|  |  |  |
| --- | --- | --- |
| **选择器** | **示例** | **示例说明** |
| [:checked](http://www.runoob.com/cssref/sel-checked.html) | input:checked | 选择所有选中的表单元素 |
| [:disabled](http://www.runoob.com/css/cssref/sel-disabled.html) | input:disabled | 选择所有禁用的表单元素 |
| [:empty](http://www.runoob.com/cssref/sel-empty.html) | p:empty | 选择所有没有子元素的p元素 |
| [:enabled](http://www.runoob.com/cssref/sel-enable.html) | input:enabled | 选择所有启用的表单元素 |
| [:first-of-type](http://www.runoob.com/cssref/sel-first-of-type.html) | p:first-of-type | 选择每个父元素是p元素的第一个p子元素 |
| [:in-range](http://www.runoob.com/cssref/sel-in-range.html) | input:in-range | 选择元素指定范围内的值 |
| [:invalid](http://www.runoob.com/cssref/sel-invalid.html) | input:invalid | 选择所有无效的元素 |
| [:last-child](http://www.runoob.com/cssref/sel-last-child.html) | p:last-child | 选择所有p元素的最后一个子元素 |
| [:last-of-type](http://www.runoob.com/cssref/sel-last-of-type.html) | 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)](http://www.runoob.com/cssref/sel-not.html) | :not(p) | 选择所有p以外的元素 |
| [:nth-child(n)](http://www.runoob.com/cssref/sel-nth-child.html) | p:nth-child(2) | 选择所有p元素的第二个子元素 |
| [:nth-last-child(n)](http://www.runoob.com/cssref/sel-nth-last-child.html) | 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)](http://www.runoob.com/cssref/sel-nth-last-of-type.html) | 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)](http://www.runoob.com/cssref/sel-nth-of-type.html) | p:nth-of-type(2) | 选择所有p元素第二个为p的子元素 |
| [:only-of-type](http://www.runoob.com/cssref/sel-only-of-type.html) | p:only-of-type | 选择所有仅有一个子元素为p的元素 |
| [:only-child](http://www.runoob.com/cssref/sel-only-child.html) | p:only-child | 选择所有仅有一个子元素的p元素 |
| [:optional](http://www.runoob.com/cssref/sel-optional.html) | input:optional | 选择没有"required"的元素属性 |
| [:out-of-range](http://www.runoob.com/cssref/sel-out-of-range.html) | input:out-of-range | 选择指定范围以外的值的元素属性 |
| [:read-only](http://www.runoob.com/cssref/sel-read-only.html) | input:read-only | 选择只读属性的元素属性 |
| [:read-write](http://www.runoob.com/cssref/sel-read-write.html) | input:read-write | 选择没有只读属性的元素属性 |
| [:required](http://www.runoob.com/cssref/sel-required.html) | input:required | 选择有"required"属性指定的元素属性 |
| [:root](http://www.runoob.com/cssref/sel-root.html) | root | 选择文档的根元素 |
| [:target](http://www.runoob.com/cssref/sel-target.html) | #news:target | 选择当前活动#news元素(点击URL包含锚的名字) |
| [:valid](http://www.runoob.com/cssref/sel-valid.html) | input:valid | 选择所有有效值的属性 |
| [:link](http://www.runoob.com/cssref/sel-link.html) | a:link | 选择所有未访问链接 |
| [:visited](http://www.runoob.com/cssref/sel-visited.html) | a:visited | 选择所有访问过的链接 |
| [:active](http://www.runoob.com/cssref/sel-active.html) | a:active | 选择正在活动链接 |
| [:hover](http://www.runoob.com/cssref/sel-hover.html) | a:hover | 把鼠标放在链接上的状态 |
| [:focus](http://www.runoob.com/cssref/sel-focus.html) | input:focus | 选择元素输入后具有焦点 |
| [:first-letter](http://www.runoob.com/cssref/sel-firstletter.html) | p:first-letter | 选择每个<p> 元素的第一个字母 |
| [:first-line](http://www.runoob.com/cssref/sel-firstline.html) | p:first-line | 选择每个<p> 元素的第一行 |
| [:first-child](http://www.runoob.com/cssref/sel-firstchild.html) | p:first-child | 选择器匹配属于任意元素的第一个子元素的 <p> 元素 |
| [:before](http://www.runoob.com/cssref/sel-before.html) | p:before | 在每个<p>元素之前插入内容 |
| [:after](http://www.runoob.com/cssref/sel-after.html) | p:after | 在每个<p>元素之后插入内容 |
| [:lang(*language*)](http://www.runoob.com/cssref/sel-lang.html) | p:lang(it) | 为<p>元素的lang属性选择一个开始值 |

**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。

**CSS - :first - child伪类**

您可以使用 :first-child 伪类来选择元素的第一个子元素

**注意：**在IE8的之前版本必须声明[<!DOCTYPE>](http://www.runoob.com/tags/tag-doctype.html) ，这样 :first-child 才能生效。

**匹配第一个 <p> 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 <p> 元素：

**实例**

p:first-child { color:blue; }

**匹配所有<p> 元素中的第一个 <i> 元素**

在下面的例子中，选择相匹配的所有<p>元素的第一个 <i> 元素：

**实例**

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

**匹配所有作为第一个子元素的 <p> 元素中的所有 <i> 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 <p> 元素中的所有 <i> 元素：

**实例**

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

**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[<!DOCTYPE>](http://www.runoob.com/tags/tag-doctype.html)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: "~" "~";}

对于:nth-child选择器，意味着选择一个元素如果：

1. 这是个段落元素
2. 这是父标签的第二个孩子元素

对于:nth-of-type选择器，意味着选择一个元素如果：

1. 选择父标签的第二个段落子元素

☆ CSS3新添加的::selection伪元素

::selection伪元素：用来匹配突出显示的文本(用鼠标选择文本时的文本)。浏览器默认情况下，用鼠标选择网页文本是以“深蓝的背

景，白色的字体”显示的。

1、IE9+、Opera、Google Chrome 以及 Safari 中支持 ::selection 选择器。

2、Firefox 支持替代的 ::-moz-selection选择器。

::selection{

background: orange;

color: white;

}

::-moz-selection{

background: orange;

color: white;

}

☆ 伪类和伪元素的特性及区别

伪类本质上是为了弥补常规CSS选择器的不足，以便获取到更多信息；

伪元素本质上是创建了一个有内容的虚拟容器；

CSS3中伪类和伪元素的语法不同；

可以同时使用多个伪类，而只能同时使用一个伪元素