



HTML5与CSS3基础知识

项目1

项目导读

在HTML技术发展之初，因为各浏览器没有统一的标准，导致它们解析网页的方式互不相同，增加了网站开发人员的负担。直到2014年，万维网联盟发布了通用的互联网标准，也就是HTML5。它具有跨平台使用的特性，为如今发达的网页制作技术奠定了坚实的基础，差不多与其同时出现的还有CSS和JavaScript等网页前端开发技术。本项目将介绍HTML5与CSS3的基础知识。



学习目标

- ◆ 了解HTML与CSS的发展历程。
- ◆ 掌握HTML5的文档结构与基本语法。
- ◆ 掌握创建HTML5文档的方法。
- ◆ 掌握在HTML5页面中添加CSS3样式的方法。
- ◆ 掌握CSS3选择器的使用方法。
- ◆ 了解CSS3的继承性与层叠性。



内容概览

任务1.1

HTML5基础

任务1.2

CSS3基础

任务1.3

CSS3中的选择器



任务1.1

HTML5基础

- 网站前端开发技术
- HTML的发展历程
- HTML5的文档结构
- HTML5基本语法
- HTML5主结构元素

1.1.1 网站前端开发技术

1. HTML

HTML全称hypertext markup language，即超文本标记语言。使用HTML编写的超文本文档称为HTML文档，它能独立于各种操作系统。浏览器能够识别并解析这些文档。



2. CSS

CSS全称cascading style sheets，即层叠样式表。它是一种能够定义网页样式的语言，可以直接写在HTML文档中或是单独存放为样式表文件。当CSS样式表单独存放时，多个页面可以使用同一个样式表文件，这样不仅可以轻松地统一多个页面的风格，同时还减轻了HTML文档的负担。

3. JavaScript

JavaScript是一种**解释性脚本语言**，简称JS。它可以收集页面上的数据，并记录用户对页面的操作等，从而使原本静态的页面具有交互性。

在网页制作中，**HTML属于网页的结构层部分，用于确定网页的结构和内容；CSS属于网页的表现层部分，用于设置网页的样式，主要包括各元素的外观、大小和位置等。**两者既相互独立又相互联系，熟练掌握它们就能制作出各种静态的网页。JavaScript在网页制作中主要起锦上添花的作用，使用它可以为网页添加更复杂的动画和交互效果。



1.1.2 HTML的发展历程

HTML在20世纪90年代发展迅速，从HTML 2.0版本发展到3.2版本、4.0版本，再到4.01版本。就在HTML 4.01版本发布的当年，W3C对HTML未来的发展做了分析，他们认为HTML的形式与内容无法分离，且标记单一，前途堪忧。于是W3C转向语言更加规范的XML，但是从1991年HTML出现到1999年的8年时间里，互联网上已经有成千上万由HTML编写的网页，直接转向XML是不现实的，所以W3C只能放慢脚步，开始HTML到XML的过渡，因此出现了XHTML。



1. 转向XHTML

万维网联盟（world wide web consortium，W3C）在2000年推出了XHTML 1.0，它以XML的语法重新整理了HTML 4.0，使其具有XML那样严谨的语法规则。

而后在2002年推出了XHTML 2.0，但由于其没有反映出网站开发人员在网页上的交互操作，也未能将网页上已有的内容向后兼容等，最终XHTML 2.0宣告失败。

2. HTML5的成功

就在XHTML 2.0失败的同时，HTML5的演化取得了成功。因为HTML5在开发时就综合考虑了当前与未来需要进行的开发任务，所以它向后兼容，且包含了HTML 4.0的全部特性，并进行了少量的修改与完善。

2014年，W3C正式宣布，HTML5制作完成并公开发布。目前HTML5已经得到主流浏览器的良好支持。

1.1.3 HTML5的文档结构

HTML5使用元素描述页面结构，元素一般由一个或两个标签（用尖括号括起来的代码）组成，由两个标签组成的元素包括一个起始标签和一个结束标签（结束标签中带有斜杠）。最基本的HTML5文档结构如下：

