

第5章 CSS基础

赵志立

兰州大学信息学院计算机系统结构所

zhaozhl@lzu.edu.cn

在设计和制作网页时，既要准备文本、图片等素材，又要考虑这些素材以何种视觉效果（如文本的字体、颜色、大小和对齐方式等）展示在网页上。而层叠样式表（Cascading Style Sheets, CSS）能够很好地解决这一问题，并提高网页的设计和制作效率。

- 相对于传统HTML的表现而言，CSS能够对网页中的对象的位置排版进行像素级的精确控制，支持几乎所有的字体字号样式，拥有对网页对象和模型样式编辑的能力，是目前基于文本展示最优秀的表现设计语言。
- CSS是能够真正做到网页表现与内容分离的一种样式设计语言。

- 网页的内容、结构和表现
- HTML规范及文档类型定义
- 内部样式表
- 常用的样式
- 行内样式
- 外部样式表及其应用
- 层叠样式表
 - CSS的层叠性
 - CSS的继承性

5.1 网页的内容、结构与表现



兰州大学

1. 内容

内容 (Content) 就是通过网页展示的信息，包含文本或者图片等。其中，**文本是网页中最常见的内容**。

静夜思唐·李白

床前明月光，疑是地上霜。举头望明月，低头思故乡。

这是写远客思乡之情的诗，诗以明白如话的语言雕琢出明静醉人的秋夜的意境。它不追求想象的新颖奇特，也摒弃了辞藻的精工华美；它以清新朴素的笔触，抒写了丰富深曲的内容。境是境，情是情，那么逼真，那么动人，百读不厌，耐人寻绎。无怪乎有人赞它是“妙绝古今”。

5.1 网页的内容、结构与表现



兰州大学

2. 结构

标题 静夜思

作者 唐·李白

正文 床前明月光，疑是地上霜。举头望明月，低头思故乡。

简析 这是写远客思乡之情的诗，诗以明白如话的语言雕琢出明静醉人的秋夜的意境。它不追求想象的新颖奇特，也摒弃了辞藻的精工华美；它以清新朴素的笔触，抒写了丰富深曲的内容。境是境，情是情，那么逼真，那么动人，百读不厌，耐人寻绎。无怪乎有人赞它是“妙绝古今”。

类似上面的标题、作者、正文和简析，即是文本内容的**结构 (Structure)**。显然，结构能使文本内容更加具有逻辑性和易读性。

5.1 网页的内容、结构与表现



兰州大学

3. 表现

虽然对上述文本内容进行了结构化，但在网页上展示这些文本内容时，还可以使用**表现性元素**和**表现性属性**对文本内容进行修饰。这样，文本内容就能够以特定的**字体、颜色、大小和对齐方式**展示在网页上，从而产生特定的视觉效果。文本内容的字体、颜色、大小和对齐方式即是内容在网页上的**表现 (Presentation)**。

5.1 网页的内容、结构与表现



兰州大学

【练习5-1】网页的内容、结构和表现。HTML代码如下：

```
<body bgcolor=Silver>
  <h1 align=center><font color=red>静夜思</font></h1>
  <h2 align=center>唐. 李白</h2>
  <p align=center><strong><font color=blue>床前明月光，疑是地
上霜。<br>
  举头望明月，低头思故乡。</font></strong></p>
  <p>【简析】这是写远客思乡之情的诗，诗以明白如话的语言雕琢出
明静醉人的秋夜的意境。它不追求想象的新颖奇特，也摒弃了辞藻的精工
华美；它以清新朴素的笔触，抒写了丰富深曲的内容。境是境，情是情，
那么逼真，那么动人，百读不厌，耐人寻绎。无怪乎有人赞它是“妙绝古
今”。</p>
</body>
```

问题：为了说明**结构**，使用了哪些HTML元素？为了实**现表现**，使用了哪些HTML元素或元素属性？

5.1 网页的内容、结构与表现



兰州大学

【练习5-1】网页的内容、结构和表现。HTML代码如下：

```
<body bgcolor=Silver>
  <h1 align=center><font color=red>静夜思</font></h1>
  <h2 align=center>唐. 李白</h2>
  <p align=center><strong><font color=blue>床前明月光，疑是地
上霜。<br>
  举头望明月，低头思故乡。</font></strong></p>
  <p>【简析】这是写远客思乡之情的诗，诗以明白如话的语言雕琢出
明静醉人的秋夜的意境。它不追求想象的新颖奇特，也摒弃了辞藻的精工
华美；它以清新朴素的笔触，抒写了丰富深曲的内容。境是境，情是情，
那么逼真，那么动人，百读不厌，耐人寻绎。无怪乎有人赞它是“妙绝古
今”。</p>
</body>
```

