3 CSS使用及web元素

浙 江 师 范 大 学

主讲老师：徐晓丹（663601）

主要内容

- CSS选择器
- web元素

样式表的定义与使用

- 内联方式
- 内部样式块对象
- 外部样式表

内联方式

- 内联定义即是在对象的标记内使用对象的style属性定义适用的样式表属性。
- 基本语法：
  - <标记 style="样式属性:属性值;样式属性:属性值;…">
- 语法说明：
  - 标记：HTML标签，如body、table、p等。
  - 标记的style定义只能影响标记本身。
  - style的多个属性之间用分号分隔。
  - 标记本身定义的style优先于其他所有样式定义。

内部样式块对象

- 内部样式块对象允许在它们所应用的HTML文档的顶部head区域设置样式，后在整个HTML文件中直接调用使用该样式的标记。
- 基本语法：
  - <style>
    - 选择器1{样式属性1：属性值1；样式属性2：属性值2；……}
    - 选择器2{样式属性1：属性值1；样式属性2：属性值2；……}
    - ……
    - 选择器n{样式属性1：属性值1；样式属性2：属性值2；……}
  - </style>
- 语法说明：
  - <style>元素是用来说明所要定义的样式，可以有type属性。
  - 选择器可以使用HTML标记的名称，所有HTML标记都可以作为选择器。
  - 样式属性主要是关于对选择器格式化显示风格的样式属性名称。

外部样式表

- 如果要引用外部样式表文件，一般使用链接的方式。
- 基本语法：
  - <link href="外部样式表的文件名称" rel="stylesheet" type="text/css">
- 语法说明：
  - 链接外部样式表时，不需要使用style元素，只需直接用<link>标记放在<head>标记中就可以了。
  - 外部样式表的文件扩展名为.css。
  - CSS文件一定是纯文本格式。
  - 在修改外部样式表时，引用它的所有网页也会自动地更新样式。
  - 外部样式表中的URL相对于样式表文件在服务器上的位置。
  - 外部样式表优先级低于内部样式表。
  - 可以同时链接几个样式表，靠后的样式表优先于靠前的样式表。

选择器

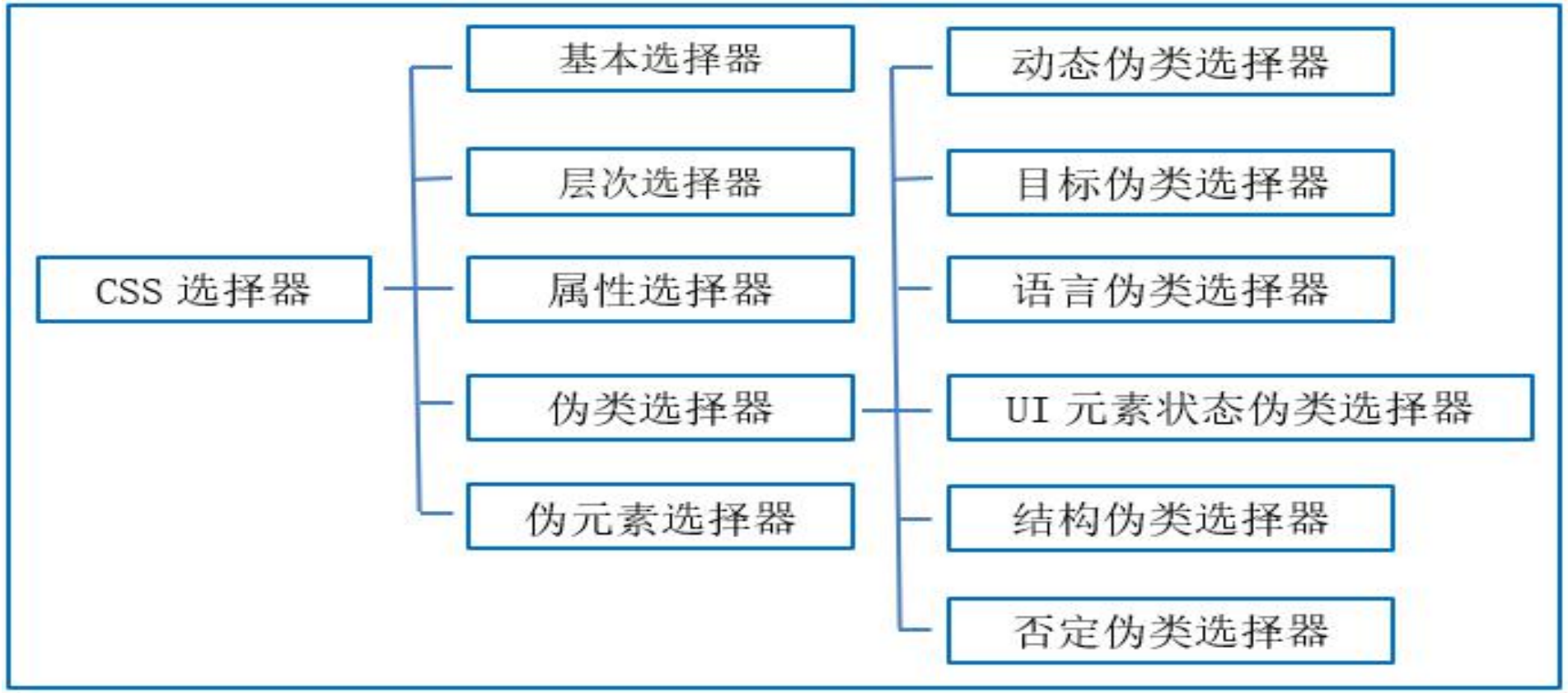


图 3-1 CSS 选择器分类图

基本选择器

- 基本选择器是CSS中使用最频繁、最基础，也是CSS中最早定义的选择器，包括通配选择器、类选择器、ID选择器、类型选择器和群组选择器等。

1. 通配符选择器（Universal Selector）

- 基本语法：
  - * { sRules }
- 使用说明：
  - 通配选择符“*”匹配任何元素名称，匹配选定文档树中的任意单一元素。
  - 例如：设置所有元素的外边距margin和内边距padding为0.

```
*{margin:0px;padding:0px;}
```

