

CSS Id 和 Class

id 和 class 选择器

如果你要在HTML元素中设置CSS样式，你需要在元素中设置"id" 和 "class"选择器。

id 选择器

id 选择器可以为标有特定 id 的 HTML 元素指定特定的样式。

HTML元素以id属性来设置id选择器,CSS 中 id 选择器以 "#" 来定义。

以下的样式规则应用于元素属性 id="para1":

实例

```
#para1{ text-align:center; color:red;}
```

尝试一下 »



ID属性不要以数字开头，数字开头的ID在 Mozilla/Firefox 浏览器中不起作用。

class 选择器

class 选择器用于描述一组元素的样式，class 选择器有别于id选择器，class可以在多个元素中使用。

class 选择器在HTML中以class属性表示, 在 CSS 中，类选择器以一个点"."号显示：

在以下的例子中，所有拥有 center 类的 HTML 元素均为居中。

实例

```
.center {text-align:center;}
```

尝试一下 »

你也可以指定特定的HTML元素使用class。

在以下实例中, 所有的 p 元素使用 class="center" 让该元素的文本居中:

实例

```
p.center {text-align:center;}
```

尝试一下 »



类名的第一个字符不能使用数字！它无法在 Mozilla 或 Firefox 中起作用。

CSS 组合选择符

CSS 组合选择符

	组合选择符说明了两个选择器之间的关系。

CSS组合选择符包括各种简单选择符的组合方式。

在 CSS3 中包含了四种组合方式:

- 后代选择器(以空格 分隔)
- 子元素选择器(以大于 > 号分隔)
- 相邻兄弟选择器 (以加号 + 分隔)
- 普通兄弟选择器 (以波浪号 ~ 分隔)

后代选择器

后代选择器用于选取某元素的后代元素。

以下实例选取所有

元素插入到

元素中:

实例

```
div p{ background-color:yellow;}
```

尝试一下 »

子元素选择器

与后代选择器相比，子元素选择器（Child selectors）只能选择作为某元素直接/一级子元素的元素。

以下实例选择了

元素中所有直接子元素

:

实例

```
div>p{ background-color:yellow;}
```

尝试一下 »

相邻兄弟选择器

相邻兄弟选择器 (Adjacent sibling selector) 可选择紧接在另一元素后的元素，且二者有相同父元素。

如果需要选择紧接在另一个元素后的元素，而且二者有相同的父元素，可以使用相邻兄弟选择器 (Adjacent sibling selector) 。

以下实例选取了所有位于

元素后的第一个

元素：

实例

```
div+p{ background-color:yellow;}
```

尝试一下 »

后续兄弟选择器

后续兄弟选择器选取所有指定元素之后的相邻兄弟元素。

以下实例选取了所有

元素之后的所有相邻兄弟元素

：

实例

```
div~p{ background-color:yellow;}
```

尝试一下 »

CSS 伪类(Pseudo-classes)

CSS伪类是用来添加一些选择器的特殊效果。

语法

伪类的语法：

```
selector:pseudo-class {property:value;}
```

CSS类也可以使用伪类：

selector.class:pseudo-class {property:value;}

anchor伪类

在支持 CSS 的浏览器中，链接的不同状态都可以以不同的方式显示

实例

`a:link {color:#FF0000;} /* 未访问的链接 /a:visited {color:#00FF00;} / 已访问的链接 /a:hover {color:#FF00FF;} / 鼠标划过链接 /a:active {color:#0000FF;} / 已选中的链接 */`

尝试一下 »

注意： 在CSS定义中，a:hover 必须被置于 a:link 和 a:visited 之后，才是有效的。

注意： 在 CSS 定义中，a:active 必须被置于 a:hover 之后，才是有效的。

注意： 伪类的名称不区分大小写。

伪类和CSS类

伪类可以与 CSS 类配合使用：

`a.red:visited {color:#FF0000;}` [CSS 语法](#)

如果在上面的例子的链接已被访问，它会显示为红色。

CSS :first-child 伪类

您可以使用 :first-child 伪类来选择父元素的第一个子元素。

注意： 在IE8的之前版本必须声明，这样 :first-child 才能生效。

匹配第一个

元素

在下面的例子中，选择器匹配作为任何元素的第一个子元素的

元素：

实例

`p:first-child{ color:blue;}`

尝试一下 »

匹配所有

元素中的第一个 元素

在下面的例子中，选择相匹配的所有

元素的第一个 元素：

实例

```
p > i:first-child{ color:blue;}
```

尝试一下»

匹配所有作为第一个子元素的

元素中的所有 元素

在下面的例子中，选择器匹配所有作为元素的第一个子元素的

元素中的所有 元素：

实例

```
p:first-child i{ color:blue;}
```

尝试一下»

