4px。

目前，box-shadow 已经得到 Firefox 3.5+、Chrome 2.0+、Safari 4+等现代浏览器的支持。

可是，当我们在基于 Webkit 的 Chrome 和 Safari 等浏览器上使用 box-shadow 属性时，需要将属性的名称写成-webkit-box-shadow 的形式。Firefox 浏览器则需要写成-moz-box-shadow 的形式。

从浏览器支持的情况来看，基于 Android 和 iOS 的移动 Web 浏览器也完全支持 box-shadow 属性。因此，我们在编写 CSS 样式时可以使用 box-shadow 属性来修饰移动

HTML5 移动 Web 开发指南

Web 应用程序的界面。

下面代码为使用 `box-shadow` 的简单示例，该示例兼容 Chrome 浏览器、Safari 浏览器及 Firefox 浏览器：

```
<style type="text/css">
div{
    /*其他浏览器*/
    box-shadow:3px 4px 2px #000;
    /*webkit 浏览器*/
    -webkit-box-shadow:3px 4px 2px #000;
    /*Firefox 浏览器*/
    -moz-box-shadow:3px 4px 2px #000;
    padding:5px 4px;
}
</style>
```

6.3.2 text-shadow

`text-shadow` 属性用于设置文本内容的阴影效果或模糊效果。

目前，`text-shadow` 属性已经得到 Safari 浏览器、Firefox 浏览器、Chrome 浏览器和 Opera 浏览器的支持。IE8 版本以下的 IE 浏览器都不支持该特性。从 Web 浏览器支持的情况来看，大部分移动平台的 Web 浏览器都能得到很好的支持。

`text-shadow` 的语法和 `box-shadow` 的语法基本上一致：

```
text-shadow:<length> <length> <length> || color
```

如下代码为使用 `text-shadow` 的简单示例：

```
<style type="text/css">
div{
    text-shadow:5px -10px 5px red;
    color:#666;
    font-size:16px;
}
</style>
```

6.4 背景

在过去,我们经常使用 CSS 的背景属性对页面进行美化修饰,可惜的是,背景属性功能单一,往往无法满足页面修饰的需求。

在 CSS3 规范中, CSS3 对背景属性增加了很多新特性。它既能支持背景的显示范围,也支持多图片背景。最重要的是它可以通过属性设置,为背景的颜色设置渐变或任何颜色效果,功能非常丰富。

CSS3 对于背景属性的增强,以往我们使用图片来替代各种页面修饰,逐渐可以通过 CSS3 背景属性替换。这些功能对页面加载速度,特别是在移动设备平台,是一个页面性能的提升。

6.4.1 background-size

background-size 属性用于设置背景图像的大小。

目前, background-size 属性已经得到 Chrome 浏览器、Safari 浏览器、Opera 浏览器的支持,同时该属性也支持 Android 和 iOS 平台的 Web 浏览器。

background-size 属性在不同的 Web 浏览器下的语法方面有一定的差别。在基于 Webkit 的 Chrome 和 Safari 浏览器下,其写法为 -webkit-background-size; 而在 Opera 浏览器下则不需要 -webkit 前缀,只需要写成 background-size。

在移动 Web 开发项目应用中,建议采用兼容模式的写法,例如下面代码:

```
background-size:10px 5px;  
-webkit-background-size:10px 5px;
```

6.4.2 background-clip

background-clip 属性用于确定背景的裁剪区域。

虽然 background-clip 属性支持除 IE 以外的大部分 Web 浏览器,但在实际项目应用中应用范围不广。其语法是:

```
background-clip:border-box | padding-box | content-box | no-clip
```

其中 border-box 是从 border 区域向外裁剪背景; padding-box 是从 padding 区域向外裁

HTML5 移动 Web 开发指南

剪背景；content-box 是从内容区域向外裁剪背景；no-clip 是从 border 区域向外裁剪背景。

6.4.3 background-origin

background-origin 属性是指定 background-position 属性的参考坐标的起始位置。

background-origin 属性有三种值可以选择，border 值指定从边框的左上角坐标开始；content 值指定从内容区域的左上角坐标开始；padding 值指定从 padding 区域开始。

6.4.4 background

background 属性在 CSS3 中被赋予非常强大的功能。其中一个非常重要的功能就是多重背景。在过去设置图片背景时，只能使用一张图片，而在 CSS3 中，则可以设置多重背景图片，例如代码：

```
background:url(background1.png) left top no-repeat,  
           url(background2.png) left top no-repeat;
```

