

NSD Devweb DAY02

1. [案例1：使用 CSS 样式](#)
2. [案例2：CSS 基础选择器](#)

1 案例1：使用 CSS 样式

1.1 问题

【参见COOKBOOK】

1. 使用内联样式为<p>元素设置样式
2. 使用内部样式表为<h1>元素设置样式
3. 使用外部样式表为<h2>元素设置样式

1.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：使用内联样式为<p>元素设置样式

1)样式定义在 HTML 元素的标准属性 style 里

2)CSS语法:

只需要将分号隔开的一个或者多个属性/值对作为元素的style 属性的值

属性和属性值之间用：连接

多对属性之间用；隔开

3)要使用内联样式，你需要在相关的标签内使用样式（style）属性。Style 属性可以包含任何 CSS 属性。下例展示如何改变段落的颜色和左外边距

```
01. <body>
02. <p style="color:red;margin-left:20px">段落文本（p）：使用内联样式</p>
03. </body>
```

4)实例演示如图-1所示：

段落文本（p）：使用内联样式

图-1

步骤二：使用内部样式表为<h1>元素设置样式

1)样式表规则位于文档头元素中的 <style> 元素内
文档的 <head> 元素内添加 <style> 元素

2)在 <style> 元素中添加样式规则

[Top](#)

可以定义多个样式规则

每个样式规则有两个部分：选择器和样式声明

3)选择器：决定哪些元素使用这些规则

4)样式声明：一对大括号，包含一个或者多个属性/值对

5)当单个文档需要特殊的样式时，就应该使用内部样式表。可以使用 `<style>` 标签在文档头部定义内部样式表，实例如下所示：

```
01. <html>
02. <head>
03. <style>
04.     h1 { color: blue; }
05. </style>
06. </head>
07. <body>
08. <h1>一级标题（h1）：使用内部样式表</h1>
09. </body>
10. </html>
```

实例演示如图-2所示：

一级标题（h1）：使用内部样式表

图-2

步骤三：使用外部样式表为<h2>元素设置样式

1)创建一个单独的样式表文件用来保存样式规则

一个纯文本文件，文件后缀为 .css

该文件中只能包含样式规则

样式规则由选择器和样式声明组成

```
01. h2 {
02.     color: green;
03.     background-color: silver;
04. }
```

2) 在需要使用该样式表文件的页面上，使用<link> 元素链接需要的外部样式表文件。在文档的 <head> 元素内添加 <link> 元素：[Top](#)

```
01. <html>
02. <head>
03. <link rel="stylesheet" type="text/css" href="myStyle.css" />
04. </head>
05. <body>
06. <h2>二级标题 (h2) : 使用外部样式表</h2>
07. </body>
08. </html>
```

实例演示如图-3所示：

二级标题 (h2) : 使用外部样式表

图-3

2 案例2：CSS 基础选择器

2.1 问题

【参见COOKBOOK】

1. 使用 CSS 基础选择器为页面元素设置样式

2.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：通用选择器

通用选择器，显示为一个星号（*）

可以与任何元素匹配

常用于设置一些默认样式，比如设置整个文档的文本的默认字体和大小、颜色等

```
01. <html>
02. <head>
03. <style>
04. * { color : red;font-size : 9pt;font-family : "Times New Roman";}
05. </style>
06. </head>
07. <body>
08. <h1>我的第一个标题</h1>
09. <p>我的第一个段落</p>
10. <h2>我的第二个标题</h2>
11. </body>
```

[Top](#)

12. `</html>`

实例演示如图-4所示：

我的第一个标题
我的第一个段落
我的第二个标题

图-4

步骤二：元素选择器

html 文档的元素就是选择器，比如 `<p>`、`<h1>` 等

```
01. <html>
02. <head>
03. <style>
04.     h1 { color : blue ;}
05.     h2 { color : silver ;}
06. </style>
07. </head>
08. <body>
09.     <h1>我的第一个标题</h1>
10.     <p>我的第一个段落</p>
11.     <h2>我的第二个标题</h2>
12. </body>
13. </html>
```

实例演示如图-5所示：

我的第一个标题
我的第一个段落
我的第二个标题

图-5

[Top](#)

步骤三：类选择器

1)语法为：`.className { color:red;}`

类名称不能以数字开头

2)所有能够附带class属性的元素都可以使用此样式声明

将元素的 class属性的值设置为样式类名

```
01. <html>
02. <head>
03. <style>
04. #样式表中声明一个样式类
05. .my Class
06. {
07. background-color : Pink;
08. font-size : 35pt;
09. }
10. </style>
11. </head>
12. <body>
13. #将元素的 class 属性的值设置为样式类的名称
14. <h1 class="my Class">我的第一个标题</h1>
15. <p class="my Class">我的第一个段落</p>
16. <h2>我的第二个标题</h2>
17. </body>
18. </html>
```

