

# CSS

张利



# CSS

- CSS (**C**ascading **S**tyle **S**heets) 层叠样式表， 用于控制网页的样式和布局， css3是css的最新标准。
- CSS 规则由两个主要的部分构成：选择器， 以及一条或多条声明。
  - selector {declaration1; declaration2; ... declarationN }
- CSS的三种形态：
  - 外部样式表： <link rel="stylesheet" type="text/css" href="mystyle.css" />
  - 内部样式表：

```
<style type="text/css">
  hr {color: sienna;}
  p {margin-left: 20px;}
  body {background-image: url("images/back40.gif");}
</style>
```
  - 内联样式：

```
<p style="color: sienna; margin-left: 20px">
  This is a paragraph
</p>
```



# CSS 选择器

- 元素选择器（类型选择器）
- 分组选择器， 用逗号相隔

```
div, p{  
    background-color: gray  
}
```

- 通配符选择器 \*
- 类（class）选择器
  - 结合元素选择器， eg: p.important {color:red;} 会匹配 class 属性包含 important 的所有 p 元素
  - CSS 多类选择器， eg: <p class="important warning">
- ID 选择器
- 属性选择器
  - a[href] {color:red;}
  - a[href="http://www.w3school.com.cn/about\_us.asp"] {color: red;}
  - a[href="http://www.w3school.com.cn/"][title="W3School"] {color: red;}
  - p[class="important warning"] {color: red;}



# CSS 选择器

- 后代选择器

- 语法: `div p`
- 两个元素之间的层次间隔可以是无限的。

- 子元素选择器

- 使用 `>`
- 相比于后代选择器，子元素选择器只会选择第一级的元素(直接后代)
- 练习： 后代选择器和子选择器
  - `table.company td > p`
  - 上面的选择器会选择作为 `td` 元素子元素的所有 `p` 元素，这个 `td` 元素本身从 `table` 元素继承，该 `table` 元素有一个包含 `company` 的 `class` 属性。



# CSS选择器

选择器	例子	例子描述	CSS
<u>.class</u>	.intro	选择 class="intro" 的所有元素。	1
<u>#id</u>	#firstname	选择 id="firstname" 的所有元素。	1
<u>*</u>	*	选择所有元素。	2
<u>element</u>	p	选择所有 <p> 元素。	1
<u>element,element</u>	div,p	选择所有 <div> 元素和所有 <p> 元素。	1
<u>element element</u>	div p	选择 <div> 元素内部的所有 <p> 元素。	1
<u>element&gt;element</u>	div>p	选择父元素为 <div> 元素的所有 <p> 元素。	2
<u>element+element</u>	div+p	选择紧接在 <div> 元素之后的所有 <p> 元素。	2
<u>[attribute]</u>	[target]	选择带有 target 属性所有元素。	2
<u>[attribute=value]</u>	[target=_blank]	选择 target="_blank" 的所有元素。	2
<u>[attribute~=value]</u>	[title~=flower]	选择 title 属性包含单词 "flower" 的所有元素。	2
<u>[attribute =value]</u>	[lang =en]	选择 lang 属性值以 "en" 开头的元素。	2
<u>:link</u>	a:link	选择所有未被访问的链接。	1
<u>:visited</u>	a:visited	选择所有已被访问的链接。	1
<u>:active</u>	a:active	选择活动链接。	1
<u>:hover</u>	a:hover	选择鼠标指针位于其上的链接。	1
<u>:focus</u>	input:focus	选择获得焦点的 input 元素。	2
<u>:first-letter</u>	p:first-letter	选择每个 <p> 元素的首字母。	1



# CSS选择器

<u>:first-line</u>	p:first-line	选择每个 <p> 元素的首行。	1
<u>:first-child</u>	p:first-child	选择属于父元素的第一个子元素的每个 <p> 元素。	2
<u>:before</u>	p:before	在每个 <p> 元素的内容之前插入内容。	2
<u>:after</u>	p:after	在每个 <p> 元素的内容之后插入内容。	2
<u>:lang(<i>language</i>)</u>	p:lang(it)	选择带有以 "it" 开头的 lang 属性值的每个 <p> 元素。	2
<u><i>element1~element2</i></u>	p~ul	选择前面有 <p> 元素的每个 <ul> 元素。	3
<u>[<i>attribute</i>^=<i>value</i>]</u>	a[src^="https"]	选择其 src 属性值以 "https" 开头的每个 <a> 元素。	3
<u>[<i>attribute</i>\$=<i>value</i>]</u>	a[src\$=".pdf"]	选择其 src 属性以 ".pdf" 结尾的所有 <a> 元素。	3
<u>[<i>attribute</i>*=<i>value</i>]</u>	a[src*="abc"]	选择其 src 属性中包含 "abc" 子串的每个 <a> 元素。	3
<u>:first-of-type</u>	p:first-of-type	选择属于其父元素的首个 <p> 元素的每个 <p> 元素。	3
<u>:last-of-type</u>	p:last-of-type	选择属于其父元素的最后 <p> 元素的每个 <p> 元素。	3
<u>:only-of-type</u>	p:only-of-type	选择属于其父元素唯一的 <p> 元素的每个 <p> 元素。	3
<u>:only-child</u>	p:only-child	选择属于其父元素的唯一子元素的每个 <p> 元素。	3
<u>:nth-child(<i>n</i>)</u>	p:nth-child(2)	选择属于其父元素的第二个子元素的每个 <p> 元素。	3
<u>:nth-last-child(<i>n</i>)</u>	p:nth-last-child(2)	同上，从最后一个子元素开始计数。	3
<u>:nth-of-type(<i>n</i>)</u>	p:nth-of-type(2)	选择属于其父元素第二个 <p> 元素的每个 <p> 元素。	3
<u>:nth-last-of-type(<i>n</i>)</u>	p:nth-last-of-type(2)	同上，但是从最后一个子元素开始计数。	3
<u>:last-child</u>	p:last-child	选择属于其父元素最后一个子元素每个 <p> 元素。	3
<u>:root</u>	:root	选择文档的根元素。	3
<u>:empty</u>	p:empty	选择没有子元素的每个 <p> 元素（包括文本节点）。	3


