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8"/>
  <title>HTML5实例</title>
</head>
<body>
  <h1><i>HTML</i>5的文档结构</h1>
  <p>最基本的文档结构</p>
</body>
</html>
```



通过浏览器可以查看该文档的页面效果（如无特殊说明，本书默认使用谷歌浏览器展示页面效果），具体效果如下图1-1所示。下面介绍该文档的基本组成部分。



1. DOCTYPE声明

- ◆ DOCTYPE声明**位于**文档首行，标签为<!DOCTYPE html>，它不属于文档的结构部分。
- ◆ DOCTYPE声明**用于**向浏览器说明当前文档使用的标准规范，“<!DOCTYPE html>”即表示当前文档使用HTML5标准。



提示：如果省略DOCTYPE声明，大多数浏览器也能够正确显示文档内容，但是编写网页代码时不应依赖于浏览器的处理，需要保持良好的编写习惯，正确声明文档所使用的标准规范。

2. html元素

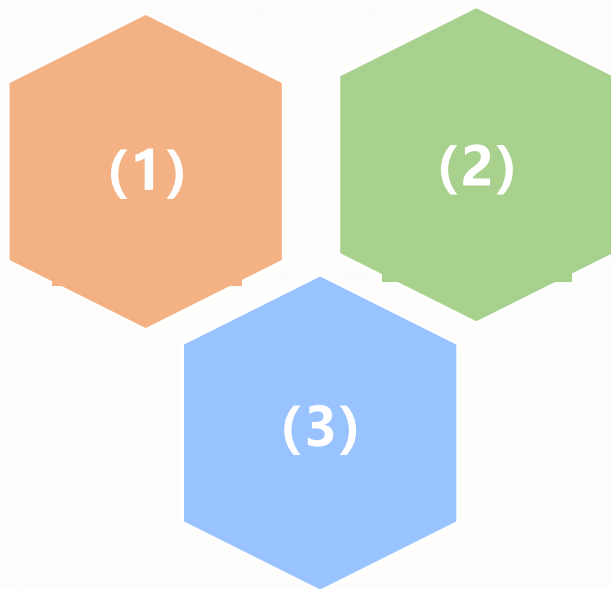
html元素是HTML5页面的根元素，**用于**告知浏览器自身是一个HTML5文档，<html>标签标志着HTML5文档的开始，</html>标签标志着HTML5文档的结束，它们之间的内容即为文档的头部结构和主体内容。



3. head元素

head元素用于标记HTML5文档的头部结构，也称为头部标记，它跟在<html>标签之后，**用于**封装其他位于文档头部的标签，这些标签中存储着网页的各种基本信息，可以使浏览器快速解析页面，其中的内容一般不会显示在页面中。位于头部的元素一般包括以下几个。

网页标题。使用<title></title>标签可以标记网页的标题，浏览器会将标签中的内容显示在标题栏或状态栏中。



网页元信息。使用<meta />标签可以标记网页元信息，一般网站都会在其中设置描述和关键词等属性，以便搜索引擎检索。它还可以标记文档的字符编码，如上文的代码“charset="utf-8"”即表示将网页文档的字符编码设置为“utf-8”。

网页视口。网页的视口就是浏览器中显示网页页面的那部分区域，通过设置视口的宽度、缩放比例等属性可以使网页适配移动端。设置视口使用“<meta name="viewport">”标签，具体内容见9.2.1节。

提示：每一个页面中都需要含有 <title> </title> 标签，且该标签的内容不具有任何格式。网页标题是搜索引擎识别网页内容的标识，它可以影响网页在搜索引擎中的排名，一个表达明确的网页标题可以为网站带来更大的流量。



4. body元素

body元素**用于**标记HTML5文档所要显示的内容，也称为主体标记。页面中的所有文本、图像、音频或视频等信息都必须写在<body>标签之内。<body>标签中的信息才是最终展示给用户的内容。



知识库

html、head和body这3种元素在一个HTML5文档中**只能各出现一次**，**且<head>和<body>标签必须在<html>标签之内**，<body>标签位于<head>标签之后，与<head>标签并列。

1.1.4 HTML5基本语法

HTML5是使用标签来描述网页的，HTML5文档中实质上只包含HTML标签和文本。浏览器的一个重要作用就是读取HTML5文档，并解释其中的一个个标签，然后将其以“网页”的形式显示。

1. HTML5标签的分类

1.1.3节中提到的<html>、<head>与<body>等都是HTML5标签，也称为HTML5标记。“标签”和“元素”这两个概念通常不做区分，但是一般来说，一个元素包括其对应的标签及其中的内容。HTML5的标签主要分为以下几类。



(1) 双标签是指由开始标签和结束标签组成的标签，基本语法格式为：

```
<标签名>内容</标签名>
```

例如，`<title>...</title>`就是一个双标签。

提示：标签不区分大小写。例如，“`<P>标签的大小写</p>`”在HTML5中是合法的。但是一般不建议这么写，而是统一小写。

(2) 单标签也称为空标签，基本语法格式为：

```
<标签名 />
```

例如，`<hr />`就是一个单标签，用于标记一条水平线。单标签中的“/”可以省略。

(3) 注释标签是HTML5中一种特殊的标签。使用注释标签括起来的内容不会显示在页面上，但是会保存在页面的源代码中。它的基本语法格式为：

```
<!--注释内容-->
```

在各功能模块的开头和结尾处添加注释可方便合作开发的人员修改和理解代码。



2. HTML5标签的属性

在HTML5中，可以根据需要为标签设置属性，比如在段落标签中设置该段落的文本颜色或字号大小等。设置标签属性的语法格式为：

```
<标签名 属性1=“属性值1” 属性2=“属性值2”>内容</标签名>
```

一个标签可以有多个属性，标签名和属性名之间要用空格隔开，属性之间不分先后顺序。标签的属性省略时将使用它们的默认值。





拓展阅读

(1) 属性值可省略引号。在HTML5中，标签的属性值两侧既可以用双引号，也可以用单引号，当属性值不包含特殊字符的时候，引号可以省略。例如，以下3行代码在HTML5中都是合法的。

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