***绿色表示内容，红色说明结构，蓝色实现表现。

5.1 网页的内容、结构与表现



兰州大学

然而，HTML 4.01规范以及Web标准网页更加推荐“**内容和结构与表现的分离**”——“**内容和结构**”出现在**HTML文档**中，“**表现**”的定义则保存在**CSS文档**中。

5.2 HTML规范及文档类型定义



兰州大学

- 为了规范HTML的应用，W3C与1999年制订了HTML 4.01 规范（HTML 4.01 Specification）。
- 在HTML 4.01规范中，HTML文档主要包括严格（Strict）和过渡（Transitional）两种类型。
- 在HTML 4.01 Strict文档类型中，不能使用被淘汰的表现性元素和属性，而使用样式或样式表实现表现。
- 在HTML 4.01 Transitional文档类型中，还允许使用这些过时的表现性元素和表现性属性。

5.2 HTML规范及文档类型定义



兰州大学

➤ 为了声明HTML文档的类型，在HTML文档的**第一行**中需要添加如下代码：

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

➤ 该行代码称为**文档类型定义**（Document Type Definition, DTD），声明HTML文档属于HTML 4.01 Transitional文档类型。

5.2 HTML规范及文档类型定义



兰州大学

➤ 如要声明HTML 4.01 Strict文档类型，则需在HTML文档的第一行中使用如下的DTD代码：

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">
```

5.2 HTML规范及文档类型定义



兰州大学

- 在Strict类型的HTML文档中，不能使用如下元素：
applet、basefont、center、dir、font、iframe、
isindex、iframe、menu、s、strike和u。
- 在Strict类型的HTML文档中，不能使用过时的属性。例如，align、background和bgcolor。

5.2.1 HTML 4.01 Transitional文档的基本要求



➤ 【练习5-2】将【练习5-1】的HTML文档改写为HTML 4.01 Transitional文档。

```
<html>
  <head>
    <title>内容、结构和表现</title>
  </head>
  <body bgcolor=Silver>
    <h1 align=center><font color=red>静夜思</font></h1>
    <h2 align=center>唐.李白</h2>
    <p align=center><strong><font color=blue>床前明月光，疑是地上霜。<br>
    举头望明月，低头思故乡。</font></strong></p>
    <p>【简析】这是写远客思乡之情的诗，诗以明白如话的语言雕琢出明静醉人的
    秋夜的意境。它不追求想象的新颖奇特，也摒弃了辞藻的精工华美；它以清新朴素的
    笔触，抒写了丰富深曲的内容。境是境，情是情，那么逼真，那么动人，百读不
    厌，耐人寻绎。无怪乎有人赞它是“妙绝古今”。</p>
  </body>
</html>
```

练习 5-1

5.2.1 HTML 4.01 Transitional文档的基本要求



➤ 【练习5-2】将【练习5-1】的HTML文档改写为HTML 4.01 Transitional文档。HTML代码如下：

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>内容、结构和表现</title>
    <meta http-equiv="content-type" content="text/html; charset=gb2312">
  </head>
  .....
</html>
```

HTML 4.01 Transitional文档的基本要求如下：

- (1) 在第1行声明HTML 4.01 Transitional文档类型。
- (2) html元素必须包含head和body两个子元素，head元素必须包含title和meta两个子元素。
- (3) 在HTML文档的头部使用meta元素将编码方式设置为中文简体，具体使用如下代码：
`<meta http-equiv="content-type" content="text/html; charset=gb2312">`
该段代码表示，网页的内容类型是text或html格式，并使用gb2312字符集，gb2312是简体中文编码方式。这样，可以避免在Web浏览器中显示乱码。

