# CSS选择器

## 属性选择器

*/\* 忽略大小写 \*/*

a[href\*="insensitive" i] {

color: cyan;

}

*/\* 匹配大小写 \*/*

a[href\*="cAsE" s] {

color: purple;

}

## 嵌套选择器和&

### 嵌套选择器

.parentRule {

/\* parent rule style properties \*/

color: yellow;

.childRule {

/\* child rule style properties \*/

color: red;

}

}

### &

&象征上一级选择器

.parentRule2 {

/\* parent rule style properties \*/

&:hover {

/\* child rule style properties \*/

color: red;

}

}

## 标签选择器

### namespace

@namespace svg url(http://www.w3.org/2000/svg);

svg|text {

fill: blue;

}

## 列组合符

col.selected || td {

background: gray;

color: white;

font-weight: bold;

}

## 命名空间分隔符

@namespace svg url(http://www.w3.org/2000/svg);

svg|text {

fill: blue;

}

## 选择器列表

### 相对选择器

相对选择器列表是一个逗号分隔的选择器列表，解析为相对选择器，

它们以一个显式或隐含的组合器开始。

在这个例子中，斜体样式将应用于任何紧跟着<p>或<ul class=“red”>的h2标题。

注意，伪元素和:has ()选择器在:has ()相对选择器列表中是无效的。

h2:has(+ p, + ul.red) {

font-style: italic;

color: red;

}

### 解决无效选择器列表

解决无效选择器列表问题的一种方法是使用:is ()或:where ()伪类，

它们接受一个宽容的选择器列表。

宽容选择器列表中的每个选择器都是单独解析的。

所以列表中的任何无效选择器都会被忽略，

而有效的选择器会被使用。

:is(h1, input:-moz-placeholder, h3) {

color:red;

}

# CSS伪类

## :blank

这是一个选择**空的用户输入元素，暂时没有支持**

textarea:blank {

border: 3px solid red;

}

## :autofill

autocomplete只有特定的属性值的时候才好使，例如name,username,email

用户一次输入过后，再输入时会出现下拉列表，

选择即可触发autofill

<form action="./active,any-link,autofill.html">

<input type="text" autocomplete="name" name="name" />

<input type="email" autocomplete="email" name="email" />

<input type="submit"/>

</form>

## :current

目前不支持，测试失败

## :defined

它表示任何已经定义的元素。这包括浏览器内置的任何标准元素，以及已经成功定义的自定义元素（即使用 CustomElementRegistry.define () 方法）。

## :dir

根据元素中包含的文本的方向性匹配元素

<style>

:dir(ltr) {

background-color: yellow;

}

:dir(rtl) {

background-color: powderblue;

}

</style>

</head>

<body>

<div dir="rtl">

<span>test1</span>

<div dir="ltr">test2

<div dir="auto">עִבְרִית</div>

</div>

</div>

## :disabled

该伪类表示任何被禁用的元素。被禁用的元素是指不能被激活（选择，点击，输入等）或接受焦点的元素。该元素还有一个启用状态，可以被激活或接受焦点。

\*:disabled {

background-color: dimgrey;

color: linen;

opacity: 1;

}

## :empty

empty CSS 伪类表示没有子元素的任何元素。子元素可以是元素节点或文本（包括空白）。注释、处理指令和 CSS 内容不影响元素是否被认为是空的。

## :enable

这个伪类表示任何不处于禁用状态的元素。

## :first

用于与 @page 规则一起使用，表示打印文档的第一页。

@page :first {

margin: 20em 10em;

}

p {

page-break-after: always;