```
<meta charset='utf-8'/>
```

```
<meta charset=utf-8/>
```

(2) 部分属性的属性名可以省略。例如：

```
<input checked="checked" type="checkbox">
```

```
<input checked type="checkbox">
```

上述代码是等价的，可以省略属性名的属性还有readonly、defer与ismap等。

3. 块级元素与行内元素

body元素中的元素分为块级元素和行内元素。

◀ (1) 块级元素在屏幕上显示时类似于首尾各有一个换行符。例如，图1-1中显示的两行文本内容，标记它们的h1与p元素都是块级元素，它们在新的行中显示内容，这些元素之后的内容同样也会另起一行。

(2) 行内元素是可以出现在同一行中的元素。例如，图1-1中显示的第一行文本内容，其中标记斜体的i元素就是一个行内元素。 ▶



知识库

- 行内元素中不可以包含块级元素，它只能位于块级元素之内。
- 块级元素既可以包含其他块级元素，也可以包含行内元素。

1.1.5 HTML5主结构元素

HTML5中包含上百个标签。例如，1.1.3节代码中的<h1>是文本类的标题标签，<p>是文本类的段落标签，这些标签主要放置在body元素内，具体应用将在后面的项目中分别讲解。在设计页面时如果任意堆叠各种标签，会导致页面的整体结构变得散乱，不便于后期添加样式。所以，为了方便设计页面结构，在对页面排版时需要将它们放在下列表示结构的元素中。

(1) 页眉。

表示页眉的元素是header，它用于标记整个页面的标题栏或一个内容块的标题区域。一个页面可以包含多个header元素，它可以包含网站Logo、主导航或搜索框等。

(2) 导航。

表示导航的元素是nav，它用来标记页面导航的链接组。一个页面可以包含多个nav元素，导航性质的链接都可以放置在nav元素中，比如主菜单链接、侧边栏导航或页内导航等。页脚的导航可以不放在该元素中。

(3) 主要区域。

表示主要区域的元素是main，它用于标记一个页面的主要内容。一个页面中只能包含一个main元素。main元素不能被放在header、nav、article、aside或footer元素中，在<main>标签中设置role属性的值为main可以帮助阅读器定位页面的主要区域。

(4) 文章块。

表示导航的表示文章块的元素是article，它用来标记页面中的一个独立内容，比如报纸文章、博客条目或用户评论等。一个页面可以包含多个article元素，在article元素中也可以包含一个或多个article元素。元素是nav，它用来标记页面导航的连接组。一个页面可以包含多个nav元素，导航性质的链接都可以放置在nav元素中，比如主菜单链接、侧边栏导航或页内导航等。页脚的导航可以不放在该元素中。

(5) 区块。

表示区块的元素是section，它用来标记文档中的节，对内容进行分区，比如文档中的章节、页眉、页脚等，可以将其嵌套在article元素中。section元素标记的是页面中的特定区域，可对其中的内容进行操作，如存储到数据库里或输出到Word文档中。

(6) 附栏。

表示附栏的元素是`aside`，它用来标记一个和其余页面内容几乎无关的部分，将它单独拆分出来后不会影响整体内容。例如，它可以包含与当前页面或主要内容相关的引用、侧边广告等。

(7) 页脚。

表示页脚的元素是`footer`，它用来标记文档或节的页脚，如版权信息、作者信息等。当`footer`元素作为整个文档即`body`元素的页脚时，一般位于页面底部。

除上述结构元素之外，HTML5中还有两个无语义容器，分别是**div元素**和**span元素**。

- `div`元素是块级的无语义容器；
 - `span`元素是行内的无语义容器，
- 灵活使用它们可以制作出更具结构化的页面样式。

创建第一个HTML5文档

任务实施

1. 使用记事本创建HTML5文档
2. 使用Dreamweaver CC 2020创建HTML5文档





任务1.2

CSS3基础

- CSS3的发展历程
- CSS3的语法规则
- 在HTML5页面中添加CSS3样式

1.2.1 CSS3的发展历程

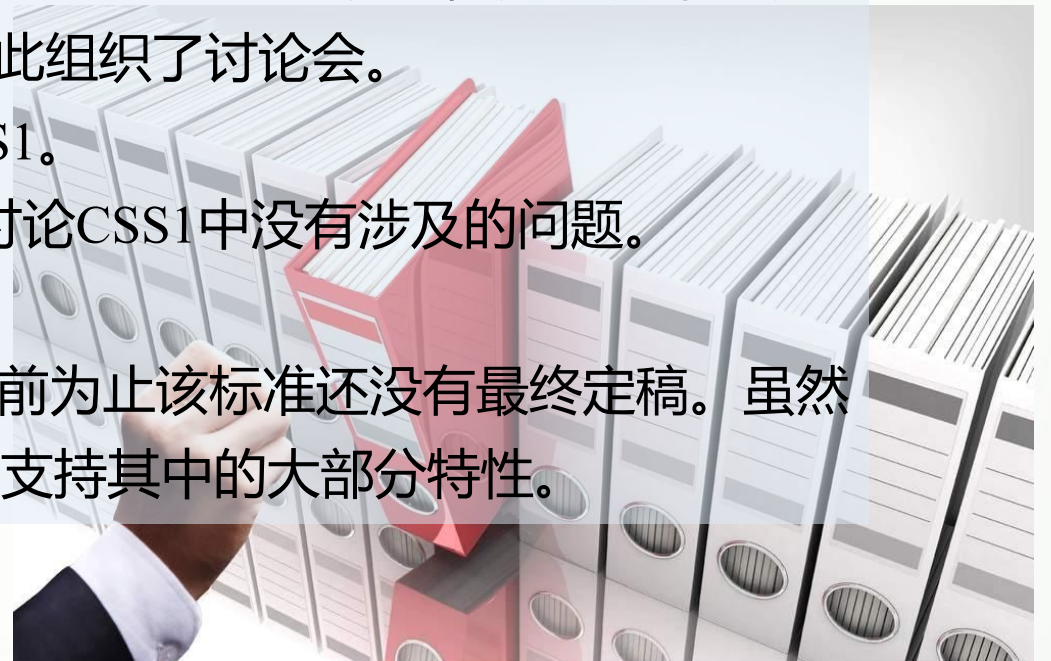
1994年，哈坤·利提出了CSS的最初设想，并和伯特·波斯一拍即合，决定共同开发CSS。在这一年的年底，哈坤在芝加哥的一次会议上展示了CSS的设想，并于次年再次向业内展示。当时刚成立的W3C对此很感兴趣，为此组织了讨论会。

1996年12月，W3C发布了CSS的第一个版本CSS1。

1997年，W3C组织专门负责CSS的工作组开始讨论CSS1中没有涉及的问题。

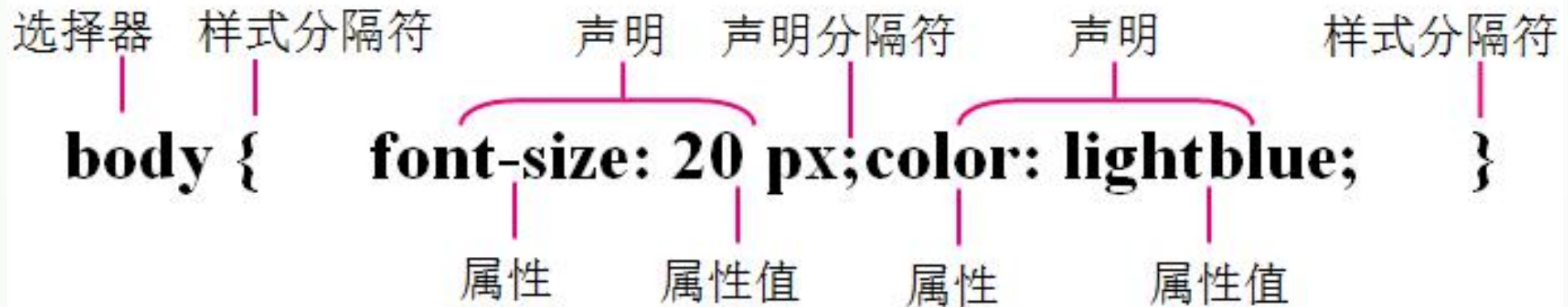
1998年5月，W3C发布了CSS2。

2005年12月，W3C开始制定CSS3的标准，到目前为止该标准还没有最终定稿。虽然完整的CSS3标准还未发布，但主流浏览器已经开始支持其中的大部分特性。



1.2.2 CSS3的语法规则

CSS3的语法规则是由选择器和声明组成的，通过选择器匹配HTML文档中的不同元素，并对它们应用声明中设置的样式属性。CSS3的语法规则如下图所示1-16所示。



下面分别介绍各组成部分。

(1) 选择器用于匹配应用声明的某个或某些元素。

(2) 声明可以有若干个，声明之间用分号隔开并放置在一对大括号中，它们命令浏览器如何渲染指定对象。声明由属性和属性值两部分组成。

(3) 属性是CSS3提供的样式选项，属性名一般由一个或多个单词组成，多个单词中间用连字符连接。

(4) 属性值是设置属性效果的参数，可以是数值、单位或关键字。

图1-16中的CSS3样式表示将body元素中的字号大小设置为20 px，文本颜色设置为浅蓝色。

1.2.3 在HTML5页面中添加CSS3样式

1. 行内样式表

行内样式也称为内联样式，是直接为HTML标签设置style属性，具体格式为：