5.2.2 W3C标记验证服务

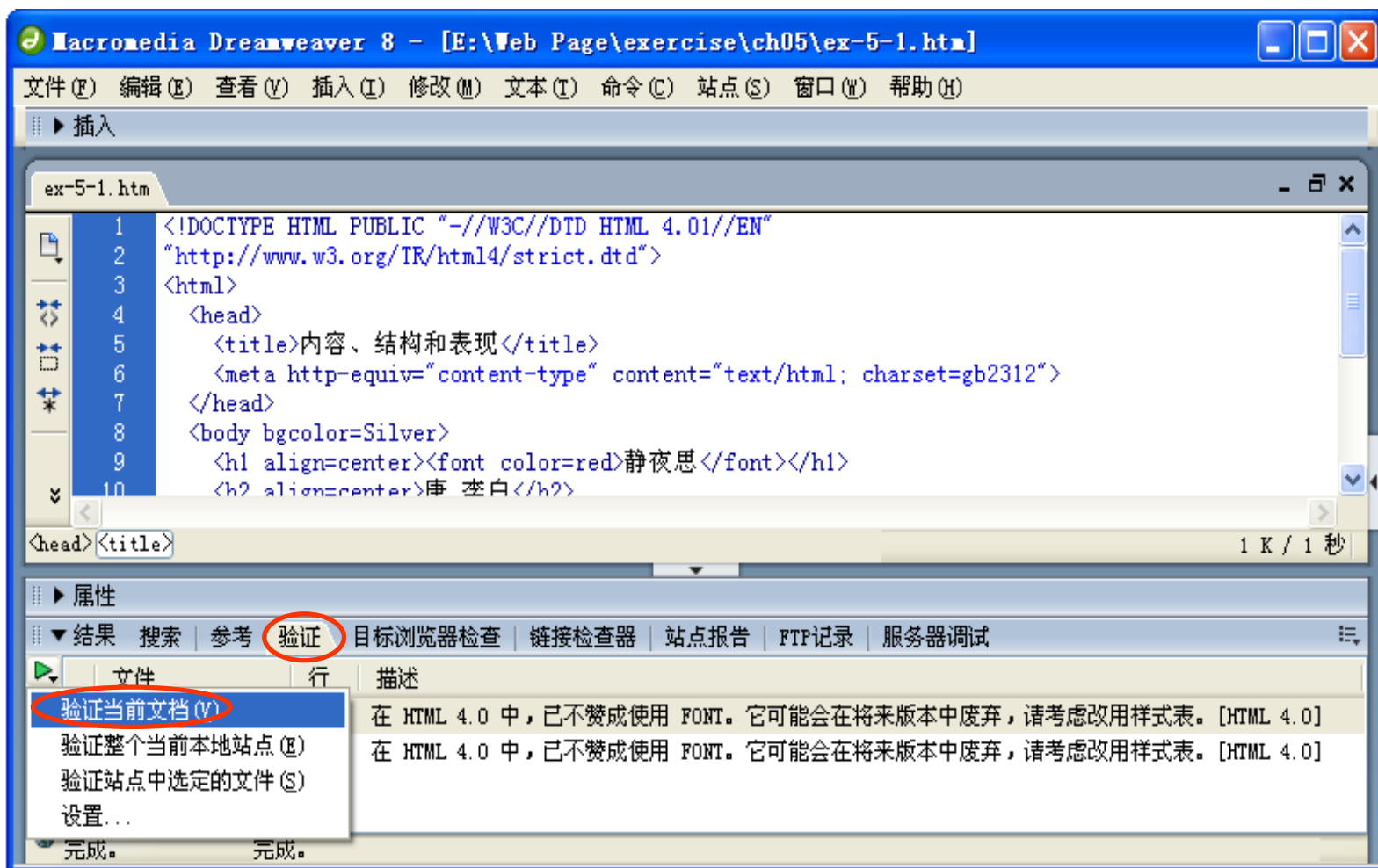


- 一个语法规则的HTML文档即可称为有效的。有效的HTML文档易于被Web浏览器识别和处理。
- W3C网站在线提供HTML文档的标记验证服务（Markup Validation Service）。标记验证服务不仅能够验证HTML文档是否符合语法规则，而且能够发现HTML文档中违反语法规则的代码。

[http://validator.w3.org/#validate by upload](http://validator.w3.org/#validate_by_upload)

- 【练习5-3】使用W3C标记验证服务。

5.2.3 使用Dreamweaver验证HTML文档的语法规范



5.3 内部样式表



- “内容和结构与表现的分离”是Web标准网页设计与制作的主要目标之一。
- 在HTML 4.01 Strict文档类型中，内容在网页中的表现是通过样式（Style）实现的。
- 在HTML 4.01 Strict文档的头部使用style元素定义内部样式表（Internal Style Sheet）是实现样式的通用方法之一。

5.3 内部样式表



- 一个样式表通常包含多条规则（Rule）。
- 每条规则主要由两个部分构成：选择器（Selector），以及一个或多个特性声明（Property Declaration）。
- 特性声明之间用分号（;）隔开。
- 规则的基本语法如下：

```
selector { property_declaration1; property_declaration2; ... }
```