- 缺点：*会匹配所有元素，影响网页渲染的时间。

将需要设置的多个元素放一起

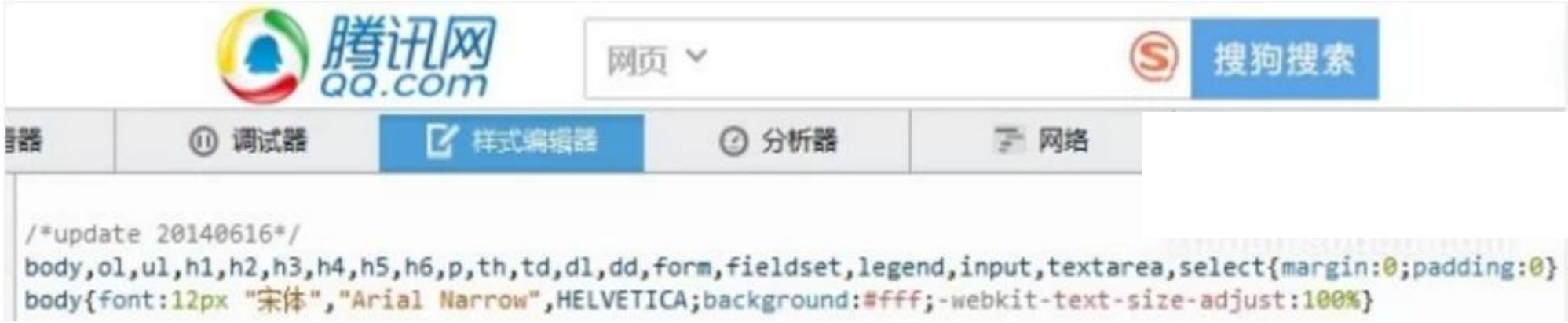
淘宝

blockquote,body,button,dd,dl,dt,fieldset,form,h1,h2,h3,h4,h5,h6,hr,input,legend,li,ol,p,pre,td,textarea,th,ul{ margin:0;　　padding:0}

将需要设置的多个元素放一起

腾讯

body,ol,ul,h1,h2,h3,h4,h5,h6,p,th,td,dl,dd,form,fieldset,legend,input,textarea,sele
ct{margin:0;padding:0}



2. 类型(元素)选择器（Type Selectors）

- 基本语法：

  E { sRules }

- 使用说明：
  - 类型选择器也可以叫元素选择器（Element Selectors），它匹配一个HTML元素，比如body、p、h1等。

3. 类选择器（Class Selectors）

- 基本语法：
  - E.className { sRules }
- 使用说明：
  - 类选择器的基本语法只对应用了该类（className）的E元素起作用，其简写形式“.className { sRules }”（比如“.news { color: red; }”）会影响所有标记中定义了“class="news"”类的元素，而“div.news{color:red;}”只影响属性中定义了“class="news"”类的div元素。
  - 可以为对象的class属性（特性）指定多于一个值（className），其方法是指定用空格隔开的一组样式表的类名，例如：<p class="class1 class2">。

4. ID选择器（ID Selectors）

- 基本语法：
  - E#IDName { sRules }
- 使用说明：
  - 以文档目录树（DOM）中作为对象的唯一标识符的ID作为选择器，特别注意其唯一性，不能在多个元素中使用相同ID样式。

层次选择器

- 层次选择器（Hierarchy Selectors）也可以称为关系选择器（Relationship Selectors），它是通过HTML的DOM元素间的层次关系获取元素，其主要的层次关系包括包含后代（包含）、父子、相邻兄弟和通用兄弟关系，通过其中某类关系可以方便快捷地选定需要的元素。

1. 后代选择器（Descendant combinator）

- 基本语法：
  - E F { sRules }
- 使用说明：
  - 要选择匹配的元素在文档树中是另一个元素的后代，也是一种包含关系，所以后代选择器也称为包含选择器，后代选择器是由空格分隔的两个或两个以上的选择器，位置排在前面的选择器是祖先元素，相对位置靠后的是后代元素。

```
h1 em {color:red;}
<h1>This is a <em>important</em> heading</h1>
<p>This is a <em>important</em> paragraph.</p>
```

2. 子选择器（Child combinator）

- 基本语法：
  - E > F { sRules }
- 使用说明：
  - 子选择器实际上是包含选择器的一种特例，即元素E和F之间是父子关系，中间不隔代。注意，“>”前后可以有空格也可以没有空格。
  - h1 > strong {color:red;}
  - <h1>This is very very important.</h1>
  - <h1>This is really very important.</h1>

把第一个 h1 下面的两个 strong 元素变为红色，但是第二个 h1 中的 strong 不受影响

3. 兄弟选择器（General sibling combinator）

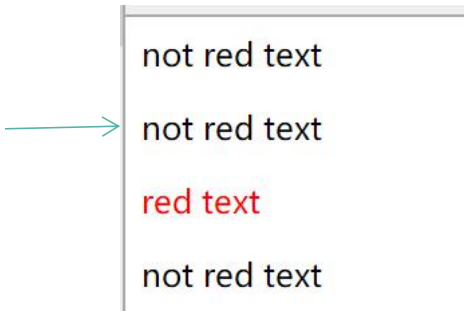
- 基本语法：
  - E ~ F { sRules }
- 使用说明：
  - CSS3新增的通用兄弟元素选择器类型，选择和E元素同级别的所有的F元素。
  - 例如，div~p匹配

1 2 3 片段中的 3 。

4. 相邻选择器（Adjacent sibling combinator）

- 基本语法：
  - E + F { sRules }
- 使用说明：
  - 相邻选择器是兄弟选择器的一种特例。
  - 选择所有作为E元素相邻的下一个元素F，它们必须有相同的父节点。

```
<!--相邻选择器选择每个div紧邻后的一个元素p-->
<style>
  div+p { color: red;   }
</style>
<div>
   <p>not red text</p>
   <p>not red text</p>
</div>
<p>red text</p>
<p>not red text</p>
```



