

## 第十讲：认识 CSS 3

学时计划：2 学时 理论，0 学时 实验

（无实验教学内容）

教学大纲：

- 1、CSS 3 概述
- 2、使用 CSS 3 能够做什么？
- 3、CSS 3 的新特征
- 4、在 HTML 中使用 CSS
- 5、案例：图像边框
- 6、案例：为 Firefox、Chrome、Safari 定义不同的样式
- 7、案例：为 Phone、Pad、PC 定义不同的样式
- 8、讨论与思考

CSS 是一种与 HTML 并存的语言，它能够为文档元素设置可视化样式，如尺寸、颜色、背景和边框等。目前，CSS 已经得到了浏览器的广泛支持。

本讲主要介绍样式表和 CSS 3 的基础支持，并向读者介绍 CSS 3 的一些新功能和浏览器支持情况，以及如何在 HTML 中使用 CSS。最终通过三个案例，展示 CSS 3 的诱人魅力。

### 一、CSS 3 概述

从 2010 年开始，HTML 5 与 CSS 3 就一直是互联网技术中最受关注的两个话题。2010 年 MIX10 上，微软的工程师从前端技术的角度，把互联网的发展分为三个阶段：第一阶段是 Web1.0 的以内容为主的网络，前端技术主要是 HTML 和 CSS；第二阶段是 Web2.0 的 Ajax 应用，热门技术是 JavaScript/DOM/异步数据请求；第三阶段是 HTML 5+CSS 3 时代，这两种技术的结合，使互联网进入了一个崭新的时代。

#### 1.1 样式表

样式表 (Style Sheet) 是一种专门描述结构文档表现方式的文

档，它既可以描述文档在屏幕上如何显示，也可以描述他们的打印效果，甚至声音效果。样式表一般不包括在结构化文档的内部，而是以独立的文档方式存在。与 HTML 描述数据显示方式的传统方式相比，样式表具有明显的优点。

#### （1）表达效果丰富。

样式表可以支持文字和图像的精确定位、三维层技术以及交互操作等，对于文档的表现力远远超过 HTML 中的标记。更重要的是，样式表的标准规范独立于其他结构文档的规范，当需要实现丰富的表达效果时，仅需要修改样式表即可。

#### （2）文档体积小。

在实际应用中，大量的结构文档中存在重复的内容定义样式，使用传统方式需要对每一个文档进行样式定义，会产生大量的重复。而在样式表中一种表现样式只需要一次定义，所有使用该样式的内容通过调用即可。

#### （3）便于信息检索。

样式表负责复杂的显示效果，但样式表的显示逻辑与数据逻辑分离，显示细节的描述并不影响文档中数据的内在结构。因此，搜索引擎可以不考虑样式表的内容，对结构文档进行检索，从而更好的检索到有用的信息。

#### （4）可读性好。

样式表对各种标记的显示进行集中定义，且定义方式直观易读。这使得样式表易学易用，可读性和可维护性都比较好。而结构化的数据文档也相对简洁、清晰，突出对内容本身的描述。

## 1.2 CSS 概述

级联样式表（Cascading Style Sheet）简称“CSS”，通常又称为“风格样式表（Style Sheet）”，它是用来进行网页风格设计的。

CSS 3 是 CSS 技术的一个升级版本，是由 Adobe Systems、Apple、Google、HP、IBM、Microsoft、Mozilla、Opera、Sun Microsystems 等许多 Web 界的巨头联合成立的“CSS Working Group”的组织共同协商策划的。

## 1.3 CSS 的发展历史

CSS 的发展历史经过了四个阶段。

### (1) CSS 1

1996 年 12 月, CSS 1 (Cascading Style Sheets, level 1) 正式推出。在这个版本中, 已经包含了 font 的相关属性、颜色与背景的相关属性、文字的相关属性、box 的相关属性等。