- 在样式表的规则中，每个特性声明又由一个特性名称（Property Name）和一个特性值（Property Value）组成，特性名称和特性值用冒号（:）隔开。
- 因此，规则的基本语法又可以进一步表示如下：

```
selector { property_name1:property_value1; property_name2:property_value2; ... }
```

5.3 内部样式表

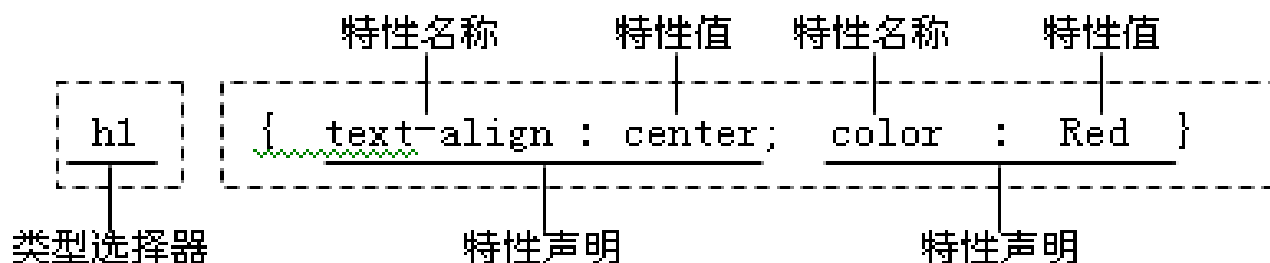


- 在样式表的规则中，选择器用于匹配HTML文档中需要进行样式控制的元素。
- 实现样式的选择器又可进一步分为类型选择器、类选择器、ID选择器、伪类选择器、后代选择器和子元素选择器等。

5.3.1 类型选择器

- 在样式表的规则中，最常见的选择器是类型选择器（Type Selectors），在Dreamweaver中又称标签选择器。
- 类型选择器用于匹配需要重新设定样式的HTML元素。

➤ `h1 { text-align:center; color:Red }`



规则的结构

➤ `body { background-color:Silver }`

5.3.2 类选择器



➤ 在样式表的规则中，还可以使用句号（.）定义类选择器（Class Selectors）。

```
.zhengwen { text-align:center; color:blue; font-weight:bolder }
```

<p class="zhengwen">床前明月光，疑是地上霜。
举头望明月，低头思故乡。</p>

5.3.2 类选择器



内部样式表

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <title>内容、结构和表现</title>
    <meta http-equiv="content-type" content="text/html; charset=gb2312">
    <style type="text/css">
      body { background-color:Silver }
      h1 { text-align:center; color:Red }
      h2 { text-align:center }
      .zhengwen { text-align:center; color:Blue; font-weight:bolder }
    </style>
  </head>
  <body>
    <h1>静夜思</h1>
    <h2>唐. 李白</h2>
    <p class="zhengwen">床前明月光，疑是地上霜。<br>
    举头望明月，低头思故乡。</p>
    <p>【简析】这是写远客思乡之情的诗，诗以明白如话的语言雕琢出明静醉人的秋夜的意境。它不追求想象的新颖奇特，也摒弃了辞藻的精工华美；它以清新朴素的笔触，抒写了丰富深曲的内容。境是境，情是情，那么逼真，那么动人，百读不厌，耐人寻绎。无怪乎有人赞它是“妙绝古今”。</p>
  </body>
</html>
```

***在HTML 4.01 Strict文档中，只能“通过样式实现内容的表现”。**

5.3.3 ID选择器



- ID选择器（ID Selectors）的定义和用法与类选择器基本相同，但一个ID选择器只能在HTML文档中的一个元素上应用一次。因此，ID选择器的针对性更强。
- 类选择器及其中的样式可以应用在多个元素上，且能够应用多次。
- 在样式表的规则中，可以使用井号（#）定义ID选择器。

5.3.3 ID选择器



【练习5-5】定义和应用ID选择器。HTML代码如下：

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <title>ID选择器</title>
    <meta http-equiv="content-type" content="text/html;
charset=gb2312">
    <style type="text/css">
      #one { color:Red }
      #two { color:Green }
    </style>
  </head>
  <body>
    <p id="one">ID选择器1</p>
    <p id="two">ID选择器2</p>
  </body>
</html>
```