属性选择器

- 在HTML中，通过各种各样的属性可以给元素增加很多附加信息。例如，通过ID属性可以将不同DIV元素进行划分。属性选择器（Attribute Selectors）是指通过元素的属性关系来定位的选择器。

1. CSS2属性选择器

- 基本语法：
  - E [attr] { sRules }
  - E [attr = "value"] { sRules }
  - E [attr ~= "value"] { sRules }
  - E [attr |= "value"] { sRules }
- 使用说明分别为：
  - 选择具有attr属性的E。
  - 选择具有attr属性且属性值等于value的E。
  - 选择具有attr属性且属性值为一个用空格分隔的字词列表，其中一个等于value的E，这里的value不能包含空格。
  - 选择具有attr属性且属性值为一个用连字符“-”分隔的字词列表，由value开始的E。

```
<html lang="en">
<head>
    <style>
     img[alt]
     {
    border: 5px solid #0d6cca;
     }
     </style>
   </head>
<h1>有边框样式：</h1>
<img src="image/22new.jpg" alt="水果之家" />
<hr />
<h1>无边框样式：</h1>
<img src="image/22new.jpg" />
</body>
</html>
```

有边框样式：

无边框样式：

2. CSS3新增的属性选择器

表 2-1 CSS3 新增的属性选择器

选择器	说明
E [attr ^= "value"] { sRules }	选择匹配 E 的元素，且该元素定义了 attr 属性，attr 属性值包含前缀为“value”的子字符串。如果省略 E 表示可匹配任意类型的元素。例如：body[lang^="en"] 匹配 <body lang="en-us"></body>，而不匹配<body lang="fr-argot"></body>
E [attr $= "value"] { sRules }	选择匹配 E 的元素，且该元素定义了 attr 属性，attr 属性值包含后缀为“value”的子字符串。如果省略 E 表示可匹配任意类型的元素。例如：img[src$="jpg"]匹配，而不匹配
E [attr *= "value"] { sRules }	选择匹配 E 的元素，且该元素定义了 attr 属性，attr 属性值包含“value”的子字符串。如果省略 E 表示可匹配任意类型的元素。例如：img[src*="jpg"]匹配，而不匹配

伪类选择器

- 伪类选择器（Pseudo-Classes Selectors）可以分为六种：动态伪类选择器、目标伪类选择器、语言伪类选择器、UI元素状态选择器、结构伪类选择器和否定伪类选择器。
- 伪类可以看作是一种特殊的类选择器，是能被支持CSS的浏览器自动所识别的特殊选择器。它的最大的用处就是可以对链接在不同状态下定义不同的样式效果。
- 基本语法（在原有的语法里加上一个伪类）：
  - selector:pseudo-class { property: value }
- selector为元素选择器，pseudo-class为CSS伪类选择器名称，中间用冒号（:）分隔。伪类和类不同，是CSS已经定义好的，不能像类选择器一样随意用别的名字，根据上面的语法可以解释为对象（选择器）在某个特殊状态下（伪类）的样式。
- 类选择器及其它选择器也同样可以和伪类混用：
  - selector.class:pseudo-class { property: value }

1. 动态伪类选择器（Dynamic pseudo-classes Selectors）

- 动态伪类并不存在于HTML中，只有当用户和网站交互的时候才能体现出来。动态伪类包含两种，第一种是在链接中常看到的锚点伪类，另一种为用户行为伪类。
- （1）E:link
  - 语法：Selector : link { sRules }
  - 说明：属于链接伪类选择器，设置a对象在未被访问前的样式。默认值由浏览器决定。对于无href属性（特性）的a对象，此伪类不发生作用。
- （2）E:visited
  - 语法：Selector : visited { sRules }
  - 说明：属于链接伪类选择器，设置a对象在其链接地址已被访问过时的样式。默认值由浏览器决定。定义网页过期时间或用户清空历史记录将影响此伪类的作用。对于无href属性(特性)的a对象，此伪类不发生作用。

- （3）E:hover
  - 语法：Selector : hover { sRules }
  - 说明：属于用户行为伪类选择器，设置对象在其鼠标悬停时的样式。在CSS2中此伪类可以应用于任何对象。
- （4）E:active
  - 语法：Selector : active { sRules }
  - 说明：属于用户行为伪类选择器，设置对象在被用户激活(在鼠标点击与释放之间发生的事件)时的样式。在CSS2中此伪类可以应用于任何对象。并且:active可以和:link以及:visited状态同时发生。
- （5）E:focus
  - 语法：Selector : focus { sRules }
  - 说明：属于用户行为伪类选择器，设置对象获得焦点时的样式。

- 如下示例代码展示了动态伪类选择器在 <a> 元素上应用：
- a:link { color: #FF0000 }　　/* 未访问的链接 */
- a:visited { color: #00FF00 }　　/* 已访问的链接 */
- a:hover { color: #FF00FF }　　/* 鼠标移动到链接上 */
- a:active { color: #0000FF }　　/* 选定的链接 */

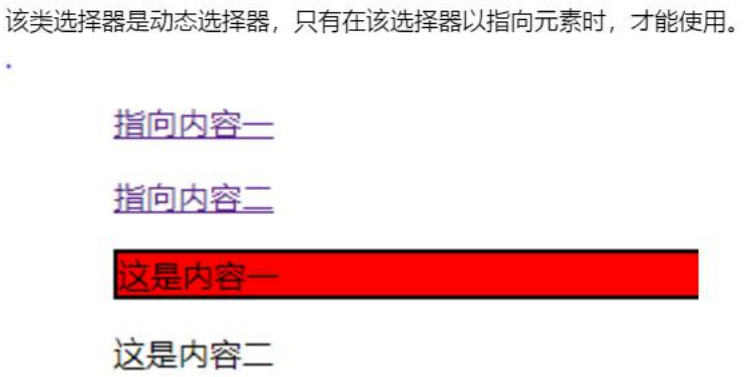
- 动态伪类选择器并不是只能应用于 <a> 元素，其他 HTML 元素同样可以使用动态伪类选择器。

```
<style>
   /* 鼠标移动到链接上 */
   button:hover {
     background-color: lightblue;
   }
  </style>
</head>
<body>
  <button>这是一个按钮</button>
</body>
```