### (2) CSS 2

1998 年 5 月, CSS 2 (Cascading Style Sheets, level 2) 正式推出。在这个版本中开始使用样式表结构。

### (3) CSS 2.1

2004 年 2 月, CSS 2.1 (Cascading Style Sheets, level 2 revision 1) 正式推出。它在 CSS 2 的基础上略微做了改动, 删除了许多诸如 text-shadow 等不被浏览器所支持的属性。

现在所使用的 CSS 基本上都在 1998 年推出的 CSS 2 的基础上发展而来的。

### (4) CSS 3

2010 年开始, CSS 3 逐步发布。具体关于 CSS 3 的发布时间和状态, 请访问: [http://www.w3.org/standards/techs/css#w3c\\_all](http://www.w3.org/standards/techs/css#w3c_all)。

## 二、使用 CSS 3 能够做什么?

### 2.1 CSS 3 模块

CSS 规范的前几版重现完全包含了构成 CSS 语言的元素的定义。在 CSS 3 中, 整个规范已经被划分为一组较短的规范, 它们被称为模块 (module)。

每个模块为给定的功能子集方面提供定义。其中一个模块定义媒体选择器, 一个模块定义颜色, 还有一个模块定义 SVG, 等等。模块方式允许 CSS 规范的某个特定部分能够作为一个标准进行审批和实施, 这样做的速度要快于将包含所有功能的大型规范作为一个标准进行审批和实施的方式; 而后面的方式无疑要花费更长的时间才能使得规范成为人们认可的标准。总而言之, 这意味着 CSS 3 是一个不断演化和完善的标准, 其中的特定部分可能在其它部分之前变成人们认可

的标准。

表 10-1 CSS 3 中的模块

模块名称	功能描述
Basic box model	定义各种与盒相关的样式。
Line	定义各种与直线相关的样式。
Lists	定义各种与列表相关的样式。
Hyperlink Presentation	定义各种与超链接相关的样式。譬如锚的显示方式、激活时的视觉效果等。
Presentation Levels	定义页面中元素的不同样式级别。
Speech	定义各种与语音相关的样式。譬如音量、音速、说话间歇时间等属性。
Background and border	定义各种与背景和边框相关的样式。
Text	定义各种与文字相关的样式。
Color	定义各种与颜色相关的样式。
Font	定义各种与字体相关的样式。
Paged Media	定义各种页眉、页脚、页数等页面元数据的样式。
Cascading and inheritance	定义怎样对属性进行赋值。
Value and Units	将页面上各种各样的值与单位进行统一定义，以供其他模块使用。
Image Values	定义对 image 元素的赋值方式。
2D Transforms	在页面中实现二维空间上的变形效果。
3D Transforms	在页面中实现三维空间上的变形效果。
Transforms	在页面中实现平滑过渡的视觉效果。
Animations	在页面中实现动画。
CSSOM View	查看管理页面或页面的视觉效果，处理元素的位置信息。
Syntax	定义 CSS 样式表的基本结构、样式表中的一些语法细节、浏览器对于样式表的分析规则。
Generated and Replaced Content	定义怎样在元素中插入内容。
Marquee	定义当一些元素的内容太大，超出了指定的元素尺寸时，是否以及怎样显示溢出部分。
Ruby	定义页面中 ruby 元素（用于显示拼音文字）的样式。
Writing Modes	定义页面中文本数据的布局方式。
Basic User Interface	定义在屏幕、纸张上进行输出时页面的渲染方式。

Namespaces	定义使用命名空间时的语法。
Media Queries	根据媒体类型来实现不同的样式。
'Reader' Media Type	定义用于屏幕阅读器之类的阅读程序时的样式。
Multi-column Layout	在页面中使用多栏布局方式。
Template Layout	在页面中使用特殊布局方式。
Flexible Box Layout	创建自适应浏览器窗口的流动布局或自适应字体大小的弹性布局。
Grid Position	在页面中使用风格布局方式。
Generated Content for Paged Media	在页面中使用印刷时使用的布局方式。