Chrome 浏览器和 Safari 浏览器都支持 background 属性的多重背景功能。由于它们都是基于 Webkit 的浏览器，因此该功能也支持 Android 和 iOS 移动平台的移动 Web 浏览器。但鉴于采用图片的方式设置背景会严重影响在移动 Web 端的体验，因此可以使用 Webkit 的其中一种特性对背景采用颜色渐变，而非采用图片方式。

语法如下：

```
-webkit-gradient(<type>, <port>[, <radius>]?,<point> [, <radius>]? [, <stop>]*)
```

上述语法比较复杂，对于入门新手的 CSS3 读者而言的确是一个门槛。不过不要紧，语法虽然复杂，但在实际使用时是极其简单的，甚至在一些网站上也提供该属性的可视化配置。

type 类型是指采用渐变类型，如线性渐变 linear 或径向渐变 radial。

如下代码：

```
background:-webkit-gradient(linear,0 0,0 100%,from(#FFF),to(#000));
```

上述代码的含义是定义一个渐变背景色，该渐变色是线性渐变并且是由白色向黑色渐变的。其中前两个 0 表示的是渐变开始 X 和 Y 坐标位置；0 和 100% 表示的是渐变结束 X 和 Y 坐标位置。

6.5 圆角边框

以前，我们在 Web 开发过程中，经常需要实现一些圆角的功能，一般的解决方案是使用图像文件实现边框圆角的效果。

HTML5 移动 Web 开发指南

现在，CSS3 已经能够轻松地实现圆角效果，代码中只要定义 `border-radius` 属性，就可以随意实现圆角效果。

到目前为止，`border-radius` 属性已经得到 Chrome 浏览器、Safari 浏览器、Opera 浏览器、Firefox 浏览器的支持。但浏览器之间样式名称的写法有些差别，例如 Chrome 浏览器和 Safari 浏览器需要写成 `-webkit-border-radius`；Firefox 浏览器需要写成 `-moz-border-radius`；而 Opera 浏览器则不需要加前缀，只需要写成 `border-radius` 即可。

示例代码如下：

```
border-radius:10px 5px;  
-moz-border-radius:10px 5px;  
-webkit-border-radius:10px 5px;
```

或

```
border-radius:10px 5px 10px 5px;  
-moz-border-radius:10px 5px 10px 5px;  
-webkit-border-radius:10px 5px 10px 5px;
```

需要注意的是，`border-radius` 属性是不允许使用负值的，当其中一个值为 0 时，则该值对应的角为矩形，否则为圆角。

6.6 Media Queries 移动设备样式

本节我们将为你带来一种全新的样式技术。通过 Media Queries 样式模块，可以实现根据移动设备的屏幕大小，定制网站页面的不同布局效果。它的优点是开发者只需要实现一套页面，就能够在所有平台的浏览器下访问网站的不同效果。

6.6.1 传统网站在 iPhone 上的显示问题

在开始介绍 Media Queries 知识之前，先来看看一个传统的网站在各种移动设备上的页面显示效果。

首先，图 6-1 所示的是 Google 首页传统网站在 iPhone Safari 浏览器下的效果图。



图 6-1 Google 首页在 iPhone Safari 下的传统网站效果

从图 6-1 中可以看到，网页上有很多部分的内容都因为浏览器的实际大小而缩小了字号。为什么会出现这样的效果呢？

实际上，在 iPhone 中使用 Safari 浏览器浏览传统 Web 网站的时候，Safari 浏览器为了能够将整个页面的内容在页面中显示出来，会在屏幕上创建一个 980px 宽度的虚拟布局窗口，并按照 980px 宽度的窗口大小显示网页。这样，我们所看到的效果就像图 6-1，同时网页可以允许以缩放的形式放大或缩小网页。

在过去，为了能够适应不同显示器分辨率大小，通常在设计网站或开发一套网站的时候，都会以最低分辨率 800×600 的标准作为页面大小的基础，而且还不会考虑适应移动设备的屏幕大小的页面。

但是，iPhone 的分辨率是 320×480，对于以最低分辨率大小显示的网站，在 iPhone 的 Safari 浏览器下访问的效果依然还是那么糟糕。那么，究竟这些传统的 Web 网站有没办法在 iPhone 等小屏幕的移动设备下显示正常呢？答案是可以的。