2. 目标伪类选择器（Target pseudo-class Selectors）

- 语法：E:target { sRules }
- 说明：匹配相关URL指向的E元素。用来匹配锚点指向的元素，突出显示活动的HTML锚。

```
<style>
        div:target{
            border: 2px solid black;
            background: red;
        }
    </style>
<body>
    <p><a href="#nei1">指向内容一</a><br></p>
    <p>  <a href="#nei2">指向内容二</a>  </p>
   <div id="nei1">这是内容一</div>
   <div id="nei2">这是内容二</div>
</body>
```



3. 语言伪类选择器（Language pseudo-class Selectors）

- 语法：E:lang(fr) { sRules }
- 说明：匹配使用特殊语言的E元素。

4. UI元素状态伪类选择器（UI element states pseudo-classes Selectors）

- UI元素状态伪类中的UI是User Interface（用户界面）的简写。UI设计是指网页的人机交互、操作逻辑、界面美观的整体设计。

表 2-2 UI 元素状态伪类选择器

选择器	说明
E:enabled	匹配所有用户界面（form 表单）中处于可用状态的 E 元素
E:disabled	匹配所有用户界面（form 表单）中处于不可用状态的 E 元素
E:checked	匹配所有用户界面（form 表单）中处于选中状态的 E 元素

5. 结构伪类选择器（Structural pseudo-classes Selectors）

- 结构伪类选择器是针对HTML的结构进行分类的选择器，特征就是位置分为三大类
- 一类

e:first-child

e:last-child

e:only-child

e:nth-child(n/even/odd)隔行换色，列表换色

e:nth-last-child

- 例如：设置表格格式

```
tr:nth-child(odd){
        background-color: #e0e0e0;
    }
    tr:first-child{
        background-color: beige;
    }
```

班级	姓名	成绩总分
软件20	王小明	260
软件20	李小东	275
软件20	李小强	275
软件20	李小许	278

结构伪类选择器

分类	选择器	说明
通用子元素过滤器	E:nth-child(n)	选择所有在其父元素中第 n 个位置的匹配 E 的子元素。参数 n 可以是数字（1、2、3）、关键字（odd、even）、公式（2n、2n+3），参数的索引起始值为 1，而不是 0。例如：tr:nth-child(3)匹配所有表格里时排第 3 行的 tr 元素；tr:nth-child(2n+1)匹配所有表格的奇数行；tr:nth-child(2n)匹配所有表格的偶数行；tr:nth-child(odd)匹配所有表格的奇数行；tr:nth-child(even)匹配所有表格的偶数行。
	E:nth-last-child(n)	选择所有在其父元素中倒数第 n 个位置且匹配 E 的子元素。该选择器的计算顺序与 E:nth-child(n)相反，但语法和用法相同。
通用子类型元素过滤器	E:nth-of-type(n)	选择父元素中第 n 个位置且匹配 E 的子元素。所有匹配 E 的子元素被分离出来单独排序，非 E 的子元素不参与排序。参数 n 可以是数字（1、2、3）、关键字（odd、even）、公式（2n、2n+3），参数的索引起始值为 1，而不是 0。例如：p:nth-of-type(2)匹配 片段中后面那个 p 元素，因为 h1 元素未参与排序。
	E:nth-last-of-type(n)	选择所有在其父元素中倒数第 n 个位置且匹配 E 的子元素。该选择器的计算顺序与 E:nth-of-child(n)相反，但语法和用法相同。

特定位置的子元素	E:first-child	选择位于其父元素中第一个位置且匹配E的子元素。示例：p a:first-child { color: green; } table td:first-child { width:200px; }
	E:last-child	选择位于其父元素中最后一个位置且匹配 E 的子元素。例如：h1:last-child 匹配 片段中的 h1 元素。
	E:first-of-type	选择在其父元素中匹配 E 的第一个同类型的子元素，该选择器的功能类似于E:nth-of-type(1)。例如：p:first-of-type 匹配 片段中的第一个 p 元素。
	E:last-of-type	选择在其父元素中匹配 E 的最后一个同类型的子元素，该选择器的功能类似于E:nth-last-of-type(1)。例如：p:last-of-type 匹配 片段中的第二个 p 元素。
特定状态的元素	E:root	选择匹配 E 所在文档的根元素。注意，在(X)HTML 文档中，根元素就是 html 元素，此时该选择器与 html 类型选择器匹配的内容相同。
	E:only-child	选择其父元素只包含一个子元素且该子元素匹配 E。例如：p:only-child 匹配 片段中的 p 元素，但不匹配 片段中的 p 元素。
	E:only-of-type	选择其父元素只包含一个同类型的子元素且该子元素匹配 E。例如：p:only-of-type 匹配 片段中的 p 元素，也匹配 片段中的 p 元素。
	E:empty	选择匹配 E 的元素且该元素不包含子节点，注意文本也属于节点。

6. 否定伪类选择器（Not pseudo-classes Selectors）

- 语法：E:not(s) { sRules }
- 说明：否定伪类选择器类型。选择匹配E的所有元素，且过滤掉匹配s选择器的任意元素。

伪元素选择器

- 伪元素（Pseudo-Elements）用于向某些选择器设置特殊效果，它是创造关于文档语言能够指定的文档树之外的抽象。例如文档语言不能提供访问元素内容第一字或者第一行的机制，伪元素还提供在源文档中不存在的内容分配样式。CSS3将伪元素选择器（Pseudo-Element Selectors）前面的单个冒号（:）修改为双冒号（::）用以区别伪类选择器（Pseudo-Classes Selectors），但以前的写法仍然有效，如表2-4所示。