## 2.2 CSS 3 != HTML 5

CSS 3 常被误解为 HTML 5 的一个子集。尽管 CSS 3 和 HTML 5 通常相互伴随，但是它们含义不同。HTML 5 是 HTML (Hyper Text Markup Language) 规范的第五版，它用于在 HTML 文档中定义结构、内容和功能。HTML 5 支持新的标签和更丰富媒体内容，而 CSS 3 定义用于定制用户界面呈现方式的新规则。具体来说，CSS 3 定义 HTML 内容在浏览器中的实际显示方式。

在标准发布上，HTML 5 和 CSS 3 的标准是独立发布的，且有不同的工作组开展工作。

### 2.3 浏览器对 CSS 3 的兼容性

CSS 3 得到了浏览器的广泛支持，具体的浏览器支持情况，可以通过 <http://www.w3.org/Style/CSS/software#browsers> 来进行查询。

网站 <http://caniuse.com/> 提供了在线的 CSS 3 的浏览器支持列表。可以通过 CSS 3 属性索引和对比表格的方式，方便的查看浏览器的支持情况。

Modernizr 是一个用来检测浏览器功能支持情况的 JavaScript 库。目前，通过检验浏览器对一系列测试的处理情况，Modernizr 可以检测 18 项 CSS3 功能以及 40 多项关于 HTML5 的功能。它比传统检测浏览器名称（浏览器嗅探）的方式更为可靠。且测试速度很快，一

整套测试的执行时间仅需几微秒。

更详细的内容通过访问 <http://modernizr.com/> 获得。

## 2.4 前缀

CSS 3 前缀是一个浏览器生产商经常使用的一种方式，它暗示该 CSS 属性或规则尚未成为 W3C 标准的一部分。

常用的前缀有：-webkit、-moz、-ms、-o。分别代表了 WebKit 内核的浏览器（例如 Safari、Chrome）、Firefox、IE、Opera。

当该属性称谓 CSS 3 标准时，可以去掉一个属性的 CSS 3 前缀。在属性称谓标准之前，可以通过前缀在特定浏览器上指定样式。

## 三、CSS 3 的新特征

CSS 3 是一个不断演化和完善的标准，目前已经完成的部分中，CSS 3 带来了许多优秀的特性。

### 3.1 CSS 3 的新功能

CSS 3 的新特征有很多，例如圆角效果、图形化边界、块阴影与文字阴影、使用 RGBA 实现透明效果、渐变效果、使用 @Font-Face 实现定制字体、多背景图、文字或图像的变形处理（旋转、缩放、倾斜、移动）、多栏布局、媒体查询等。

#### 演示：

- 1、Adobe 提供的 CSS 3 效果展示网站：  
地址：<http://beta.theexpressiveweb.com/>
- 2、Apple 提供的 CSS 3 效果展示网站：  
<http://www.apple.com/html5/>

### 3.2 CSS 3 的应用范围

在了解了 CSS 3 的基本概念后，下面介绍应该在何处能够实际使用它。

(1) 所有主流桌面浏览器。

目前，Internet Explorer、Chrome、Firefox、Safari 和 Opera 均支持各种 CSS 3 功能。但是，只有浏览器的最新版本才能实际支持

主要的 CSS 3 功能。

## (2) 移动浏览器。

最新的智能手机操作系统均基本支持 CSS 3。包括 iOS、Android、BlackBerry、Windows Phone 设备的默认 web 浏览器。实际上，移动浏览器具有一些 CSS 3 功能的最佳支持能力。

需要说明的是，并不是所有的桌面和移动浏览器均以相同的方式支持所有的功能。

## 四、在 HTML 中使用 CSS

CSS 样式单可以控制 HTML 文档的显示。但在控制文档的显示之前，首先应在需要显示的 HTML 文档中导入 CSS 样式单。为了在 HTML 文档中使用 CSS 样式单，有内联、嵌入、外部样式三种方式。