```
<标签名 style="属性1:属性值1;属性2:属性值2;.....">内容</标签名>
```

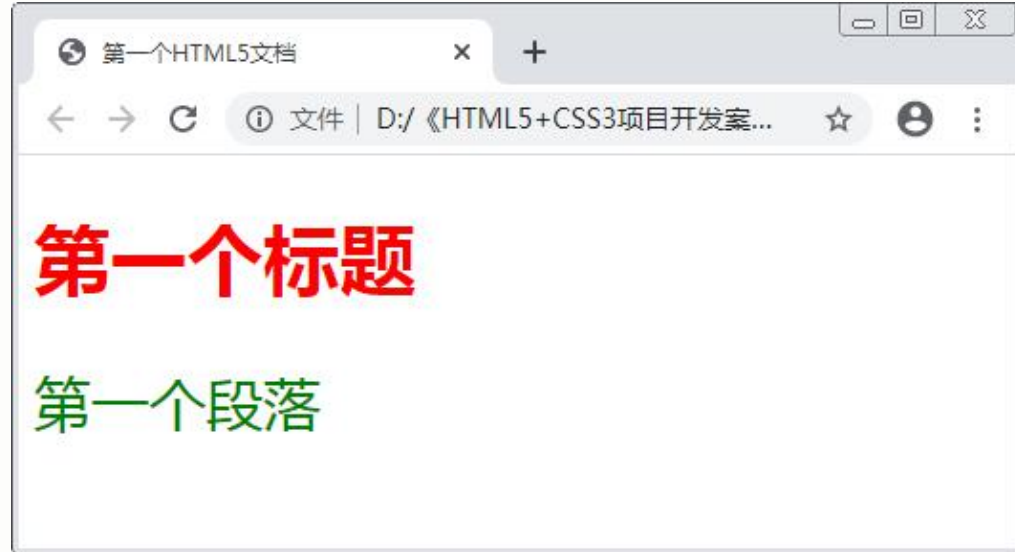
其中，style是标签的属性，它自身又包括了多个属性，这些属性与属性值的规范与CSS3样式规则相同。行内样式只对其所在的标签起作用。



【例1-1】 为任务1.1中创建的“1-1.html”文档添加行内样式，设置标题、段落标签的字号与文本颜色。代码如下：

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>第一个HTML5文档</title>
  </head>
  <body>
    <h1 style="font-size:40px;color:red;">第一个标题</h1>
    <p style="font-size:30px;color:green;">第一个段落</p>
  </body>
</html>
```


上述代码的页面效果如下图1-17所示。



行内样式虽然看起来简单，但其中也存在着一些缺陷比如每一个标签在设置样式时都需要添加style属性，增加的代码会使页面体积变大，并没有解决HTML5页面的臃肿问题。

2. 内嵌样式表

内嵌样式是将CSS3代码写在HTML5文档的头部标签中，并且用<style>标签标记，具体格式为：

```
<head>  
  <style type="text/css">  
    {属性1:属性值1;属性2:属性值2;}  
  </style>  
</head>
```

其中，type属性值为“text/css”表示<style>标签所包含的是样式代码，该属性只有这一个属性值，在HTML5中可以将其省略。

提示：在该格式中，<style>标签一般位于<head>标签中的<title>标签之后，虽然实际上它可以写在文档的任意位置，但由于浏览器是从上到下解析代码的，所以一般会将样式代码写在头部，便于浏览器最先识别并解析。

【例1-2】 使用内嵌样式为 “1-1.html”文档设置样式。代码如下：

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>第一个HTML5文档</title>
    <style>
      h1{font-size:40px;color:red;}
      p{font-size:30px;color:green;}
    </style>
  </head>
  <body>
    <h1>第一个标题</h1>
    <p>第一个段落</p>
  </body>
</html>
```

内嵌样式相对于行内样式的优势是可以作用于其所在的整个页面，但是当需要设计一个网站时，使用内嵌样式仍然不能发挥出CSS3的最大优势。

3. 链接样式表

链接样式是将所有样式放在一个或多个扩展名为 “.css” 的外部样式表文件中，然后使用<link />标签将样式表文件链接至HTML5文档中，具体格式为：

```
<link href="CSS3文档位置" type="text/css" rel="stylesheet"/>
```

其中，<link />标签需放在<head>标签中，href属性指定CSS文档的位置，type与rel属性用于表明连接是一个CSS3样式表文件。



【例1-3】 使用链接样式为 “1-1.html”文档设置样式。

使用链接样式时需要先创建一个CSS3文档，这里介绍使用记事本创建CSS3文档的方法。

步骤1

在 “1-1.html”文档的存储目录下新建一个文本文档，将其重命名为 “1-1.css”。

步骤2

将 “1-1.css”文档使用记事本打开，在编辑窗口中输入以下代码，然后保存文档。

```
h1{font-size:40px;color:red;}  
p{font-size:30px;color:green;}
```

步骤3

在“1-1.html”文档的头部标签中添加以下代码。