CSS - :lang 伪类

:lang 伪类使你有能力为不同的语言定义特殊的规则

注意：IE8必须声明才能支持lang伪类。

在下面的例子中，:lang 类为属性值为 no 的q元素定义引号的类型：

实例

```
q:lang(no) {quotes: "~" "~";}
```

尝试一下»

更多实例

[为超链接添加不同样式](#) 这个例子演示了如何为超链接添加其他样式。

[使用:focus](#) 这个例子演示了如何使用:focus伪类。

所有CSS伪类/元素

选择器	示例	示例说明
:checked	input:checked	选择所有选中的表单元素
:disabled	input:disabled	选择所有禁用的表单元素
:empty	p:empty	选择所有没有子元素的p元素
:enabled	input:enabled	选择所有启用的表单元素
:first-of-type	p:first-of-type	选择的每个 p 元素是其父元素的第一个 p 元素
:in-range	input:in-range	选择元素指定范围内的值
:invalid	input:invalid	选择所有无效的元素
:last-child	p:last-child	选择所有p元素的最后一个子元素
:last-of-type	p:last-of-type	选择每个p元素是其母元素的最后一个p元素
:not(selector)	:not(p)	选择所有p以外的元素
:nth-child(n)	p:nth-child(2)	选择所有 p 元素的父元素的第二个子元素
:nth-last-child(n)	p:nth-last-child(2)	选择所有p元素倒数的第二个子元素
:nth-last-of-type(n)	p:nth-last-of-type(2)	选择所有p元素倒数的第二个为p的子元素
:nth-of-type(n)	p:nth-of-type(2)	选择所有p元素第二个为p的子元素
:only-of-type	p:only-of-type	选择所有仅有一个子元素为p的元素
:only-child	p:only-child	选择所有仅有一个子元素的p元素
:optional	input:optional	选择没有"required"的元素属性
:out-of-range	input:out-of-range	选择指定范围以外的值的元素属性
:read-only	input:read-only	选择只读属性的元素属性
:read-write	input:read-write	选择没有只读属性的元素属性
:required	input:required	选择有"required"属性指定的元素属性
:root	root	选择文档的根元素
:target	#news:target	选择当前活动#news元素(点击URL包含锚的名字)
:valid	input:valid	选择所有有效值的属性
:link	a:link	选择所有未访问链接
:visited	a:visited	选择所有访问过的链接
:active	a:active	选择正在活动链接

选择器	示例	示例说明
:hover	a:hover	把鼠标放在链接上的状态
:focus	input:focus	选择元素输入后具有焦点
:first-letter	p:first-letter	选择每个 元素的第一个字母
:first-line	p:first-line	选择每个 元素的第一行
:first-child	p:first-child	选择器匹配属于任意元素的第一个子元素的 元素
:before	p:before	在每个 元素之前插入内容
:after	p:after	在每个 元素之后插入内容
:lang(language)	p:lang(it)	为 元素的lang属性选择一个开始值

CSS 伪元素

CSS 伪元素是用来添加一些选择器的特殊效果。

语法

伪元素的语法：

```
1 selector:pseudo-element {property:value;}
```

CSS类也可以使用伪元素：

```
1 selector.class:pseudo-element {property:value;}
```

:first-line 伪元素

"first-line" 伪元素用于向文本的首行设置特殊样式。

在下面的例子中，浏览器会根据"first-line" 伪元素中的样式对p 元素的第一行文本进行格式化：

实例

```
p:first-line { color:#ff0000; font-variant:small-caps;}
```

尝试一下»

注意： "first-line" 伪元素只能用于块级元素。

注意： 下面的属性可应用于"first-line" 伪元素：

- font properties
- color properties
- background properties
- word-spacing
- letter-spacing
- text-decoration
- vertical-align
- text-transform
- line-height
- clear

:first-letter 伪元素

"first-letter" 伪元素用于向文本的首字母设置特殊样式：

实例

```
p:first-letter { color:#ff0000; font-size:xx-large;}
```

尝试一下»

注意： "first-letter" 伪元素只能用于块级元素。

注意： 下面的属性可应用于"first-letter" 伪元素：

- font properties
- color properties
- background properties
- margin properties
- padding properties
- border properties
- text-decoration
- vertical-align (only if "float" is "none")
- text-transform
- line-height
- float

- clear

伪元素和CSS类

伪元素可以结合CSS类:

```
1 p.article:first-letter {color:#ff0000;}
2
3 <p class="article">文章段落</p>
```

上面的例子会使所有 class 为 article 的段落的首字母变为红色。

多个伪元素

可以结合多个伪元素来使用。

在下面的例子中，段落的第一个字母将显示为红色，其字体大小为 xx-large。第一行中的其余文本将为蓝色，并以小型大写字母显示。

段落中的其余文本将以默认字体大小和颜色来显示:

实例

```
p:first-letter{ color:#ff0000; font-size:xx-large;}p:first-line { color:#0000ff; font-variant:small-caps;}
```

尝试一下»

CSS - :before 伪元素

":before" 伪元素可以在元素的内容前面插入新内容。

下面的例子在每个

元素前面插入一幅图片:

实例

```
h1:before { content:url(smiley.gif);}
```

尝试一下»

CSS - :after 伪元素

":after" 伪元素可以在元素的内容之后插入新内容。

下面的例子在每个

元素后面插入一幅图片：

实例

```
h1:after{ content:url(smiley.gif);}
```

尝试一下»

所有CSS伪类/元素

选择器	示例	示例说明
<u>:link</u>	a:link	选择所有未访问链接
<u>:visited</u>	a:visited	选择所有访问过的链接
<u>:active</u>	a:active	选择正在活动链接
<u>:hover</u>	a:hover	把鼠标放在链接上的状态
<u>:focus</u>	input:focus	选择元素输入后具有焦点
<u>:first-letter</u>	p:first-letter	选择每个元素的第一个字母
<u>:first-line</u>	p:first-line	选择每个元素的第一行

选择器	示例	示例说明
<u>:first-child</u>	<i>p:first-child</i>	选择器匹配属于任意元素的第一个子元素的元素
<u>:before</u>	<i>p:before</i>	在每个元素之前插入内容
<u>:after</u>	<i>p:after</i>	在每个元素之后插入内容
<u>:lang(language)</u>	<i>p:lang(it)</i>	为元素的lang属性选择一个开始值

CSS 属性选择器

具有特定属性的HTML元素样式

具有特定属性的HTML元素样式不仅仅是class和id。

注意：IE7和IE8需声明DOCTYPE才支持属性选择器！IE6和更低的版本不支持属性选择器。

属性选择器

下面的例子是把包含标题（title）的所有元素变为蓝色：

实例

```
[title]{ color:blue;}
```

尝试一下»

属性和值选择器

下面的实例改变了标题title='runoob'元素的边框样式:

实例

```
[title=runoob]{ border:5px solid green;}
```

尝试一下»

属性和值的选择器 - 多值

下面是包含指定值的title属性的元素样式的例子, 使用 (~) 分隔属性和值:

实例

```
[title~=hello] { color:blue; }
```

尝试一下»

下面是包含指定值的lang属性的元素样式的例子, 使用 (|) 分隔属性和值:

实例

```
[lang|=en] { color:blue; }
```

尝试一下»

表单样式

属性选择器样式无需使用class或id的形式:

实例

```
input[type="text">{ width:150px; display:block;  
margin-bottom:10px; background-  
color:yellow;}input[type="button">{  
width:120px; margin-left:35px; display:block;}
```

CSS 属性选择器 ~, |, ^, \$, * 的区别

先上总结:

"value 是完整单词" 类型的比较符号: ~, |

"拼接字符串" 类型的比较符号: *, ^, \$

1.attribute 属性中包含 value:

[attribute~=value] 属性中包含独立的单词为 value, 例如:

```
1 [title~=flower] --> 
```

[attribute*=value] 属性中做字符串拆分, 只要能拆出来 value 这个词就行, 例如:

```
1 [title*=flower] --> 
```

2.attribute 属性以 value 开头:

[attribute|=value] 属性中必须是完整且唯一的单词, 或者以 - 分隔开: , 例如:

```
1 [lang|=en] --> <p lang="en"> <p  
lang="en-us">
```

attribute^=value] 属性的前几个字母是 value 就可以, 例如:

```
1 [lang^=en] --> <p lang="ennn">
```

3.attribute 属性以 value 结尾:

```
1 [attribute$=value] 属性的后几个字母是  
value 就可以, 例如:
```

```
1 a[src$=".pdf"]
```