### 4.1 内联样式

内联样式是在元素属性中设置样式。此种方式很适合用于测试样式和快速查看样式效果，但是不推荐在整个文档上使用此方法。

在使用此方法时，必须在每一个元素上重复设置各个样式，这样既增加文档大小，也增加文档的更新和维护难度。例如，所有的<p>元素都设置了文字大小，如果需要修改，就需要反复更改多处样式定义内容。

### 案例：

#### 示例 10-1：内联样式

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>内联样式</title>
</head>

<body>
<nav>
  <ul>
    <li style="color:#F00; font-weight:bold;">首页</li>
    <li>Web 前端开发</li>
    <li>Linux 操作系统</li>
    <li>计算机网络</li>
```

```
<li>MySQL 数据库管理</li>
</ul>
</nav>
</body>
</html>
```

## 4.2 嵌入样式

嵌入样式是通过在 HTML 文档头定义样式单部分来实现的。

通常不建议使用嵌入样式，因为此方式必须在 HTML 文档内部定义样式，如果此文档的 CSS 样式需要被其他 HTML 文档使用，那么就必须要重新定义。大量 CSS 嵌套在 HTML 文档中，也会导致 HTML 文档过大，也会造成网络负担过重。如果需要修改整站风格时，必须对网站的每一个网页进行修改，不利于更新和管理。

### 案例：

#### 示例 10-2：嵌入样式

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>内联样式</title>
<style>
body {
margin:20px 0px;
color:#FFF;
font-size:13px;
font-family:微软雅黑, "Times New Roman", Times, serif;
}
nav {
width:100%;
height:30px;
padding-left:30px;
background-color:#F63;
}
li {
float:left;
padding:4px 5px 0px 5px;
list-style:none;
}
</style>
</head>
```



```
<body>
<nav>
  <ul>
    <li>首页</li>
    <li>Web 前端开发</li>
    <li>Linux 操作系统</li>
    <li>计算机网络</li>
    <li>MySQL 数据库管理</li>
  </ul>
</nav>
</body>
</html>
```

### 4.3 外部样式

外部样式是将所有样式写在一个外部文件中，然后在 HTML 文档中使用<link>元素，将文件链接到需要设置样式的文档上。使用这种方法，只需要修改链接的文件，就可以完全改变网页的整体风格。此外，也可以使用这种方法修改或调整文档，使之适应不同环境或设备的显示要求。

此种方法，是推荐使用的方式。

#### 案例：

##### 示例 10-3：外部样式

文件一：10-3.css（路径：style/10-3.css）

```
@charset "utf-8";
body {
  margin:0px 0px;
  color:#FFF;
  font-size:13px;
  font-family:微软雅黑, "Times New Roman", Times, serif;
  background:url(../images/Top-BG-1.png) #FFF no-repeat center;
}
header {
  margin:80px 0px 30px 0px;
  text-align:center;
}
nav {
  width:100%;
  height:30px;
  background-color:#F30;
}
```

```
div {
    margin:0px auto;
    width:1000px;
    height:30px;
    padding-left:30px;
}
li {
    float:left;
    padding:4px 5px 0px 5px;
    list-style:none;
}
```

文件二：10-3.html

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>外部文件</title>
<link type="text/css" href="style/10-3.css" rel="stylesheet">
</head>