```
<link href="1-1.css" type="text/css" rel="stylesheet"/>
```

步骤4

保存“1-1.html”文档，在浏览器中查看它的页面效果。



为“1-2.html”文档设置样式

任务实施

使用DW创建样式表文件，并将其链接至HTML5文档的方法。



扫一扫



任务1.3

CSS3中的选择器

- 基本选择器
- 组合选择器
- 属性选择器
- 伪类选择器
- CSS3的继承性和层叠性

1.3.1 基本选择器

1. 标签选择器

HTML文档是由很多标签组成的，标签选择器就是直接引用标签名的选择器。使用标签选择器可以统一设置使用某类标签设置的元素的样式。例如，以下代码使用了body标签选择器。

```
body{font-size:20px;color:lightblue;}
```

标签选择器是最常用的选择器，通常用它来统一设置某些元素的基本样式。



知识库

CSS3中有一个特殊的选择器——通配选择器，它使用“*”表示，用于匹配所有标签。例如，以下代码表示重置浏览器的部分默认样式。

```
*{margin:0;padding:0;border:0;list-style:0;}
```



2. 类选择器

类选择器以 “.”为前缀，跟随一个自定义类名。**类选择器可以使不同的元素拥有相同的样式，也可以使相同的元素拥有不同的样式。**

类选择器可以通过设置class属性实现，就是说在定义好选择器后，还需要为标签设置class属性值（class属性可以包含类列表，多个属性值之间用空格隔开）为类选择器的名称，这样才能将类选择器样式应用于标签。此外，还可以把标签与类捆绑在一起，来限定类的适用范围，这种限定方式称为指定类选择器。



知识库

为了能够识别各种标签，可以设置标签的class或id属性来赋予它们额外的语义。

- **class属性可以应用于页面中任意数量的元素，适合用来标识样式相同的对象。**
- **id属性用来标识页面上固定存在的结构性元素，或者区分语义相近的元素，它的属性值必须是唯一的。**

【例1-4】 使用类选择器设置样式，页面效果如下图1-21所示。

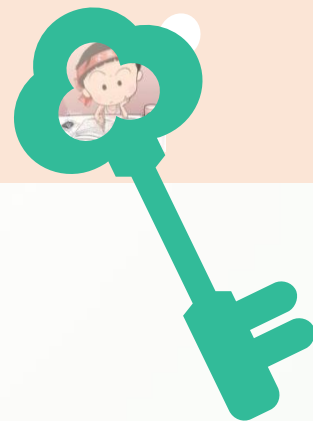


步骤1



创建HTML5文档，在<body>标签中输入以下代码，构建诗文内容的结构。

```
<h1 align="center" class="p1">无题</h1>  
<p>相见时难别亦难，东风无力百花残。</p>  
<p class="p1">春蚕到死丝方尽，蜡炬成灰泪始干。</p>  
<p class="p2">晓镜但愁云鬓改，夜吟应觉月光寒。</p>  
<p class="p2 b">蓬山此去无多路，青鸟殷勤为探看。</p>
```

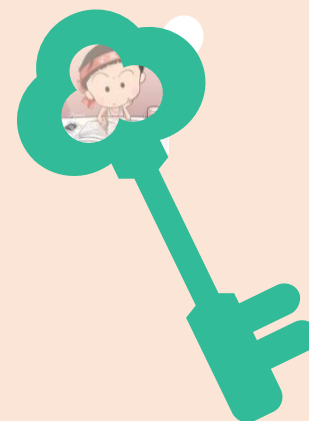


步骤2



在<head>标签中添加<style>标签，在其中输入以下代码，使用类选择器为各p元素设置不同的样式。

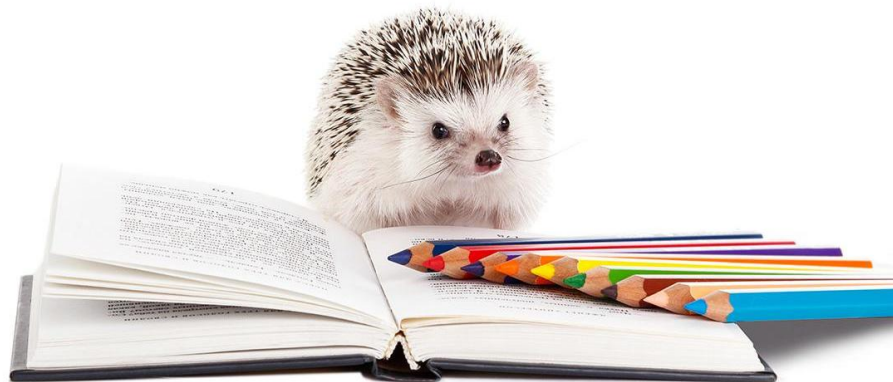
```
p{text-align:center;font-size:18px;}
/*设置标签选择器p中文本居中对齐，字号为18 px*/
.b{font-weight:bold;}
/*设置类选择器b的字体加粗显示*/
.p1{font-family:"楷体";font-size:30px;}
/*设置类选择器p1的字体为楷体，字号为30 px*/
p.p1{text-decoration-line:underline;font-size:20px;}
/*设置标签选择器p的类p1中，文本增加下划线，字号为20 px*/
.p2{font-family:"宋体";font-size:22px;}
/*设置类选择器p2的字体为宋体，字号为22 px*/
```



3. ID选择器

ID选择器以“#”为前缀，跟随一个自定义的ID名。ID选择器可以通过设置id属性实现，就是在定义好ID选择器后，需要设置标签的id属性值为ID选择器的名称，以将ID选择器应用于标签。

ID选择器的使用方式与类选择器基本相同，区别在于ID选择器只能应用于HTML文档中的一个元素，而类选择器可以应用于多个元素。



1.3.2 组合选择器

1. 包含选择器

包含选择器通过空格连接两个选择器，前面选择器表示包含的祖先元素，后面选择器表示被包含的后代元素。



2. 类选择器

子选择器通过 “>”连接两个选择器，前面选择器表示要匹配的父元素，后面选择器表示被包含的子元素。

【例1-5】 使用包含与子选择器设置元素样式，页面效果如下图1-22所示。



步骤1



创建HTML5文档，在<body>标签中输入以下代码，构建文本内容的结构。

```
<h1 align="center"><strong>重</strong><i>中之</i><strong>重</strong></i></h1>
<h2 align="center"><strong>重</strong><i>中之</i><strong>重</strong></i></h1>
```

步骤2



在<head>标签中添加<style>标签，在其中输入以下代码，分别使用包含与子选择器设置strong元素的样式。