现在来看看 Google 是如何把传统网站首页变成移动版本的网站首页的，如图 6-2 所示。

HTML5 移动 Web 开发指南

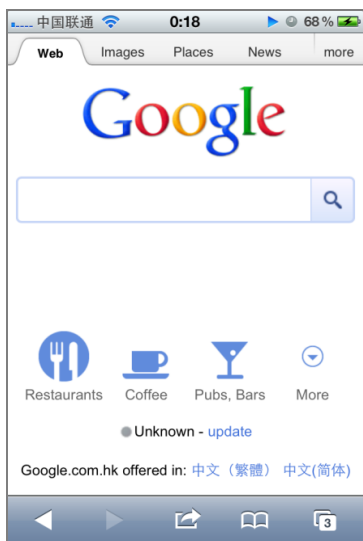


图 6-2 Google 移动版首页效果图

Google 首页转成移动版后，整个页面上的内容已经清晰可见，页面的样式风格和传统网站有一些差异。Google 究竟是如何将传统的网站转变为移动版本的网站的呢？同时，其他复杂的网站风格又需要做些什么才能变成移动版本呢？

若要实现上述的功能，我们需要在 HTML 页面用到 viewport 及 Media Queries 样式模块。

接下来我们将会介绍如何使用这两个技术知识点。

6.6.2 viewport 设置适应移动设备屏幕大小

1. 什么是 viewport

Apple 为了解决移动版 Safari 的屏幕分辨率大小问题，专门定义了 viewport 虚拟窗口。它的主要作用是允许开发者创建一个虚拟的窗口（viewport），并自定义其窗口的大小或缩放功能。

如果开发者没有定义这个虚拟窗口，移动版 Safari 的虚拟窗口默认大小为 980 像素。现在，除了 Safari 浏览器外，其他浏览器也支持 viewport 虚拟窗口。但是，不同的浏览器对 viewport 窗口的默认大小支持都不一致。默认值分别如下：

- Android Browser 浏览器的默认值是 800 像素；
- IE 浏览器的默认值是 974 像素；
- Opera 浏览器的默认值是 850 像素。

2. 如何使用 viewport

viewport 虚拟窗口是在 meta 元素中定义的，其主要作用是设置 Web 页面适应移动设备的屏幕大小。

如以下代码：

```
<meta name="viewport" content="width=device-width,  
                                initial-scale=1,user-scalable=0" />
```

该代码的主要作用是自定义虚拟窗口，并指定虚拟窗口 width 宽度为 device-width，初始缩放比例大小为 1 倍，同时不允许用户使用手动缩放功能。

在上面的代码中，我们使用了一个特别的名字：device-width。自 iPhone 面世以来，其屏幕的分辨率一致维持在 320×480。由于 Apple 在加入 viewport 时，基本上使用 width=device-width 的表达方式来表示 iPhone 屏幕的实际分辨率大小的宽度，比如 width=320。因此，其他浏览器厂商在实现其 viewport 的时候，也兼容了 device-width 这样的特性。

代码中的 content 属性内共定义三种参数。实际上 content 属性允许设置 6 种不同的参数，分别如下：

- width 指定虚拟窗口的屏幕宽度大小。
- height 指定虚拟窗口的屏幕高度大小。
- initial-scale 指定初始缩放比例。
- maximum-scale 指定允许用户缩放的最大比例。
- minimum-scale 指定允许用户缩放的最小比例。
- user-scalable 指定是否允许手动缩放。

6.6.3 Media Queries 如何工作

前面提到，Media Queries 样式模块在传统网页布局向移动版本网页布局转换中起着最重要的作用。为什么 Media Queries 能够作为在移动 Web 设备中进行页面布局的解决方案呢？首先来看一下如图 6-3 所示的效果。

从图 6-3 中可以看到，图中同一个网页在不同分辨率下一共出现三种不同布局方式。

HTML5 移动 Web 开发指南

其中，第 1 张图是在计算机的浏览器下传统网站页面布局，该布局主要是两列模式，左列显示网页的 Logo 或图片等，右列则显示导航工具栏、文章区域及头像列表等页面内容。