5.3.4 伪类选择器



- 伪类选择器（Pseudo-Classes Selector）主要应用于a元素，用来定义文本超链接的不同状态。
- a元素有四个伪类，分别是link（未被访问的超链接）、visited（已被访问的超链接）、hover（鼠标悬停在超链接上）和active（被激活的超链接）。
- 默认情况下，未被访问的文本超链接是蓝色并且有下划线，已被访问的文本超链接则是紫色并且有下划线。
- 但使用伪类选择器可以定义动态的文本超链接——在不同状态下，文本超链接有更多不同的表现。

5.3.4 伪类选择器



```
<head>
  <title>伪类选择器</title>
  <meta http-equiv="content-type" content="text/html; charset=gb2312">
  <style type="text/css">
    a { text-decoration:none }
    a:link { color:Red }
    a:visited { color:Gray }
    a:hover { color:Green }
    a:active { color:Blue }
  </style>
</head>
<body>
  <p><a href="http://www.sina.com">创建指向新浪网的超链接</a></p>
</body>
```

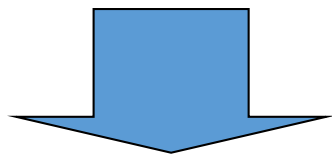
注意：在a元素上定义伪类选择器时，必须遵循LVHA的顺序，即 a:link → a:visited → a:hover → a:active 的顺序。否则，文本超链接的鼠标悬停和被激活样式就不起作用了。

5.3.5 选择器分组

➤ 如果几个选择器的特性声明相同，则可以在样式表中对**选择器分组（Grouping）**。这样，可以简化特性声明相同的选择器定义。

➤ 例如，有以下三条规则：

```
h1 { text-align:center; font-family:sans-serif; color:green }  
h2 { text-align:center; font-family:sans-serif; color:green }  
h3 { text-align:center; font-family:sans-serif; color:green }
```



```
h1, h2, h3 { text-align:center; font-family:sans-serif; color:green }
```

注意：在定义选择器分组时，选择器之间用逗号（,）隔开。

5.3.6 后代选择器



➤ 在样式表中，还可以定义后代选择器（Descendant Selectors）来创建一些规则。这些规则及其样式能对某些HTML元素的后代元素起作用，但对其他的HTML元素不起作用。

➤ 例如，`h1 em { color:red }`
该规则及其样式会把作为h1元素后代的em元素中的文本变为红色，但对其他的em元素及其中的文本则不起作用。

➤ 在后代选择器的定义中，选择器之间的空格是一种结合符（combinator）。空格结合符可以解释为“作为……后代的……”。因此，选择器（h1 em）可以解释为“作为h1元素后代的任何em元素将应用以下样式……”。

5.3.6 后代选择器



```
<head>
  <title>后代选择器及其应用示例</title>
  <meta http-equiv="content-type" content="text/html;
charset=gb2312">
  <style type="text/css">
    h1 em { color:red }
    .c1 em { color:blue }
  </style>
</head>
<body>
  <h1>This is an <em>important</em> heading.</h1>
  <h1>This is an <strong><em>important</em></strong> heading.</h1>
  <p class="c1">This is an <em>important</em> paragraph.</p>
  <p>This is an <em>important</em> paragraph.</p>
</body>
```

5.3.7 子元素选择器



➤ 在CSS样式表的规则中，还可以定义子元素选择器（Child Selectors）。这样，规则及其中的样式只能对某HTML元素的特定子元素起作用，但对与特定子元素同名的其它后代元素则不起作用。子元素选择器的定义使用大于号（>）。

例如，在CSS样式表中定义如下规则及子元素选择器：

```
h1>strong { color:red }
```

其中，h1>strong即是子元素选择器，表示h1元素与strong元素是父元素与子元素的关系。

在XHTML文档中有如下代码：

```
<h1>This is <strong>very</strong> important!</h1>
```

```
<h1>This is <em>really <strong>very</strong></em> important!</h1>
```


5.4 常用的样式特性



根据样式表中特性的作用对象，可以将样式特性分为**类型**、**背景**、**区块**、方框、边框、列表、定位和扩展等几个大类。

5.4.1 类型特性



1. 字体 (font-family)

该特性用于设置文本的字体系列。CSS 2.1定义了Serif、Sans-serif、Cursive、Fantasy和Monospace等5种通用字体系列。每种字体系列又包含若干字体。例如，Serif字体系列包括Times、Georgia和New Century Schoolbook等字体。

2. 大小 (font-size)

该特性用于设置文本的字体大小。font-size特性值通常是以px（像素）和em（字体高度）为单位的相对值。其中，em是相对于当前文本大小的宽度单位（即font-size值），如果当前的font-size值为16px，则1em=16px。

3. 粗细 (font-weight)