伪元素选择器

表 2-4 伪元素伪类选择器

选择器	说明
E::first-letter	设置对象内的第一个字符的样式。
E::first-line	设置对象内的第一行的样式。
E::before	设置在对象前（依据对象树的逻辑结构）发生的内容。用来和 content 属性一起使用。
E::after	设置在对象后（依据对象树的逻辑结构）发生的内容。用来和 content 属性一起使用。
E::placeholder	设置对象文字占位符的样式。
E::selection	设置对象被选择时的颜色。

常用Web元素

副标题

内容

- 特殊符号的使用方法
- 图像、超链接、列表、表格元素的使用方法
- figure和iframe元素的使用

添加特殊符号

表 5-1 特殊符号代码表

特殊符号	实体名称	特殊符号	实体名称	特殊符号	实体名称
&	&amp;	±	&plusmn;	·	&middot;
©	&copy;	×	&times;	§	&sect;
®	&reg;	<	&lt;	¢	&cent;
£	&pound;	>	&gt;	€	&euro;
¥	&yen;	"	&quot;	TM	&trade;

图像

- 高质量的图片一般用TIFF格式保存，但其图片体积过大，不太适合网络传输。不同的图片格式会表现出不同的颜色分辨率和颜色标准，当然也会影响其体积的大小。网页中常用的图片格式有图像格式（如GIF、JPEG、PNG）和图形格式（如SVG）。

JPG图片

PNG图片

图像标签

- 一般在网页设计中选择的图片尽量控制在20KB以下，如果必须选用较大图片时，可先将其分成若干小图片，显示时再通过表格将这些小图片拼合起来。如果在同一文件中多次使用相同的图片时，最好使用相对路径查找该图片。
- 基本语法：<img src="url" alt=" " >
- 说明：
  - 1）img元素向网页中嵌入一幅图像。但从技术上讲，<img> 标签并不会在网页中插入图像，而是从网页上链接图像，<img>标签创建的是被引用图像的占位空间。其中src是其必须的属性；src属性的功能是指定图像源，即图像的URL路径；alt属性规定图像的替代文本。
  - 2）src属性用来指定图像源，即图像的URL路径。该路径可以是相对路径，也可以是绝对路径。
  - 3）alt属性用于对图像信息进行描述。如果图片在Web浏览器不支持图像显示、用户关闭图像、网速慢还未下载完等情况下无法显示时用来替代图像的说明文字，也是个必需的属性。

图像的宽高、间距与边框

- 默认情况下，在HTML页面中显示的是图像的原始大小。要显示自己指定的图像大小，可以使用CSS的width和height属性来代替<img>的宽高属性。使用CSS的边距margin来代替<img>标签的hspace和vspace属性，指定图像的水平间距和垂直间距，默认值为0。使用CSS的border属性来实现图片的边框，默认没有边框。

图像的对齐方式

- 使用CSS来实现图片的垂直对齐相对复杂一点，一定要注意的是图像的对齐方式是设在其块级（block）父元素上，不是设在img本身。

figure

- <figure>标签规定独立的流内容（图像、图表、照片、代码等），比如文档中插图，figure元素的内容应该与主内容相关，但如果被删除，也不应对文档流产生影响。
- figure元素内一般还会用到<figcaption>标签，它定义figure元素的标题（caption）。figcaption元素应该被置于figure元素的第一个或最后一个子元素的位置。
- figure用于对元素进行组合。多用于图片与图片描述组合。而img只是一个图片元素而已。可以嵌套在figure中使用。

超链接

- 超链接在本质上属于一个网页的一部分，它是一种允许一个网页同其他网页或站点之间进行链接的元素。各个网页链接在一起后，才能真正构成一个网站。
- 超文本具有的链接能力，层层链接相关文件，这种具有超级链接能力的特性，即称为超链接。超链接除了可链接文本外，也可链接各种媒体，如声音、图像和动画等，通过它们可以将网站建设成一个丰富多彩的多媒体世界。

创建超链接

- 语法：超链接内容
- 说明：
  - 1）<a>表示一个链接的开始，表示链接的结束。
  - 2）href属性规定链接的目标地址，可以链接到网页或其它文件地址，比如一个pdf等其它文档。既可以是一个绝对地址，也可以是一个相对地址。
  - 3）target属性用于指定打开链接的目标窗口，仅在href属性存在时使用，其默认方式是原窗口，具体值如下：
    - _blank：在新窗口中打开被链接文档，如无特殊需要请勿滥用。
    - _self：在被点击的同一框架中打开被链接文档（默认）。
    - _parent：在父框架集中打开被链接文档。
    - _top：在窗口主体中打开被链接文档。
    - framename：在指定的框架中打开被链接文档。不再推荐。
    - _self、_parent 和_top 这三个值大多数时候与iframe一起使用。
  - 4）title属性用于指定指向链接时所显示的提示信息。
  - 5）超链接内容是指链接目标资源的文字显示说明，单击“超链接内容”，就会链接到指定URL的资源。

1. 超链接路径

（1）绝对路径

（2）相对路径

（3）根路径

2. 内部链接

表 5-2 站点内部链接

主链接文件描述	被链接文件描述	超链接代码
网站根文件夹下的文件 index.html	链接到新闻目录 news 下的文件 news1.html	关于第一次野外活动的通知
新闻目录 news 下的文件 new1.html	链接到 news 目录上一级目录（根文件夹下）的文件为 index.html	首页
新闻目录 news 下的文件 new1.html	链接到本级目录中的文件 news2.html	关于驾照考试的通知

3. 外部链接

- 所谓外部链接，指的是跳转到当前网站外部，与其它网站中页面或其它元素之间的链接关系。这种链接的URL地址一般要用绝对路径（以http://、ftp://等开头），要有完整的URL地址。