<body>
<header>
    
</header>
<nav>
<div>
    <ul>
        <li>首页</li>
        <li>Web 前端开发</li>
        <li>Linux 操作系统</li>
        <li>计算机网络</li>
        <li>MySQL 数据库管理</li>
    </ul>
</div>
</nav>
</body>
</html>
```

#### 4.4 网站 CSS 文件的规划

随着 CSS 的应用重要性的增强，一个网站的 CSS 文件也就越来越多。如何规划 CSS 文件的目录和结构，对于网站的性能有着重要的意

义。常用的网站 CSS 文件规划的方法如下所示。

### (1) 基于原型

基于原型页面进行 CSS 文件的构件是最为常用的策略。通常共享的 CSS 放到主样式文件，每个页面都载入，各子页面的样式文件单独存放，在需要时载入。

### (2) 基于页面元素或模块

对于基本的元素，例如 header、body、nav、link 等元素的定义，放到主样式文件。例如 aside、article、address 等元素的定义，分别单独定义样式文件。

### (3) 基于标记

基于标记的规划是最为常见的方法。例如网站中所有的内容列表、网站版权、导航、Logo 等，定义到主样式文件中。例如表单、搜索、内容推荐等，分别定义样式文件，在需要时再进行载入。

总之，并没有规定的 CSS 文件的规划方法，在具体的应用中，要根据网站的实际，为样式文件创建专门的目录，并规划样式文件的定义。要尽量保障样式文件的重用，又不要让所有的页面过多的载入不需要的样式定义，从而造成网络流量的浪费和响应时间的降低。

## 五、案例：图像边框

### 案例：

#### 示例 10-4：图像边框

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>图像边框</title>
<style type="text/css">
div {
width:500px;
border-image-width:10px;
border-image:url(images/Border-Bg-imgs.jpg) repeat repeat;
padding:30px;
}
</style>
</head>
```

```
<body>
<div>
  <ul>
    <li>这是第一项内容
    <li>这是第二项内容
    <li>这是第三项内容
    <li>这是第四项内容
  </ul>
</div>
</body>
</html>
```

## 六、案例：为 Firefox、Chrome、Safari 定义不同的样式

### 案例：

#### 示例 10-5：CSS 3 前缀

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>为不同的浏览器设置不同的样式</title>
<style>
div {
  width:500px;
  height:300px;
  border:2px #000 solid;
  border-radius:5px;
  -moz-border-top-colors:#F00;
  -webkit-border-radius:40px;
}
</style>
</head>

<body>
<ul>
  <li>IE 浏览器：黑色边框，5 像素圆角。</li>
  <li>Firefox 浏览器：黑色边框，上边框为红色，5 像素圆角。</li>
  <li>Chrome 浏览器：黑色边框，40 像素圆角。</li>
</ul>
<div></div>
</body>
</html>
```

## 七、案例：为 Phone、Pad、PC 定义不同的样式

案例：

示例 10-6：CSS 3 判断媒介类型

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>为不同的屏幕设置不同的样式</title>
<style>
body {
  font-size:13px;
  font-family:微软雅黑, 黑体;
  color:#FFF;
}
@media screen and (max-width:600px) {
body {
  background-color:#F00;
}
}
@media screen and (min-width:600px) and (max-width:900px) {
body {
  background-color:#F0C;
}
}
@media screen and (min-width:900px) {
body {
  background-color:blue;
}
}
@media screen and (-webkit-min-device-pixel-ratio: 2) {
body {
  background-color:#0F6;
}
}
</style>
</head>

<body>
<ul>
<li>Phone 设备，分辨率为 600px 一下，背景颜色为红色。</li>
<li>Pad 设备，分辨率为 600px 至 900px，背景颜色为紫色。</li>
<li>PC 设备，分辨率为 900px 以上，背景颜色为蓝色。</li>
```

```
<li>iPhone 设备，背景颜色为绿色。</li>
<li>此案例仅为演示，实际应用要复杂的多。</li>
</ul>
</body>
</html>
```

## 八、讨论与思考

### 8.1 CSS 的兼容性？

- (1) CSS 3 的兼容性的处理办法有哪些？
- (2) 对于不同的设备，特别是现在 Android 的碎片化越来越严重，移动设备的分辨率越来越高的情况下，如何正确的判断媒体设备？

### 8.2 CSS 3 的前缀。

- (1) CSS 3 中四种前缀，这些前缀有哪些作用？
- (2) CSS 3 的前缀，对于标准推进是积极的推进还是灾难？