当用户在 iPad 等平板电脑上访问该网站的时候，由于屏幕分辨率比显示器的分辨率小很多，因此网站整个页面需要调整布局风格。首先将原来右列的导航工具栏位置调整到左列的 Logo 下面。同时，由于传统页面布局中的头像列表显示共 6 个头像，然而在 iPad 等设备上很难在一行中显示全部 6 个头像，因此需要调整一行中的头像数量，如图 6-3 中的第 2 张效果图所示，头像列表被重新定义成 2 行 3 列。



图 6-3 同一个网页、三种不同屏幕分辨率的页面布局

当用户在 iPhone 等手机上访问该网站时，其页面的布局效果就如图 6-3 中的第 3 张图所示。图中的导航工具栏位于小屏幕中页面的顶部，Logo 图片则在导航栏的下面，接着就是文章区域及头像列表区域，头像列表依然是第 2 张图的布局方式：2 行 3 列。

从该图可以看出，当用户使用 iPhone 浏览网站时，网站的页面布局方式几乎采用的是 1 行 1 列来展示页面内容。

现在，我们来看看 Media Queries 样式模块究竟能做什么样的工作，可以适配这些不同设备、不同分辨率之间的样式结构。

要实现 Media Queries 样式模块，需要在 head 标签内导入一个 CSS 样式文件，例如，下面代码使用 media 属性定义当前屏幕可视区域的宽度最大值是 600 像素时应用该样式文件。

```
<link rel="stylesheet" media="screen and(max-width:600px)" href="small.css"/>
```

在 small.css 样式文件内，需要定义 media 类型的样式，例如：

```
@media screen and (max-width:600px){  
  .demo{  
    background-color:#CCC;  
  }  
}
```

当屏幕可视区域的宽度长度在 600 像素和 900 像素之间时，应用该样式文件。导入 CSS 文件写法如下：

```
<link rel="stylesheet"  
  media="screen and(min-width:600px) and(max-width:900px)" href="small.css"/>
```

small.css 样式文件内对应写法如下：

```
@media screen and (min-width:600px) and(max-width:900px){  
  .demo{  
    background-color:#CCC;  
  }  
}
```

当手机（如 iPhone）最大屏幕可视区域是 480 像素时，应用该样式文件。导入 CSS 文件写法如下：

```
<link rel="stylesheet" media="screen and(max-device-width:480px)" href="small.css"/>
```

small.css 样式文件内对应写法如下：

```
@media screen and (max-device-width:480px){  
  .demo{  
    background-color:#CCC;  
  }  
}
```


HTML5 移动 Web 开发指南

```
    }  
}
```

同样也可以判断当移动设备（如 iPad）的方向发生变化时应用该样式。以下代码是当移动设备处于纵向（portrait）模式下时，应用 portrait 样式文件；当移动设备处于横向（landscape）模式下时，应用 landscape 样式文件。

```
<link rel="stylesheet" media="all and(orientation:portrait)" href="portrait.css"/>  
<link rel="stylesheet" media="all and(orientation:landscape)" href="landscape.css"/>
```

上述 4 种不同情况显示了使用 Media Queries 样式模块定义在各种屏幕分辨率下的不同样式应用。这种语法风格有点类似于编写兼容 IE 浏览器各个版本的方式，唯一不同的是将需要兼容 IE 的 CSS 样式导入文件写在<!--和-->之间。

在本章节中，我们通过一些简单的示例介绍了 Media Queries 样式。它的出现让开发者开发一套跨平台的网站应用成为可能。在下一节我们会总结一下 Media Queries 样式模块的基本语法。

6.6.4 Media Queries 语法总结

Media Queries 的语法如下所示：

```
@media [media_query] media_type and media_feature
```

使用 Media Queries 样式模块时都必须以“@media”方式开头。

media_query 表示查询关键字，在这里可以使用 not 关键字和 only 关键字。not 关键字表示对后面的样式表达式执行取反操作。例如如下代码：

```
@media not screen and (max-device-width:480px)
```

only 关键字的作用是，让不支持 Media Queries 的设备但能读取 Media Type 类型的浏览器忽略这个样式。例如如下代码：

```
@media only screen and (max-device-width:480px)
```

对于支持 Media Queries 的移动设备来说，如果存在 only 关键字，移动设备的 Web 浏览器会忽略 only 关键字并直接根据后面的表达式应用样式文件。对于不支持 Media Queries 的设备但能够读取 Media Type 类型的 Web 浏览器，遇到 only 关键字时会忽略这个样式文件。