链接对象1

- 1. **图片链接**
  - 语法：<img src=" ">
- 2. **书签链接**
  - 链接标题
  - 链接标题
  - 链接内容
- 3. **电子邮件链接**
  - 语法：链接内容

链接对象2

- 4. **FTP链接**
  - 语法： 链接内容
- 5. **下载文件的链接**
  - 语法： 链接内容

网页链接属性设置

表 5-3 网页链接属性

语法	说明
a:link	未访问过的超链接文字的样式
a:visited	已访问过的超链接文字的样式
a:hover	当鼠标移动到超链接上方时，超链接文字的样式
a:active	鼠标按下时的超链接文字的样式
a	为该标签内设置样式时，则上述四种伪类的默认值将同时引用此值

- 一个便于记忆的“爱恨原则”（LoVe/HAte），即四种伪类的首字母：LVHA。实际上用“就近原则”很好理解，或者你记住正确的顺序：a:link、a:visited、a:hover、a:active。

图像映射

- 图像映射就是在图像上先划分出不同的区域，然后定义哪个目标与图像的哪个区域对应，但单击图像的某一区域，就会把用户带到一个目标，单击另一区域，又会把用户带到另外的目标。
- 语法：
  - <img src="URL" usemap="id/name">
  - <map name="name" id="id">
  - <area shape=" " coords=" " href="URL">
  - </map>
- 说明：
  - 1）shape说明映射区域的形状，取值可以是：rect（矩形）、circle（圆形）、poly（多边形）、default（整个图像区域）。
  - 2）coords用于标识映射区域的边界。它有四个“坐标”值，分别表示区域左边缘距离图像左边缘的距离、区域上边缘距离图像上边缘的距离、区域右边缘距离图像左边缘的距离、区域下边缘距离图像上边缘的距离。
  - 3）map标记符中，name属性的取值必须与img标记符中usemap属性的取值相同，只是usemap属性的值前面多了一个#。

图像映射举例

```
<body>
<table width="600" border="0" align="center" cellpadding="0" cellspacing="0">
  <tr>
    <td><h1>柯南人物简介</h1></td>
  </tr>
  <tr>
    <td align="center"><img src="img/xiaolan.jpg"  border="0" usemap="#Map" /></td>
  </tr>
</table>

<map name="Map" id="Map">
  <area shape="rect" coords="0,0,284,394" href="lan.html" />
  <area shape="rect" coords="290,0,504,395" href="kenan.html" />
</map>
</body>
```

柯南人物简介

4 列表

- 列表（List）是指在网页中将相关资料以条目的形式有序或者无序排列而形成的表。常用的列表有无序列表（ul）、有序列表（ol）和定义列表（dl）三种。另外，还有不太常用的目录列表（dir）和菜单列表（menu）。

无序列表

- 无序列表（Unordered List）是一个没有特定顺序的列表项的集合，也称为项目列表
- 基本语法：

```
<ul>
    <li>列表项1</li>
    <li>列表项2</li>
      ……
    <li>列表项n</li>
</ul>
```

语法说明：

- 1）在HTML文件中，可以利用成对的
标记来插入无序列表，中间的列表项（list-items）标签
用来定义列表项序列。
- 2）使用无序列表标签ul的type属性（使用CSS的list-style来代替），用户可以指定出现在列表项前的项目符号的样式，其取值以及相对应的符号样式如下：
  - Disc、circle、square、none

list-style-type属性设置

- 除了特定的list-style-type属性设置列表符号的样式外，利用css（list-style-image）还可以把列表项的符号设置成自己喜欢的图片。
- 基本语法：
  - list-style-image:url | none
- 语法说明：
  - url是指定要载入的图片路径，在使用上与插入图片<img>的用法差不多；none表示不使用图片式的列表符号。

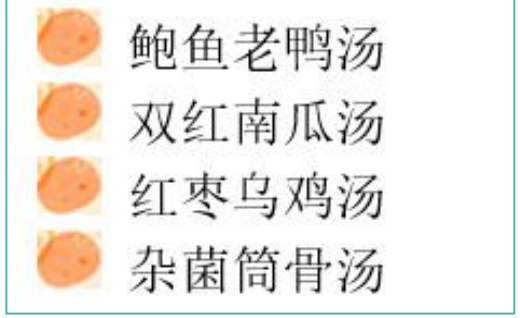
```
<ul>
    <li>鲍鱼老鸭汤</li>
    <li>双红南瓜汤</li>
    <li>红枣乌鸡汤</li>
    <li>杂菌筒骨汤</li>
</ul>
```

- 鲍鱼老鸭汤
- 双红南瓜汤
- 红枣乌鸡汤
- 杂菌筒骨汤

```
ul{ list-style-type:circle; }
```

- 鲍鱼老鸭汤
- 双红南瓜汤
- 红枣乌鸡汤
- 杂菌筒骨汤

```
list-style-image:url(image/p1.jpg);
```



列表符号位置

- list-style-position属性主要是设置每个列表项目的符号或图片，是否向外凸出。
- 基本语法：
  - list -style-position:inside | outside
- 语法说明：
  - inside表示列表符号不向外凸出，也可以理解成列表项上的第二行文字与列表符号对齐；outside表示列表符号向外凸出，也是默认值。

有序列表

- 有序列表（Ordered List）又叫编号列表，它是一个有特定顺序的列表项的集合。在有序列表中，各个列表项有先后顺序之分，它们之间以编号来标记。

1. 编号样式

- 基本语法：
  -
  - 列表项1
  - 列表项2
  - ……
  - 列表项n
  -
- 语法说明：
  - 1）在HTML文件中，可以利用成对的
标记来插入有序列表，其中间的列表项标记
用来定义列表项顺序。
  - 2）有序列表标记的type属性：list-style，用户可以指定出现在列表项前的项目编号的样式，其取值以及相对应的编号样式如下：
    - Decimal、lower-alpha、upper-alpha、lower-roman、upper-roman、none

```
<h1>古人最爱的花卉排名</h1>
    <ol>
        <li>梅花</li>
        <li>兰花</li>
        <li>菊花</li>
        <li>牡丹</li>
    </ol>
```

```
ol{list-style-type:decimal ; }
```