}

## :first-child

这个伪类表示一组兄弟元素中的第一个元素

## :first-of-type

这个伪类表示一组兄弟元素中类型相同的第一个元素。

## :focus

这个 CSS 伪类表示一个元素（如表单输入）获得了焦点。它通常在用户点击或触摸一个元素或用键盘的 Tab 键选择它时触发。

## :focus-visible

:focus 伪类表示当前获得焦点的元素，

而 :focus-visible 伪类表示当前获得焦点的元素，

但只有在用户需要知道焦点位置时才匹配。

如下案例，如果您依次单击每个元素，

您会发现当使用 :focus 伪类来设置焦点环时，

当用户单击按钮时，UA(用户代理) 会绘制焦点环。

但是，当使用 :focus-visible 来设置焦点环时，

UA 不会在用户单击按钮时绘制焦点环，就像默认情况下一样。

然后，如果您通过键盘遍历每个元素，

您会发现在所有三种情况下（默认、:focus 和 :focus-visible），

当用户使用键盘导航到按钮时，UA 会在按钮周围绘制焦点环。

这说明了 :focus-visible 如何使设计师能够遵循浏览器的逻辑，确定何时应该显示焦点环。

就是比考虑更多的用户操作，能比focus更人性化

## :focus-within

这个 CSS 伪类匹配一个获得了焦点的元素或其任何后代元素获得了焦点的元素

## :future

这是一个时间维度的伪类选择器，用于匹配在与 :current 匹配的元素之后完全出现的任何元素。例如，在使用 WebVTT 显示字幕的视频中。目前测试失败

## :has()

当has括号中的值满足条件的情况下匹配，这个伪类通过把可容错相对选择器列表作为参数，提供了一种针对引用元素选择父元素或者先前的兄弟元素的方法。

### has使用逻辑

* **:has () 伪类的逻辑**：:has () 伪类表示一个元素，如果作为参数传递的任何相对选择器与以该元素为锚点的至少一个元素匹配，则该元素被选中。这个伪类提供了一种方法，通过使用相对选择器列表作为参数，来选择一个元素的父元素或前一个兄弟元素。
* **:has () 伪类的特异性**：:has () 伪类的特异性与其参数中最具特异性的选择器相同，与 :is () 和 :not () 的行为一致。
* **:has () 伪类的嵌套限制**：:has () 伪类不能嵌套在另一个 :has () 中。这是因为许多伪元素是根据其祖先的样式有条件地存在的，允许这些伪元素被 :has () 查询会引入循环查询。
* **:has () 伪类的伪元素限制**：伪元素也不是 :has () 中的有效选择器，伪元素也不是 :has () 的有效锚点。
* **:has () 伪类的逗号分隔值和链式操作**：使用逗号分隔值在 :has () 关系选择器中，你可以检查是否有一个或多个特征是真的。x:has (a, b) 会选择 x，如果后代 a 或 b 存在。通过将多个 :has () 关系选择器链接在一起，你可以检查是否所有的特征都是真的。x:has (a):has (b) 会选择 x，如果后代 a 和 b 都存在。
* **:has () 伪类与正则表达式的类比**：我们可以将一些 CSS :has () 构造与正则表达式中的前瞻断言进行类比，因为它们都允许你根据一个条件选择元素（或字符串），而不实际选择与该条件匹配的元素（或字符串）本身。例如，正则表达式 abc (?=xyz) 只匹配紧跟 xyz 的 abc 字符串。CSS 中的类似构造是 .abc:has (+ .xyz)：它只选择有一个下一个兄弟元素是 .xyz 的 .abc 元素。部分 :has (+ .xyz) 作为一个前瞻操作，因为选择的是元素 .abc 而不是元素 .xyz。同样，对于否定前瞻的情况，正则表达式 abc (?!xyz) 只匹配不跟 xyz 的 abc 字符串。CSS 中的类似构造 .abc:has (+ :not (.xyz)) 不选择下一个元素是 .xyz 的 .abc 元素。

### 示例代码

h1,  
h2,  
h3 {  
 margin: 0 0 1rem 0;  
}  
  
:is(h1, h2, h3):has(+ :is(h2, h3, h4)) {  
 margin: 0 0 0.25rem 0;  
 color: #aaa;  
}

## :XXX

## :XXX

## :XXX

## :XXX

## :XXX

## :XXX