该特性用于设置文本的字体粗细。font-weight特性值可以是lighter、normal、bold或bolder，这些特性值将使文本的字体依次由细变粗。

5.4.1 类型特性



4. 字体样式 (font-style)

该特性用于设置文本的字体样式。font-style特性值可以是normal（正常）、italic（斜体）或oblique（倾斜）。

5. 行高 (line-height)

该特性用于设置文本所在行的行高。该特性作用于一个块级元素时，定义了该元素中基线之间的最小距离。line-height特性值通常并应该大于font-size特性值。line-height与font-size 的特性值之差一分为二，分别加到一个文本行内容的顶部和底部。

6. 修饰 (text-decoration)

使用该特性，可以对文本添加下划线（underline）、上划线（overline）、删除线（line-through），或使文本闪烁（blink），也可以不加任何修饰（none）。

5.4.1 类型特性



7. 颜色 (color)

该特性用于设置文本的颜色。

color特性值可以是预定义的颜色名称。例如，red (红)、yellow (黄)、blue (蓝)、silver (银)、teal (深青)、white (白)、navy (深蓝)、orchid (淡紫)、olive (橄榄)、purple (紫)、gray (灰)、green (绿)、lime (浅绿)、maroon (褐)、aqua (水绿) 和fuchsia (紫红)。

color特性值也可以是rgb代码。例如，rgb(255, 0, 0)对应红色，rgb(0, 255, 0)对应绿色，rgb(0, 0, 255)对应蓝色。

color特性值还可以是一个三位或六位的十六进制数。例如，#f00和#ff0000对应红色，#0f0和#00ff00对应绿色，#00f和#0000ff对应蓝色。

5.4.2 背景特性



➤背景特性主要作用于body、table和div等结构性和块级元素。

1. 背景颜色 (background-color)

该特性用于设置元素的背景颜色。与color特性值类似，background-color特性值可以是预定义的颜色名称，也可以是rgb代码，还可以是一个三位或六位的十六进制数。此外，background-color特性的默认值是transparent，表示背景是透明的。

2. 背景图像 (background-image)

该特性用于设置元素的背景图像。background-image特性值是一个指向图像的路径及文件名的URL。例如，body { background-image:url("../images/bkImage.jpg") }表示，将当前文档所在文件夹的上一级文件夹的下一级文件夹images中的图像文件bkImage.jpg设置为主体的背景图像。

5.4.3 区块特性



区块特性用来对文本中的单词间距（word-spacing）、字符间距（letter-spacing）、垂直对齐（vertical-align）、文本对齐（text-align）、文本缩进（text-indent）和显示（display）等特性进行设置。

1. 单词间距（word-spacing）

该特性用于设置单词之间的间隔。word-spacing特性值通常是表示固定宽度的绝对值，例如16px、1em和2cm。

2. 字符间距（letter-spacing）

该特性用于设置字符或字母之间的间隔。letter-spacing特性值通常是表示固定宽度的绝对值，例如16px、1em和2cm。

3. 垂直对齐（vertical-align）

该特性用于设置文本的垂直对齐方式。vertical-align特性值可以是sub（下标）、super（上标）、top（顶端对齐）、middle（居中）或bottom（底端对齐）。

5.4.3 区块特性



4. 文本对齐 (text-align)

该特性用于设置文本的**水平**对齐方式。text-align特性值可以是left（左对齐）、right（右对齐）、center（居中）或justify（两端对齐）。

5. 文本缩进 (text-indent)

该特性用于设置段落的首行**缩进**。text-indent特性值通常是表示固定缩进宽度的绝对值，例如16px、1em和2cm。

6. 显示 (display)

该特性用于设置是否显示元素以及如何显示元素。display特性值是inline、block或none。其中，**inline**是默认值，表示将元素显示为行内元素，行内元素的前后没有换行符；**block**表示将元素显示为块级元素，块级元素的前后带有换行符；**none**表示不显示元素。

5.5 行内样式



➤ 在HTML元素的开始标签中使用style属性，可以定义行内样式（Inline Style），又称内联样式。

➤ 例如，

```
<h1 style="text-align:center; color:red; letter-spacing:1em">静夜思</h1>
```