虽然 media_query 这个类型在整个 Media Queries 语法中并不是必需的类型，但是有

第 6 章 移动 Web 界面样式

时候在实际开发过程中却是非常重要的查询参数类型。

`media_type` 参数的作用是指定设备类型，通常称为媒体类型。实际上在 CSS2.1 版本时已经定义了该媒体类型。表 6-1 列出了 `media_type` 允许定义的 10 种设备类型。

表 6-1 `media_type` 设备可用类型一览表

media_type	设备类型说明
all	所有设备
aural	听觉设备
braille	点字触觉设备
handed	便携设备，如手机、平板电脑
print	打印预览图等
projection	投影设备
screen	显示器、笔记本、移动端等设备
tty	如打字机或终端等设备
tv	电视机等设备类型
embossed	盲文打印机

`media_feature` 的主要作用是定义 CSS 中的设备特性，大部分移动设备特性都允许接受 `min/max` 的前缀。例如，`min-width` 表示指定大于等于该值；`max-width` 表示指定小于等于该值。

表 6-2 显示 `media_feature` 设备特性的种类一览表。

表 6-2 `media_feature` 设备特性一览表

设备特性	是否允许 min/max 前缀	特性的值	说明
width	允许	含单位的数值	指定浏览器窗口的宽度大小，如 480 像素
height	允许	含单位的数值	指定浏览器窗口的高度大小，如 320 像素
device-width	允许	含单位的数值	指定移动设备的屏幕分辨率宽度大小，如 480 像素
device-height	允许	含单位的数值	指定移动设备的屏幕分辨率高度大小，如 320 像素
orientation	不允许	字符串值	指定移动设备浏览器的窗口方向。只能指定 portrait（纵向）和 landscape（横向）两个值
aspect-ratio	允许	比例值	指定移动设备浏览器窗口的纵横比例，如 16:9
device-aspect-ratio	允许	比例值	指定移动设备屏幕分辨率的纵横比例，如 16:9
color	允许	整数值	指定移动设备使用多少位的颜色值
color-index	允许	整数值	指定色彩表的色彩数
monochrome	允许	整数值	指定单色帧缓冲器中每像素的字节数
resolution	允许	分辨率值	指定移动设备屏幕的分辨率
scan	不允许	字符串值	指定电视机类型设备的扫描方式。只能指定两种值：

HTML5 移动 Web 开发指南

			progressive 表示逐行扫描和 interlace 表示隔行扫描
grid	不允许	整数值	指定设备是基于栅格还是基于位图。基于栅格时该值为 1，否则为 0

到目前为止，Media Queries 样式模块在桌面端都得到了大部分现代浏览器的支持。例如 IE 9 浏览器、Firefox 浏览器、Safari 浏览器、Chrome 浏览器、Opera 浏览器。但是 IE 系列的浏览器中只有最新版本才支持该特性，IE8 以下的版本不支持 Media Queries。

从移动平台来说，基于两大平台 Android 和 iOS 的 Web 浏览器也都得到了良好的支持。同时，黑莓系列手机也支持 Media Queries 特性。

6.6.5 如何将官方网站移植成移动 Web 网站

接下来，让我们来看一下如何将一个真正的网站实现为移动端的 Web 网站版本。如图 6-4 所示是遇见官方网站 (<http://www.iaround.net>) 的首页在浏览器下的显示效果。

从图 6-4 可以看到，首页区域一共分成 5 部分。第一部分是页面的顶部，这部分主要是显示 Logo 及导航栏。第二部分是介绍遇见社交软件的展示区域以下载链接提示，属于介绍区域。第三部分则是介绍遇见社交软件的特性部分。第四部分主要是合作伙伴相关的友情链接。最后一部分则是页面的底部，该部分主要显示版权相关信息。

这是一个非常普通的 Web 页面，现在我们根据这四部分的页面区域内容来分析如何将整个页面转换成移动版本的首页。



图 6-4 遇见官方网站首页显示效果图

1. 导入 Media Queries 样式文件

在首页的 HTML 文件的 head 元素内新增以下 Media Queries 样式文件模块：

```
<link rel="stylesheet" type="text/css"
media="only screen and (max-width:480px),only screen and(max-device-width:
480px)"
href="/resources/style/device.css"/>
```