```
h1,h2{font-size:30px;}
h1 strong{font-size:40px;}
h2>strong{font-size:40px;}
```

```
/*使用包含选择器*/
/*使用子选择器*/
```

提示：多个选择器之间用英文逗号隔开时，称为选择器的集体声明。当多个选择器设置的样式完全相同时，可以将这些选择器归为一组，并用英文逗号隔开以进行集体声明，从而提高代码的效率和速度，同时也可以降低代码的冗余。



高手点拨

相对于子选择器而言，包含选择器的作用范围更广，在祖先元素之下的所有后代元素都将受影响。子选择器只能影响父元素的下一级中的元素。在例1-5中，应用了包含选择器的h1元素中，所有strong元素都受影响。而应用了子选择器的h2元素中，包含在i元素中的strong元素没有受影响，因为它不在h2元素的下一级中。

3. 相邻选择器

相邻选择器通过 “+” 连接两个选择器，只有满足这个连接顺序的、同级的、相邻的元素才会成功匹配。



4. 兄弟选择器

兄弟选择器通过 “~” 连接两个选择器，它在相邻选择器的基础上，通过连接顺序匹配相邻的元素之后，会将后续同级别的该类型元素一同匹配。



高手点拨

相邻选择器与兄弟选择器的区别在于，相邻选择器会严格按照连接顺序匹配相邻的元素，成功匹配后不会匹配该元素的后续元素，而是继续检索页面对象，直到再次出现符合连接顺序的元素，才会再次匹配相邻元素。

【例1-6】 使用相邻与兄弟选择器设置元素样式，页面效果如图1-23所示。



创建HTML5文档，参照以下代码段分别在<style>和<body>标签中输入代码，分别使用相邻与兄弟选择器设置p元素的样式。

```
<style>
  h1+p{font-size:20px;}
  hr~p{font-size:20px;}
</style>
.....
<body>
  <h1>无题</h1>
  <p>相见时难别亦难，东风无力百花残。 </p>
  <p>春蚕到死丝方尽，蜡炬成灰泪始干。 </p>
  <hr />
  <p>晓镜但愁云鬓改，夜吟应觉月光寒。 </p>
  <p>蓬山此去无多路，青鸟殷勤为探看。 </p>
</body>
```

/*使用相邻选择器*/
/*使用兄弟选择器*/

<!--只匹配一次p元素-->

<!--匹配同级别所有p元素-->

1.3.3 属性选择器

属性选择器根据标签的属性来匹配元素。CSS3中共有7种属性选择器，为便于后面讲解，此处先创建一个简单的HTML页面，本节中后面使用的所有选择器都将针对该页面进行操作。

【例1-7】 创建一个简单的导航示例，页面效果如下图1-24所示。



步骤1



创建HTML5文档，在<body>标签中输入以下代码，构建导航示例的结构。

```
<nav align="center">  
  <a href="#" title="w3c" id="first">1</a>  
  <a href="#" title="test web" lang="zh">2</a>  
  <a href="#" title="this is a link" lang="zh-cn">3</a>  
  <a href="#" lang="zh-tw">4</a>  
  <a href="#" title="zh-cn">5</a>  
  <a href="#" title="web link" lang="zh">6</a>  
  <a href="#" title="open the web" lang="cn">7</a>  
  <a href="#" title="close the web" lang="en-zh">8</a>  
  <a href="#" title="http://www.baidu.com">9</a>  
  <a href="#" id="last">10</a>  
  <!--标记10个超链接标签-->  
</nav>
```


步骤2



在<head>标签中添加<style>标签，在其中输入以下代码，设置导航示例的样式。

```
nav a{border:2px solid darkslategray;font-size:30px;  
display:inline-block;height:35px;line-height:30px;width:40px;  
-moz-border-radius:20px;border-radius:20px;text-align:center;  
background:darkslategrey;color:#fff;margin-top:1px;  
margin-right:5px;text-decoration:none;}  
/*设置超链接的样式*/
```



1. E[attr]选择器

E[attr]选择器是最基本的属性选择器，用于匹配所有拥有attr属性的E元素，无论attr属性值是什么。例如：

```
nav a[id]{border:2px solid #2f4f4f;background:white;  
color:cadetblue;}
```

上述代码表示匹配nav元素中所有拥有id属性的a元素，修改其边框、背景颜色与文本颜色，页面效果如下图1-25所示。E[attr]选择器不仅能够匹配单一属性，还可以匹配多个属性，如E[attr1][attr2]。



2. E[attr="value"]选择器

E[attr="value"]选择器用于匹配attr属性值为“value”的E元素，它缩小了匹配范围，能够更加精确地匹配需要的元素。例如：

```
nav a[id="first"]{border:2px solid #2f4f4f;background:white;
color:cadetblue;}
```

上述代码精确地匹配了nav元素中id属性值为“first”的a元素，修改其边框、背景颜色与文本颜色，页面效果如下图1-26所示。与E[attr]选择器相同，它也可以匹配多个属性，如E[attr1="value1"][attr2="value2"]。



3. E[attr~="value"]选择器

E[attr~="value"]选择器用于匹配attr属性值列表中包含了“value”的E元素，不需要完全匹配。如果“value”是一个列表，需要用空格隔开。例如：

```
nav a[title~="web"]{background:lightcyan;color:#2f4f4f;}
```

上述代码表示匹配nav元素中所有title属性值列表中包含了“web”的a元素，修改其背景颜色与文本颜色，页面效果如下图1-27所示。



4. E[attr^="value"]选择器

E[attr^="value"]选择器用于匹配attr属性值以“value”开头的E元素，例如：