CSS 选择器

CSS选择器用于选择你想要的元素的样式的模式。

"CSS"列表示在CSS版本的属性定义（CSS1，CSS2，或对CSS3）。

选择器	示例	示例说明	CSS
<u>.class</u>	.intro	选择所有class="intro"的元素	1
<u>#id</u>	#firstname	选择所有id="firstname"的元素	1
<u>*</u>	*	选择所有元素	2
element	p	选择所有元素	1
element,element	div,p	选择所有元素和元素	1
<u>element element</u>	div p	选择元素内的所有元素	1
<u>element>element</u>	div>p	选择所有父级是元素的元素	2
<u>element+element</u>	div+p	选择所有紧跟在元素之后的第一个元素	2
<u>[attribute]</u>	[target]	选择所有带有target属性元素	2
<u>[attribute=value]</u>	[target=-blank]	选择所有使用target="-blank"的元素	2
<u>[attribute~=value]</u>	[title~=flower]	选择标题属性包含单词"flower"的所有元素	2
<u>[attribute =language]</u>	[lang =en]	选择lang 属性等于en, 或者以en- 为开头的所有元素	2
<u>:link</u>	a:link	选择所有未访问链接	1
<u>:visited</u>	a:visited	选择所有访问过的链接	1
<u>:active</u>	a:active	选择活动链接	1
<u>:hover</u>	a:hover	选择鼠标在链接上面时	1
<u>:focus</u>	input:focus	选择具有焦点的输入元素	2
<u>:first-letter</u>	p:first-letter	选择每一个元素的第一个字母	1

选择器	示例	示例说明	CSS
<u>:first-line</u>	<code>p:first-line</code>	选择每一个元素的第一行	1
<u>:first-child</u>	<code>p:first-child</code>	指定只有当元素是其父级的第一个子级的样式。	2
<u>:before</u>	<code>p:before</code>	在每个元素之前插入内容	2
<u>:after</u>	<code>p:after</code>	在每个元素之后插入内容	2
<u>:lang(language)</u>	<code>p:lang(it)</code>	选择一个lang属性的起始值="it"的所有元素	2
<u>element1~element2</u>	<code>p~ul</code>	选择p元素之后的每一个ul元素	3
<u>[attribute^=value]</u>	<code>a[src^="https"]</code>	选择每一个src属性的值以"https"开头的元素	3
<u>[attribute\$=value]</u>	<code>a[src\$=".pdf"]</code>	选择每一个src属性的值以".pdf"结尾的元素	3
<u>[attribute**=value*]</u>	<code>a[src*="runoob"]</code>	选择每一个src属性的值包含子字符串"runoob"的元素	3
<u>:first-of-type</u>	<code>p:first-of-type</code>	选择每个p元素是其父级的第一个p元素	3
<u>:last-of-type</u>	<code>p:last-of-type</code>	选择每个p元素是其父级的最后一个p元素	3
<u>:only-of-type</u>	<code>p:only-of-type</code>	选择每个p元素是其父级的唯一p元素	3
<u>:only-child</u>	<code>p:only-child</code>	选择每个p元素是其父级的唯一子元素	3
<u>:nth-child(n)</u>	<code>p:nth-child(2)</code>	选择每个p元素是其父级的第二个子元素	3
<u>:nth-last-child(n)</u>	<code>p:nth-last-child(2)</code>	选择每个p元素的是其父级的第二个子元素，从最后一个子项计数	3
<u>:nth-of-type(n)</u>	<code>p:nth-of-type(2)</code>	选择每个p元素是其父级的第二个p元素	3

选择器	示例	示例说明	CSS
<u>:nth-last-of-type(n)</u>	<code>p:nth-last-of-type(2)</code>	选择每个p元素的是其父级的第二个p元素，从最后一个子项计数	3
<u>:last-child</u>	<code>p:last-child</code>	选择每个p元素是其父级的最后一个子级。	3
<u>:root</u>	<code>:root</code>	选择文档的根元素	3
<u>:empty</u>	<code>p:empty</code>	选择每个没有任何子级的p元素 (包括文本节点)	3
<u>:target</u>	<code>#news:target</code>	选择当前活动的#news元素 (包含该锚名称的点击的URL)	3
<u>:enabled</u>	<code>input:enabled</code>	选择每一个已启用的输入元素	3
<u>:disabled</u>	<code>input:disabled</code>	选择每一个禁用的输入元素	3
<u>:checked</u>	<code>input:checked</code>	选择每个选中的输入元素	3
<u>:not(selector)</u>	<code>:not(p)</code>	选择每个并非p元素的元素	3
<u>::selection</u>	<code>::selection</code>	匹配元素中被用户选中或处于高亮状态的部分	3
<u>:out-of-range</u>	<code>:out-of-range</code>	匹配值在指定区间之外的input元素	3
<u>:in-range</u>	<code>:in-range</code>	匹配值在指定区间之内的input元素	3
<u>:read-write</u>	<code>:read-write</code>	用于匹配可读及可写的元素	3
<u>:read-only</u>	<code>:read-only</code>	用于匹配设置 "readonly" (只读) 属性的元素	3
<u>:optional</u>	<code>:optional</code>	用于匹配可选的输入元素	3
<u>:required</u>	<code>:required</code>	用于匹配设置了 "required" 属性的元素	3
<u>:valid</u>	<code>:valid</code>	用于匹配输入值为合法的元素	3
<u>:invalid</u>	<code>:invalid</code>	用于匹配输入值为非法的元素	3