2. 首页 HTML 源码

接着来看一下首页的 HTML 代码，如代码 6-1 所示。

代码 6-1 首页的 HTML 代码

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8" />
<meta name="keywords" content="遇见 iphone Android symbian 陌生人" />
<meta name="description" content="遇见是一款是基于附近陌生人的社交应用，帮助你与你认识的、但就在附近的人进行即时沟通。" />
<meta name="viewport" content="width=device-width"/>
```

HTML5 移动 Web 开发指南

```
<title>遇见 - 基于附近陌生人的社交应用</title>
<link rel="stylesheet" href="/resources/style/base.css" type="text/css" />
<link rel="stylesheet" href="/resources/style/style.css" type="text/css" />
<link rel="stylesheet" href="/resources/style/home.css" type="text/css" />
<!--[if lte IE 9]>
  <link rel="stylesheet" href="/resources/style/style-ie.css" type="text/css" />
  <script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script>
  <![endif]-->
<!--[if lt IE 8]>
  <link rel="stylesheet" href="/resources/style/style-ie6.css" type="text/css"
/>
  <![endif]-->
<link rel="stylesheet" type="text/css"
  media="only screen and (max-width:480px),only screen and(max-device-width:
480px)"
  href="/resources/style/device.css"/>
</head>
<body>
  <header class="header">
    <div class="logo">
      <nav class="header-nav">
        <ul>
          <li class="selected2"><a href="#">欢迎使用</a></li>
          <li><a href="download.html">免费下载</a></li>
          <li><a href="faq.html">遇见 FAQ</a></li>
          <li><a href="http://bbs.iaround.net">官方论坛</a></li>
        </ul>
      </nav>
    </div>
  </header>
  <section class="followme">
    <em>关注我们</em>
    <a target="_blank" href="http://weibo.com/iAround"></a>
    <a target="_blank" href="http://t.qq.com/iAround"></a>
  </section>
  <section class="mobile">
    <div class="d_img"></div>
    <div class="d">
      
    </div>
  </section>
</body>
</html>
```

```
<a class="download" href="download.html"></a>

<ul class="app_type">
  <li><a class="app_ip" href="http://itunes.apple.com/cn/app/
iaround-chat-with-people-nearby/id468944728?l=en&mt=8"></a></li>
  <li><a class="app_az" href="/dl/iaround.apk"></a></li>
  <li><a class="app_sb"></a></li>
</ul>
</div>
</section>
<section class="intro">
  <div>
    <header class="title"></header>
    <div class="content">
      <ul>
        <li class="i1"><h1>无须注册登录</h1>遇见使用全新的技术，无须注册也
无须登录，下载安装后即可使用，即使您的手机重新刷机或安装系统，只需重新安装遇见客户端，之
前的所有账号信息同样保留。</li>
        <li class="i3"><h1>简单快捷流畅</h1>遇见无须查找好友、无须关注任何
人，即可与附近的陌生人进行聊天，聊天人数在 15 个左右。超过这个数字的人系统自动创建新的房间，
解决了聊天室拥挤问题，让聊天更流畅。</li>
        <li class="i5"><h1>丰富便捷聊天方式</h1>遇见支持发送图片、发送地理
位置、发送视频、发送语音，更有 500+ 的聊天表情，让用户之间的沟通更加丰富、便捷。</li>
        <li class="i7"><h1>照片分享</h1>用相片连接全世界爱好摄影的陌生人，
把他们变成好朋友和彼此吹捧的粉丝，让影像成为他们唯一对话的语言，通过遇见的附近照片的功能，
无论你走到世界哪里，都可以查看本地附近有哪些照片。</li>
      </ul>
      <ul>
        <li class="i2"><h1>邂逅聊天</h1>遇见是基于附近陌生人的社交应用，帮
助你与你认识的、但就在附近的人进行沟通。系统会根据用户的年龄、性别、星座、职业、兴趣、
教育等信息，优先自动配对最适合的且在附近的一群人进行交流和沟通。</li>
        <li class="i4"><h1>跨平台</h1>遇见将会陆续推出 iPhone 版、Android
版、Symbian、iPad、黑莓版本、winPhone 版本，实现移动设备的全平台，不管您的朋友使用什么
手机，都可以使用遇见。</li>
        <li class="i6"><h1>多语言</h1>遇见已有简体中文版、繁体中文版、英文
版多语言版本。即将推出日文版、韩文版、西班牙语版、法文版、德文等版本，面向全球用户。</li>
        <li class="i8"><h1>免费短信</h1>您不仅可以与附近的陌生人交友聊天，
还可以免费向手机通讯录中的好友发送短信消息，支持推送功能，所以有人来消息后可以像标准短信
一样显示在手机上，不用担心漏看消息。</li>
      </ul>
```

HTML5 移动 Web 开发指南