```
nav a[title^=http]{background:#ffffff;color:#2f4f4f;}
```

上述代码表示匹配nav元素中title属性值以“http”开头的a元素，修改其背景颜色与文本颜色，页面效果如下图1-28所示。

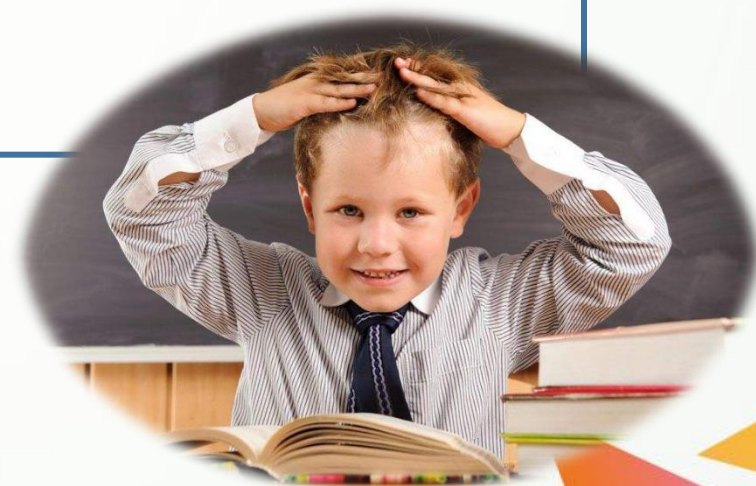


5. E[attr\$="value"]选择器

E[attr\$="value"]选择器与E[attr^="value"]选择器相反，用于匹配attr属性值以“value”结尾的E元素，例如：

```
nav a[title$="link"]{background:lightcyan;color:#2f4f4f;}
```

上述代码表示匹配nav元素中title属性值以“link”结尾的a元素，修改其背景颜色与文本颜色。



6. E[attr*="value"]选择器

E[attr*="value"]选择器用于匹配attr属性值中包含字符串“value”的E元素，无论“value”在属性值的什么位置。例如：

```
nav a[title*="t"]{background:lightcyan;color:#2f4f4f;}
```

上述代码表示匹配nav元素中title属性值含有“t”的a元素，修改其背景颜色与文本颜色，页面效果如左图1-29所示。



提示： E[attr*="value"]选择器与 E[attr~="value"]选择器不同： E[attr~="value"]选择器匹配的值需要包含在值列表中并用空格隔开，而E[attr*="value"]选择器没有这个限制，“value”字符串出现在属性值的任意位置都能够匹配成功。

7. E[attr|"value"]选择器

E[attr|"value"]选择器用于匹配attr属性值为“value”或以“value”开头的元素。例如：

```
nav a[lang|"zh"]{background:#ffffff;color:darkolivegreen;}
```

上述代码表示匹配nav元素中lang属性值为“zh”或是以“zh”开头的a元素，修改其背景颜色与文本颜色，页面效果如下图1-30所示。



1.3.4 伪类选择器

伪类是同一个标签的不同状态，所以普通的伪类选择器可以分为**静态伪类选择器**和**动态伪类选择器**，除此之外还有**结构伪类选择器**、**状态伪类选择器等**其他种类。

伪类选择器以冒号为前缀，跟随伪类或伪类对象，在冒号前可以使用其他选择器限制伪类应用的范围。



1. 静态伪类选择器

静态伪类选择器只用于设置链接的样式，包括“:link”和“:visited”，它们分别表示链接被访问前的状态与链接被访问后的状态。

【例1-8】 静态伪类选择器的应用。

创建HTML5文档，参照以下代码段分别在<style>和<body>标签中输入代码，设置链接访问前的文本颜色为黑色，访问后的文本颜色为米黄色。

```
<style>
  a:link{color:black;}
  a:visited{color:beige;}
</style>
.....
<body>
  <a href="#"> 《HTML5+CSS3项目开发案例教程》 微课视频</a>
</body>
```



2. 动态伪类选择器

动态伪类选择器可用于所有元素，包括“:hover”“:active”和“:focus”等，这3种伪类选择器分别匹配鼠标指针移动至标签上时、鼠标指针单击标签区域但不松开时，以及某个标签获得焦点时的样式。

【例1-9】 动态伪类选择器的应用。

创建HTML5文档，参照以下代码段分别在<style>和<body>标签中输入代码，设置鼠标指针移至链接上方时的背景颜色为浅蓝色，单击链接时的背景颜色为米黄色。

```
<style>
  a:hover{background:lightblue;}
  a:active{color:beige;}
</style>
.....
<body>
  <a href="#">《HTML5+CSS3项目开发案例教程》微课视频</a>
</body>
```



3. 结构伪类选择器

结构伪类选择器可以根据文档的结构来匹配元素，主要包括以下几种。

- (1) **:first-child**。匹配父元素的第一个子元素。
- (2) **:last-child**。匹配父元素的最后一个子元素。
- (3) **:nth-child()**。按正序匹配父元素中的某个元素。
- (4) **:nth-last-child()**。按倒序匹配父元素中的某个元素。
- (5) **:nth-of-type()**。按正序匹配父元素下同种标签的某个元素。
- (6) **:nth-last-of-type()**。按倒序匹配父元素下同种标签的某个元素。
- (7) **:first-of-type**。匹配父元素下同种标签的第一个元素。
- (8) **:last-of-type**。匹配父元素下同种标签的最后一个元素。
- (9) **:only-child**。匹配父元素下没有兄弟元素的元素。
- (10) **:empty**。匹配没有子元素的元素。
- (11) **:root**。匹配根元素。



【例1-10】 结构伪类选择器的应用，页面效果如下图1-31所示。



步骤1



创建HTML5文档，在<body>标签中输入以下代码，构建诗文内容的结构。

```
<p>无题</p>  
<p>相见时难别亦难，东风无力百花残。 </p>  
<p>春蚕到死丝方尽，蜡炬成灰泪始干。 </p>  
<p>晓镜但愁云鬓改，夜吟应觉月光寒。 </p>  
<p>蓬山此去无多路，青鸟殷勤为探看。 </p>
```

步骤2



在<head>标签中添加<style>标签，在其中输入以下代码，使用结构伪类选择器依次设置文档中各p元素的字号。

```
p{text-align:center;}  
p:first-child{font-size:30px;}  
p:nth-child(2){font-size:15px;}  
p:nth-child(3){font-size:18px;}  
p:nth-child(4){font-size:20px;}  
p:nth-child(5){font-size:25px;}
```



4. 否定伪类选择器

否定伪类选择器是 “:not()”，它能够过滤掉括号内匹配的元素。

【例1-11】 否定伪类选择器的应用，页面效果如下图1-32所示。



步骤1



创建HTML5文档，在<body>标签中输入以下代码，构建诗文内容的结构。