# CSS选择器

<u>:target</u>	#news:target	选择当前活动的 #news 元素。	3
<u>:enabled</u>	input:enabled	选择每个启用的 <input> 元素。	3
<u>:disabled</u>	input:disabled	选择每个禁用的 <input> 元素	3
<u>:checked</u>	input:checked	选择每个被选中的 <input> 元素。	3
<u>:not(selector)</u>	:not(p)	选择非 <p> 元素的每个元素。	3
<u>::selection</u>	::selection	选择被用户选取的元素部分。	3



# CSS3 新特性 - 边框

- 浏览器支持

属性	浏览器支持				
border-radius					
box-shadow					
border-image					

- border-radius
  - border-radius: npx
  - border-direction-direction-radius: npx
- box-shadow
  - box-shadow: h-shadow v-shadow blur spread color inset;
- border-image
  - border-image:url(border.png) 30 30 round;



# CSS3 新特性 - 背景

- background-size 属性规定背景图片的尺寸。
- 在 CSS3 之前，背景图片的尺寸是由图片的实际尺寸决定的。在 CSS3 中，可以规定背景图片的尺寸
- 多重背景：

```
body
{
    background-image:url(bg_flower.gif),url(bg_flower_2.gif);
}
```



# CSS3 新特性 - 文本效果

- `text-shadow`: 文本阴影
  - 语法: `text-shadow: h-shadow v-shadow blur color;`
- `word-wrap`: 自动换行
  - 属性允许您允许文本强制文本进行换行 – 即使这意味着会对单词进行拆分。
  - 语法: `word-wrap: normal|break-word;`



# CSS3 新特性 — 转换

- 通过 CSS3 转换，我们能够对元素进行移动、缩放、转动、拉长或拉伸。
- 语法：transform: none|transform-functions;
  - #Chrome 和 Safari 需要前缀 -webkit-。
- transform-functions 包括：
  - rotate(angle): 旋转， 参数angle定义角度。Eg: 90deg
  - translate(x, y): 移动， 沿着X和Y轴移动元素。
  - translateX(n)
  - translateY(n)
  - scale(x, y): 缩放转换， 改变元素的宽度和高度。
  - scaleX(n)
  - scaleY(n)
  - skew(x-angle,y-angle): 倾斜， 沿着X轴和Y轴。
  - matrix(n,n,n,n,n,n): 包含旋转， 缩放， 以及移动



# CSS3 新特性 – 过度

- 通过 CSS3，我们可以在不使用 Flash 动画或 JavaScript 的情况下，当元素从一种样式变换为另一种样式时为元素添加效果。
- 要实现这一点，必须规定两项内容：
  - 规定您希望把效果添加到哪个 CSS 属性上
  - 规定效果的时长
- 语法：transition: property duration timing-function delay;
  - Safari 需要前缀 -webkit-。



# CSS3 新特性 – 动画

- 通过 CSS3，我们能够创建动画，这可以在许多网页中取代动画图片、Flash 动画以及 JavaScript。
  - 通过规定至少以下两项 CSS3 动画属性，即可将动画绑定到选择器：
    - 规定动画的名称
    - 规定动画的时长
  - 首先定义keyframes确定动画的轨迹，然后定义animation属性引用轨迹。
  - 连续动画使用时间百分比来控制。
- 
- 注意：Chrome 和 Safari 需要前缀 `-webkit-`。



# CSS3 新特性 – 多列

- 通过 CSS3，您能够创建多个列来对文本进行布局 – 就像报纸那样！
- 属性：
  - `column-count: n`：规定元素应该被分隔的列数。
  - `column-gap: npx`：规定列之间的间隔
  - `column-width: npx`：规定列的宽度，会和`column-count`冲突
  - `column-rule(width, style, color)`：规定列的样式，width是分割线的宽度。
- 注意：Firefox 需要前缀 `-moz-`。Chrome 和 Safari 需要前缀 `-webkit-`。