```
</div>
<footer class="footer"></footer>
</div>
</section>
<section class="friendlink">
  <h1>合作伙伴: </h1>
  <ul>
    <!-- 此部分略去所有友情链接代码 -->
    ...
  </ul>
</section>
<footer class="footer">
  <div class="bg">
    <nav>
      <ul>
        <li><a 关于 我们 </a><span>|</span></li><li><a 服务 条款
</a><span>|</span></li><li><a 联系 我们 </a><span>|</span></li><li><a 遇见 帮助
</a></li>
      </ul>
    </nav>
    <div class="copyright">
      Guangzhou Zoega Information Technology Co., Ltd. 版权所有 粤 ICP 备
09077414
    </div>
  </div>
</div>
</body>
</html>
```

3. device.css 适应手机浏览器屏幕的样式

然后, 通过 device.css 文件重新修饰首页的 CSS 部分样式, 让整个网页能够适应手机浏览器上访问的 1 行 1 列的排列方式, 如代码 6-2 所示。

代码 6-2 device.css 样式代码

```
@media only screen and (max-width:480px),only screen and (max-device-width:480px){
  .header .logo{
    margin-left:0px;
    position:relative;
    background:url(/resources/images/device-logo.jpg) 0px 37px no-repeat;
    width:100%;
```

```
    left:0;
    height:150px;
}
.header .header-nav{
    padding:0 0;
}
.header .header-nav ul{
    text-align:left;
    -webkit-padding-start:0;
}
.header .header-nav ul li{
    display:inline-block;
}
.header .header-nav ul li a{
    padding:0 0.2em;
}
section.followme {
    position:relative;
    left:0;
    margin:0 0;
    text-align:right;
    width:100%;
}
.mobile{
    margin:0 0;
    text-align:left;
    height:775px;
    position: relative;
    width:100%;
    left:0;
    background:url(/resources/images/device-mobile.jpg) left top no-repeat;
}
.mobile .d{
    margin:0 0 0 -550px;
    padding:270px 0 0 0;
}
.intro {
    margin: 1em 0 2em 0.5em;
    text-align:center;
    position: relative;
    width:95%;
```


HTML5 移动 Web 开发指南

```
        left:0;
    }
    .intro header{
        width:100%;
        height:0;
        background:none;
    }
    .intro footer {
        width:100%;
        height:0;
        background:none;
    }
    .intro .content{
        padding: 1em 0.6em;
        border-radius:0.6em;
        -webkit-border-radius:0.6em;
    }
    .intro ul {
        display: inline-block;
        width: 100%;
    }
    .intro ul li{
        margin: 0 0 5.8em 0;
        line-height: 22px;
    }
    .friendlink {
        margin:1em 0.6em 2em 0.6em;
        text-align:left;
        position:relative;
        width:95%;
        left:0;
    }
    .platform {
        margin:0 .4em;
        width:95%;
        left:0;
        border-radius:0.4em;
        -webkit-border-radius:0.4em;
    }
    .platform ul li{
        border-right:0;
```

```
padding: 1.4em 3em;
}
.phoneModel {
margin:0.4em .4em;
border:1px solid #CFD1D6;
width:95%;
left:0;
border-radius:0.4em;
-webkit-border-radius:0.4em;
}
}
```

从上面的 CSS 样式代码中可以看到，实际上大部分的 CSS 代码都是最常用的样式属性。唯一的区别就是多了 @media 元素括住整块 CSS 代码。

然而，细心的读者可以看到，device.css 里面的很多代码都是在原有传统网站的样式代码的基础上通过继承模式重新设置其属性样式，也就是我们常说的兼容性写法。

经过 device.css 样式文件重新定义页面布局后，其页面在 iPhone Safari 浏览器下的效果如图 6-5 所示。

HTML5 移动 Web 开发指南



图 6-5 遇见官方网站移动版首页页面效果图

6.7 本章小结

本章主要介绍了 CSS3 中常用的几个属性的基本语法和解释，但并没有针对 CSS3 的重要特性进行展开并详细讲解。

在本章的最后还重点介绍了 Media Queries 样式模块如何将传统网站制作成移动版 Web 网站。