```
<p>卜算子?咏梅</p>  
<p class="p1">陆游</p>  
<p>驿外断桥边，寂寞开无主。已是黄昏独自愁，更着风和雨。 </p>  
<p>无意苦争春，一任群芳妒。零落成泥碾作尘，只有香如故。 </p>
```

步骤2



在<head>标签中添加<style>标签，在其中输入以下代码，使用否定伪类选择器为除class属性值为“p1”的p元素外的其他元素设置字体加粗。

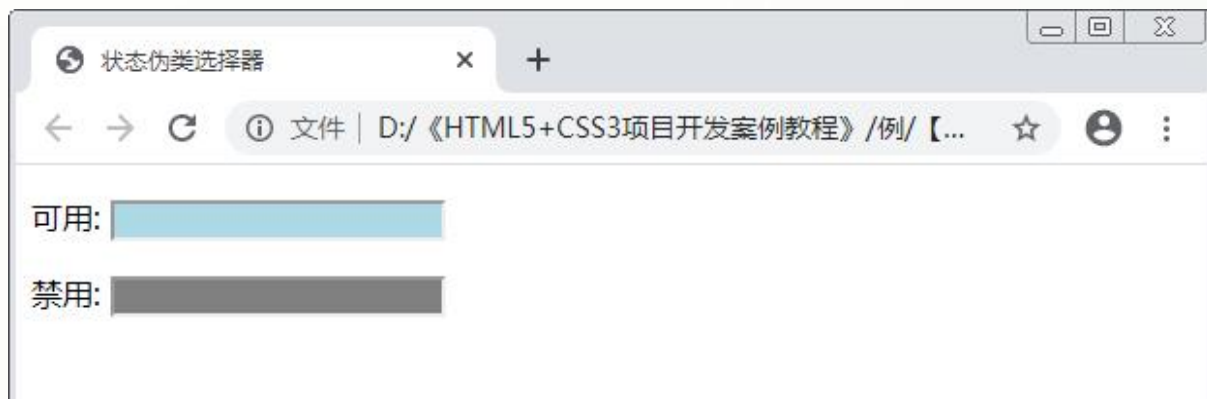
```
p{text-align:center;font-size:18px;}  
p:not(.p1){font-weight:bold;}
```


5. 状态伪类选择器

CSS3中包含十几种状态伪类选择器，常用的有如下3种。

- (1) **:enabled**。匹配指定范围内所有可用的UI元素。
- (2) **:disabled**。匹配指定范围内所有不可用的UI元素。
- (3) **:checked**。匹配指定范围内所有选择类的UI元素。

【例1-12】 状态伪类选择器的应用，页面效果如下图1-33所示。



拓展阅读

UI元素是指与用户交互相关的元素，一般指表单对象，如文本框、复选框等。

步骤1



创建HTML5文档，在<body>标签中输入以下代码，构建表单对象的结构。

```
<p>  
  <label for="enabled">可用:</label>  
  <input type="text" name="enabled" />  
</p>  
<p>  
  <label for="disabled">禁用:</label>  
  <input type="text" name="disabled"  
  disabled="disabled"/>  
</p>
```



步骤2



在<head>标签中添加<style>标签，在其中输入以下代码，使用状态伪类选择器将可用的文本框的背景颜色设置为浅蓝色，不可用的文本框的背景颜色为灰色。

```
input[type="text"]:enabled{background:lightblue;}  
input[type="text"]:disabled{background:gray;}
```



1.3.5 CSS3的继承性和层叠性

1. 继承性

继承性是指在HTML结构中，后代元素会继承祖先元素的CSS3样式。能够被继承的属性包括字号、文本颜色等，不能被继承的属性包括边框、背景等。例如，设置body元素的字号为20 px，那么在网页中除了单独设置过样式的元素外，其他元素都将继承该属性。

2. 层叠性

层叠性是指可以为同一个对象应用多个样式。如果同一个对象拥有多个样式，那么这些样式会根据各自的权重来确定呈现的优先级，然后显示最终效果。

各类选择器的权重值如下。

- (1) ID选择器：100。
- (2) 类、伪类与属性选择器：10。
- (3) 标签选择器：1。
- (4) 通配选择器：0。



除此之外，行内样式表的优先级是最高的，权重值为1000。继承样式的优先级是最低的，没有权重值。当权重值相同时，遵循就近原则。下面通过一个案例来说明权重值在网页中的应用。

【例1-13】 权重值的计算。

创建HTML5文档，参照以下代码段分别在<style>和<body>标签中输入代码，为p元素设置两种样式，然后计算它们的权重值。

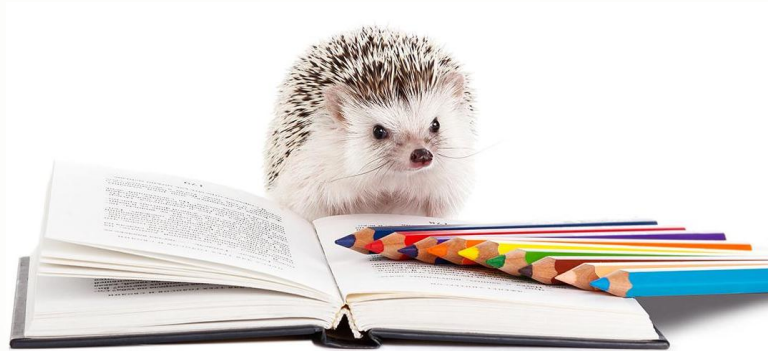
```
<style>
  #divid p{color:red;}           /*权重值为100+1=101*/
  .divclass .pclass{color:green;} /*权重值为10+10=20*/
</style>
.....
<body>
  <div class="divclass" id="divid">
    <p class="pclass">权重值的计算</p>
  </div><!--权重值计算：101>20，故文本颜色显示为红色-->
</body>
```





拓展阅读

CSS3中含有一个特殊的声明“!important”，写在样式代码的最后，用于提高指定样式规则的优先级，它的权重值为正无穷。



任务实施



扫一扫

使用选择器设置CSS3样式