5.5 行内样式



```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <title>行内样式表</title>
    <meta http-equiv="content-type" content="text/html; charset=gb2312">
  </head>
  <body style="background-color:Silver">
    <h1 style="text-align:center; color:Red; letter-spacing:1em">静夜思</h1>
    <h2 style="text-align:center">唐. 李白</h2>
    <p style="text-align:center; color:Blue; font-weight:bolder">
      床前明月光，疑是地上霜。<br>
      举头望明月，低头思故乡。</p>
    <p>【简析】这是写远客思乡之情的诗，诗以明白如话的语言雕琢出明静醉人的秋夜的意境。它不追求想象的新颖奇特，也摒弃了辞藻的精工华美；它以清新朴素的笔触，抒写了丰富深曲的内容。境是境，情是情，那么逼真，那么动人，百读不厌，耐人寻绎。无怪乎有人赞它是“妙绝古今”。</p>
  </body>
</html>
```

注意：行内样式直接定义并作用在单个的HTML元素上。

5.6 外部样式表及其应用



- 虽然在HTML文档中可以使用内部样式表或行内样式，但更有效、更规范的方法是将样式表定义并保存在单独的CSS文档中。
- 定义并保存在CSS文档中的样式表，称为外部样式表（External Style Sheet）。

5.6.1 创建外部样式表



- 外部样式表可以直接在文本编辑软件（如Notepad）中进行创建和修改，并保存在CSS文档中。
- CSS文档是后缀名为css的文本文件。在CSS文档中，只包含定义样式的规则。

5.6.2 W3C在线CSS验证服务



- W3C网站不仅在线提供HTML文档的标记验证服务，而且在线提供CSS验证服务（CSS Validation Service）。
- CSS验证服务不仅能够验证CSS文档是否符合语法规则，而且能够发现CSS文档中违反语法规则的代码。

5.6.3 在HTML文档中链接外部样式表



- 在HTML文档的**头部**，可以使用**link元素**链接CSS文档及其中的样式表。link元素的具体用法如下：

<link rel="stylesheet" type="text/css" href="CSS文档路径及文件名">

- 这样，Web浏览器会从CSS文档中读取样式规则，并根据**样式规则中的选择器**匹配HTML文档中的元素，进而对HTML元素所作用的内容应用相应的样式。

5.6.3 在HTML文档中链接外部样式表



【练习5-9】使用外部样式表改写【练习5-4】。具体步骤如下：

1. 在CSS文档中创建外部样式表。使用Notepad软件编辑如下代码，并将这些代码保存在CSS文档（5-9.css）中。

```
body { background-color:Silver }  
h1 { text-align:center; color:Red; letter-spacing:1em }  
h2 { text-align:center }  
.zhengwen { text-align:center; color:Blue; font-weight:bolder }
```

2. 验证CSS文档的规范性。

5.6.3 在HTML文档中链接外部样式表

3. 在HTML文档（5-9.htm）中使用link元素链接外部样式表。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <title>内容和结构</title>
    <meta http-equiv="content-type" content="text/html; charset=gb2312">
    <link rel="stylesheet" type="text/css" href="5-9.css">
  </head>
  <body>
    <h1>静夜思</h1>
    <h2>唐. 李白</h2>
    <p class="zhengwen">床前明月光，疑是地上霜。<br>
    举头望明月，低头思故乡。</p>
    <p>【简析】……</p>
  </body>
</html>
```

5.6.4 在HTML文档中嵌入外部样式表

➤ 在HTML文档头部的style元素中，也可以使用import语句嵌入外部样式表。import语句的具体用法如下：

```
<head>
.....
<style type="text/css">
    @import url("CSS文档路径及文件名称");
    或者
    @import "CSS文档路径及文件名称";
</style>
.....
</head>
```


5.7 层叠样式表（CSS）



- 层叠样式表（Cascading Style Sheets, CSS）是用于控制网页样式、实现“内容和结构与表现的分离”的一种方法和技术。
- HTML文档与CSS文档的关系就是“内容和结构”与“表现”的关系——由HTML文档组织网页的内容和结构，而通过CSS文档控制网页内容在Web浏览器中的表现。

5.7 层叠样式表（CSS）



在CSS方法和技术中，能够以三种方式定义样式。

（1）在HTML元素的开始标签中使用style属性可以定义和应用行内样式。

（2）在HTML文档的头部使用style元素可以集中定义内部样式表。

（3）在CSS文档中单独定义外部样式表。

5.7.1 CSS的层叠性



➤ **层叠性 (Cascading)** 是指当有**多个选择器**作用于**同一HTML元素**时，Web浏览器如何处理多个选择器中的样式。

(1) 如果多个选择器定义的样式未发生冲突，则元素将应用相匹配的所有选择器定义的样式。

```
<head>
```

```
.....
```

```
<style type="text/css">
```

```
  p { color:red }
```

```
  #i1 { text-decoration:underline }
```

```
  .c1 { font-weight:bolder }
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>p元素将应用……选择器定义的样式 </p>
```