实例演示如图-6所示：

我的第一个标题

我的第一个段落

我的第二个标题

图-6

3)多类选择器

可以将多个类选择器应用于同一个元素

HTML 元素的 class 属性的值中可能包含一个词列表

各个词之间用空格分隔，每个词都是一个类选择器

[Top](#)

```
01. <html>
02. <head>
03. <style>
04. #样式表中声明一个样式类
05. .important {
06. font-weight : bold;
07. }
08. .warning {
09. color : red;
10. }
11. </style>
12. </head>
13. <body>
14. #将元素的 class 属性的值设置为样式类的名称
15. <h1 class="important">我的第一个标题</h1>
16. <p class="warning">我的第一个段落</p>
17. <h2>我的第二个标题</h2>
18. </body>
19. </html>
```

实例演示如图-7所示：

我的第一个标题

我的第一个段落

我的第二个标题

图-7

4)分类选择器

可以将类选择器和元素选择器结合起来使用，实现对某种元素中不同样式的细分控制

5)语法为：元素选择器.className { }

先声明一个元素选择器

然后使用一个点号（.）代表将使用类选择器

然后声明一个类的名称

最后使用一对大括号声明样式规则

6)将样式规则与附带 class 属性的某种元素匹配

元素的 class 属性的值为分类选择器中指定的样式类名

[Top](#)

```
01. <html>
02. <head>
03. <style>
04. #样式表中声明一个样式类
05. p.important
06. {
07. color : red ;
08. font-size : 20pt;
09. border: 1px solid black;
10. }
11. </style>
12. </head>
13. <body>
14. #将元素的 class 属性的值设置为样式类的名称
15. <h1 class="important">我的第一个标题</h1>
16. <p class="important">我的第一个段落</p>
17. <h2>我的第二个标题</h2>
18. </body>
19. </html>
```

注意：只有 <p> 元素使用样式，实例演示如图-8所示：

我的第一个标题

我的第一个段落

我的第二个标题

图-8

步骤四：id选择器

1)id 选择器以一种独立于文档元素的方式来指定样式

它仅作用于 id 属性的值

2)语法为：

选择器前面需要有一个 # 号

选择器本身则为文档中某个元素的 id 属性的值

```
01. <html>
02. <head>
03. <style>
04. #定义一个 id 选择器
```

[Top](#)

```
05.  #mostImportant
06.  {
07.    color:red;
08.    background:yellow;
09.  }
10.  </style>
11.  </head>
12.  <body>
13.    将元素的 id 属性的值设置为选择器的名称
14.    <h1 id="mostImportant">我的第一个标题</h1>
15.    <p>我的第一个段落</p>
16.    <h2>我的第二个标题</h2>
17.  </body>
18.  </html>
```

实例演示如图-9所示：



图-9

步骤五：群组选择器

选择器声明为以逗号隔开的选择器列表，将一些相同的规则作用于多个元素

```
01.  <html>
02.  <head>
03.    <style>
04.      h2,p.important
05.      {
06.        color:green;
07.        font-size:20pt;
08.        border:1px solid red;
09.      }
10.    </style>
11.  </head>
12.  <body>
13.    <h1>我的第一个标题</h1>
14.    <p class="important">我的第一个段落</p>
```

[Top](#)

15. `<h2>我的第二个标题</h2>`
16. `</body>`
17. `</html>`

实例演示如图-10所示：

我的第一个标题

我的第一个段落

我的第二个标题

图-10

步骤六：伪类选择器

1)伪类用于向某些选择器添加特殊的效果

2)使用冒号(:)作为结合符，结合符左边是其他选择器，右边是伪类

选择器：伪类选择器

3)链接伪类

: link ，适用于尚未访问的链接

: visited ，适用于访问过的链接

4)动态伪类，用于呈现用户操作

:hover，适用于鼠标悬停在 HTML 元素时

:active，适用于 HTML 元素被激活时

:focus，适用于 HTML 元素获取焦点时

```
01. #定义伪类选择器
02. a: link {
03.     color: black;
04.     font-size: 15pt;
05. }
06. a: visited {
07.     color: pink;
08.     font-size: 15pt;
09. }
10. a: hover {
11.     font-size : 20pt;
12. }
13. a: active {
14.     color : red;
```

[Top](#)

15. }

实例演示如图-11、图-12所示：

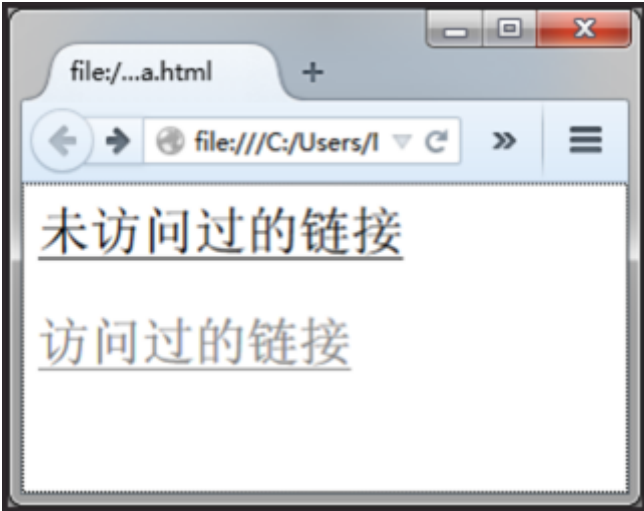


图-11



图-12

[Top](#)