古人最爱的花卉排名

1. 梅花
2. 兰花
3. 菊花
4. 牡丹

```
ol{list-style-type:lower-alpha ; }
```

古人最爱的花卉排名

a. 梅花
b. 兰花
c. 菊花
d. 牡丹

2. 编号起始值

- 通常，在指定列表的编号样式后，浏览器会从“1”、“a”或“A”、“i”或“I”开始自动编号。而在使用有序列表标记的start属性后，用户则可改变编号的起始值。start属性值是一个整数，表示从哪一个数字或字母开始编号。例如，设置start="3"，则有序列表的列表项编号将从“3”、“c”、“C”、“iii”或“Ⅲ”开始编号。

```
<ol start="2">
        <li>梅花</li>
        <li>兰花</li>
        <li>菊花</li>
        <li>牡丹</li>
    </ol>
```

古人最爱的花卉排名

b. 梅花
c. 兰花
d. 菊花
e. 牡丹

3. 列表项样式

- 在列表标签<ol>的list-style和列表项标签<li>的list-style发生冲突的情况下，所指定的单个列表项遵循<li>的list-style进行显示。

4. 列表项编号

- 列表项标签<li>的list-style只能改变列表项的符号或编号的样式，并不会改变其值的大小。而使用列表项标标签<li>的value属性，可以改变当前列表项的编号大小，并会影响其后所有列表项的编号大小。但该属性只适用于有序列表。

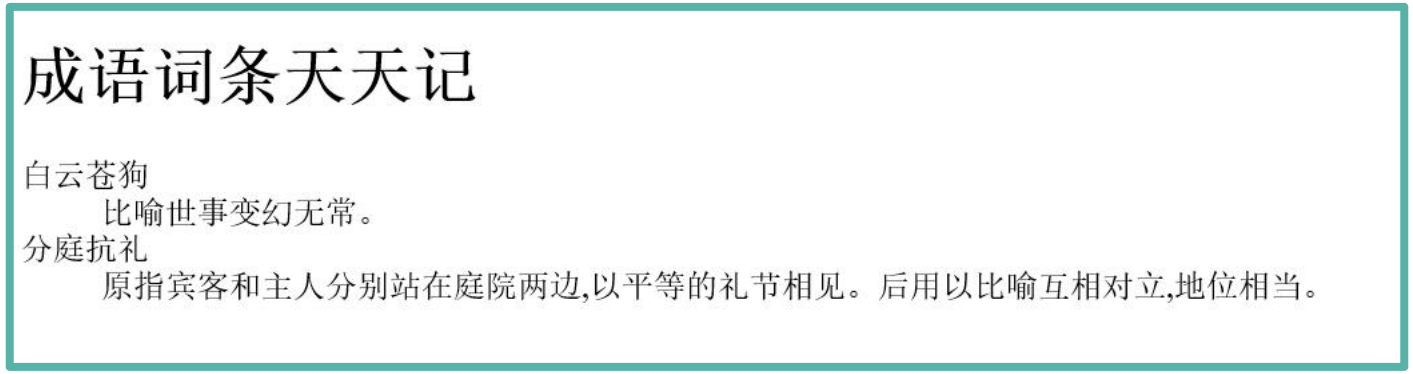
嵌套列表

- 列表还可以嵌套使用，也就是一个列表中还可以包含有多层子列表。
- 在网页文件中，对于内容层次较多的情况，使用嵌套列表不仅使网页的内容布局更加合理美观，而且使其内容看起来更加清晰明了。
- 嵌套的列表可以是无序列表的嵌套，也可以是有序列表的嵌套，还可以是无序列表和有序列表的混合嵌套。

定义列表

- 在HTML文件中，只要在适当的地方插入<dl></dl>标记，即可自动生成定义列表（Definition List）。
  - <dl>
  - <dt>…</dt>
  - <dd>…</dd>
  - …
  - <dt>…</dt>
  - <dd>…</dd>
  - …
  - </dl>
- 语法说明：
  - 1）<dl>标签用来创建定义列表。
  - 2）dt是用来创建列表中的每个元素标题，它只能在dl元素中使用。<dt>标签定义的内容将左对齐显示。
  - 3）dd用来创建列表元素的内容描述，它也只能在dl元素中使用。<dd>标签定义的内容将相对于<dt>标签定义的内容向右缩进显示。

```
<h1>成语词条天天记</h1>
    <dl>
        <dt>白云苍狗</dt>
        <dd>比喻世事变幻无常。</dd>
        <dt>分庭抗礼</dt>
        <dd>原指宾客和主人分别站在庭院两边,以平等的礼节相见。后用以比喻互相对立,地位相当。</dd>
    </dl>
```



菜单列表

- 菜单列表（Menu List）通常用于显示一个简单的单列列表，一般不做嵌套。目录列表（Directory List）通常用于显示一个多列的文件列表，可做嵌套。他们的使用与无序列表类似，并且可以看着是无序列表的一种特殊形式。早期Web标准不建议使用此两项列表（menu和dir），但在新的Web标准HTML5中重新定义了menu元素，用于排列表单控件。
- 基本语法：
    - <menu>
    - 列表项1
    - 列表项2
    - …
    - </menu>

表格

- 表格在网站应用中非常广泛，几乎所有的Web页面中都或多或少地采用表格，网页内容多数都存储在数据库中，表格用于在网页上显示二维表格式数据是最佳方式。

表格元素

- <table>
- <tr>
- <td>
- <caption>
- <th>
- <thead>
- <tbody>
- <tfoot>

表格标签

- 在HTML语法中，表格主要通过3个标签来构成：<table>、<tr>、<td>。
- 基本语法：
  - <table>
  - <tr> <td>…</td> … </tr>
  - <tr> <td>…</td> … </tr>
  - …
  - </table>
- 语法说明：
  - 1）<table>代表表格的开始，</table>代表表格的结束。
  - 2）<tr>代表行的开始，</tr>代表行的结束。
  - 3）<td></td>之间是单元格的内容，可以是文字，也可以是元素。
  - 4）在一个表格中，<tr>的个数代表表格的行数，每对<tr></tr>之间<td>的个数代表该行单元格数。