```
<p class="c1">p元素将应用……选择器和……选择器定义的样式</p>
```

```
<p id="i1" class="c1">p元素将应用……选择器、……选择器和……选择器定义的样式</p>
```

```
</body>
```

5.7.1 CSS的层叠性



(2) 如果多个选择器定义的样式发生冲突，则HTML元素将根据选择器的优先级应用某个选择器定义的样式。CSS规范约定，选择器的优先级从高到低依次为：行内样式 > ID选择器 > 类选择器 > 类型选择器。

```
<head>
.....
<style type="text/css">
  p { font-style:italic; color:red }
  .c_green { color:green }
  #i_blue { color:blue }
  #i_olive { color:olive }
</style>
</head>
<body>
  <p>p元素应用以下…选择器定义的样式</p>
  <p class="c_green">p元素应用以下…选择器和…选择器定义的样式</p>
  <p id="i_blue" class="c_green">p元素应用以下…选择器和…选择器定义的样式</p>
  <p style="color:purple" id="i_olive">p元素应用以下…选择器定义的样式以及…样式
</p>
</body>
</html>
```

5.7.2 CSS的继承性

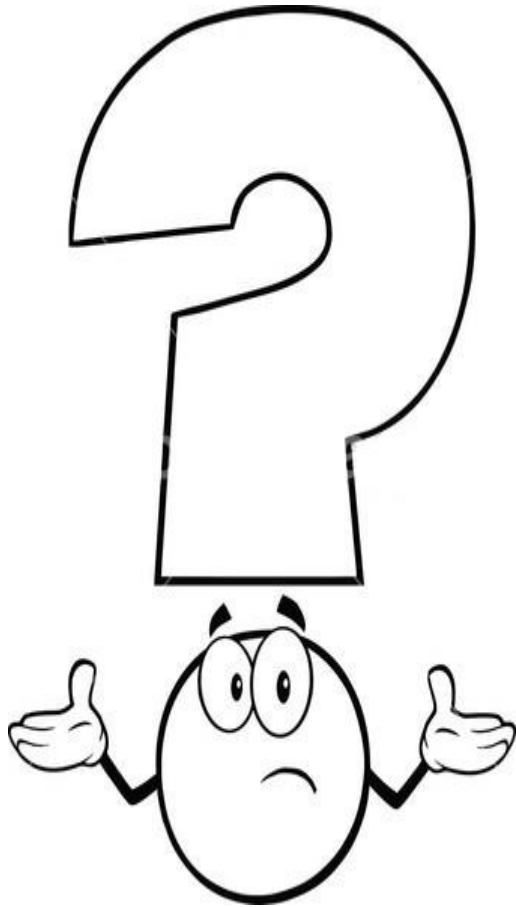


➤CSS的继承性（Inheritance）依赖于HTML元素之间的祖先—后代关系。CSS的继承性允许样式不仅可以应用于某个HTML元素，而且可以应用于该HTML元素的后代元素，除非出现优先级更高的样式。

```
<head>
.....
<style type="text/css">
  body { color:red }
  .c1 { color:green; text-decoration:underline }
</style>
</head>
<body>
  <p>第一个p元素.....<br>
    <em>第一个em元素.....</em>
  </p>
  <p class="c1">第二个p元素.....<br>
    <em>第二个em元素.....</em>
  </p>
</body>
```

- 网页的内容、结构与表现
 - 内容：网页展示的信息
 - 结构：能使文内容更加具有逻辑性和易读性
 - 表现：使内容以特定的字体、颜色、大小和对齐方式展示在网页上
- HTML文档主要包括严格和过渡两种类型
- CSS：实现了内容和结构与表现的分离
- 在样式表的规则中，选择器用于匹配HTML文档中需要进行样式控制的元素。
- 实现样式的选择器又可进一步分为类型选择器、类选择器、ID选择器、伪类选择器、后代选择器和子元素选择器等。

- 三种方式定义样式：
 - (1) 在HTML元素的开始标签中使用style属性可以定义和应用行内样式。
 - (2) 在HTML文档的头部使用style元素可以集中定义内部样式表。
 - (3) 在CSS文档中单独定义外部样式表。
- 如果多个选择器定义的样式发生冲突，则HTML元素将根据选择器的优先级应用某个选择器定义的样式。
CSS规范约定，选择器的优先级从高到低依次为：行内样式 > ID选择器 > 类选择器 > 类型选择器。



Questions?