2. 表格标题

- 表格标题一般放在表格的上面，是对表格内容的简单说明，用<caption>标签实现。<caption> 标签必须紧随<table>标签之后。
- 基本语法：
  - <caption>...</caption>
- 语法说明：
  - <caption>标记之间就是标题的内容。

3. 表格表头

- 表头是指表格的第一行或第一列对表格内容的说明，文字样式为居中、加粗显示，通过<th>标签实现。
- 基本语法：
  - <table>
  - <tr> <th>…</th> …</tr>
  - <tr> <td>…</td> …</tr>
  - …
  - </table>
- 语法说明：
  - 1）<th>表头标记，包含在<tr>标记中。
  - 2）在表格中，只要把标记<td>改成<th>就可以实现表格的表头。

4. 划分表格结构

- 为了清楚区分表格结构，将一个表格分三个部分在网页上显示出来，HTML语言规定了<thead>、<tbody>、<tfoot>三个标签分别对应于表格的表首、表主体和表尾。
- 基本语法：
  - <thead>…</thead>
  - <tbody>…</tbody>
  - <tfoot>…</tfoot>

表格修饰

表 5-4 表格标签的属性

属性	描述	属性	描述
border	设置边框粗细（默认值为 0）	background	设置背景图片
bordercolor	设置边框颜色	frame	设置边框样式
bordercolorlight	设置亮边框颜色（左上边框颜色）	rules	设置内部边框样式
bordercolordark	设置暗边框颜色（右下边框颜色）	cellspacing	设置单元格间距
width	设置表格宽度	cellpadding	设置单元格边距
height	设置表格高度	align	设置表格水平对齐方式
bgcolor	设置背景颜色		

设置表格行属性

表 5-5 <tr>标签的属性

属 性	描 述	属 性	描 述
align	行内容的水平对齐	bordercolor	行的边框颜色
valign	行内容的垂直对齐	bordercolorlight	行的亮边框颜色
bgcolor	行的背景颜色	bordercolordark	行的暗边框颜色

5.5.4 设置单元格属性

表 5-6 <td>标签属性

属 性	描 述	属 性	描 述
align	单元格内容的水平对齐	bordercolordark	单元格的暗边框颜色
valign	单元格内容的垂直对齐	width	单元格的宽度
bgcolor	单元格的背景颜色	height	单元格的高度
background	单元格的背景图像	rowspan	单元格跨行
bordercolor	单元格的边框颜色	colspan	单元格跨列
bordercolorlight	单元格的亮边框颜色		

1. 设置单元格跨行

- 单元格的rowspan属性可实现单元格的跨行合并（纵向合并）。
- 基本语法：
  - <td rowspan=" ">…</td>
- 语法说明：
  - rowspan的值为单元格跨越的行数。如果创建跨越两行的单元格（即rowspan="2"），那么在下一行中就不用定义相应的单元格，如果创建跨越三行的单元格（即rowspan="3"），那么在下两行中就不用定义相应的单元格，以此类推。

2. 设置单元格跨列

- colspan属性可以进行单元格的跨列合并（横向合并）。
- 基本语法：
  - <td colspan=" ">...</td>
- 语法说明：
  - colspan的值为所跨单元格的列数。若在一行中创建跨越两列的单元格（即colspan="2"），那么在该行中就应该少定义一个单元格，若在一行中创建跨越三列的单元格（即colspan="3"），那么在该行中就少定义两个单元格，以此类推。

单线表格制作

- 表格和单元格都加上边框线后，我们发现所有的边框都是双线，而在Word等其它软件中一般都是用单线，在CSS中可以使用table的border-collapse属性实现。

表格的嵌套

- 表格嵌套就是根据插入元素的需要，在一个表格的某个单元格里再插入一个若干行和列的表格。对嵌套表格，可以像对任何其他表格一样进行格式设置，但是其宽度受它所在单元格的宽度的限制。利用表格的嵌套，一方面可以编辑出复杂而精美的效果，另一方面可根据布局需要来实现精确的编排。不过，需要注意的是，嵌套层次越多，网页的载入速度就会越慢。所以，不建议使用表格的嵌套功能。

内联框架

- iframe元素创建包含另外一个文档的内联框架（即行内框架）

表 5-7 内联框架属性

属性	值	描述
src	URL	规定在 iframe 中显示的文档的 URL
srcdoc	HTML 代码	规定在<iframe>中显示的页面的 HTML 内容
name	frame_name	规定 iframe 的名称
sandbox	"" allow-forms allow-same-origin allow-scripts allow-top-navigation	启用一系列对 <iframe> 中内容的额外限制
allowfullscreen		是否允许 iframe 的内容使用 requestFullscreen()
width	pixels 或%	定义 iframe 的宽度
height	pixels 或%	规定 iframe 的高度

小结

- 本章主要介绍一些常用网页元素的使用方法，包括特殊符号、图片、超链接、列表、表格和内联框架等。图片的使用是美化网页的最重要手段，也是传递信息的另一种方式，文字传播的信息更准确，图片传递的信息更丰富，能用文字表达清楚的地方就没有必要用图片，所以在网页中不要滥用图片。超链接可以指向网络上的任何资源一张HTML页面、一幅图像、一个声音或视频文件等。列表的作用不仅仅用来呈现列表，更重要的任务格式化多项内容的显示，比如导航菜单。使用表格的目的是用来显示二维式数据和排版简单元素，千万不要用来布局。

谢　谢　！

web开发

主讲老师：徐晓丹

浙 江 师 范 大 学