

## ☰ CSS 参考手...

[CSS 参考手册](#)[🔍 CSS 选择器](#)[CSS 听觉参考手册](#)[CSS Web安全字体](#)[CSS 动画](#)[CSS 函数](#)[CSS 单位](#)[CSS 颜色](#)[CSS 合法颜色值](#)[CSS 颜色名称](#)[CSS 颜色十六进制值](#)[CSS 浏览器支持](#)[Sass 教程](#)

## CSS 属性

[align-content](#)[align-items](#)[align-self](#)[all](#)[animation](#)[animation-delay](#)[animation-direction](#)[animation-duration](#)[animation-fill-mode](#)[← CSS 参考手册](#)[CSS 听觉参考手册 →](#)

## ☰ 分类导航

[HTML / CSS](#)[JavaScript](#)[服务端](#)[数据库](#)[数据分析](#)[移动端](#)[XML 教程](#)[ASP.NET](#)[Web Service](#)[开发工具](#)[网站建设](#)[Advertisement](#)

## CSS 选择器

CSS（层叠样式表）提供了多种选择器，用于选择要应用样式的 HTML 元素。CSS 选择器用于选择你想要的元素的样式的模式。

下列表格中的 "CSS" 列表示在 CSS 版本的属性定义，CSS1、CSS2 和 CSS3 是层叠样式表（Cascading Style Sheets）的不同版本，每个版本引入了新的特性和功能，以改进网页样式的控制和设计。

### 常见的 CSS 选择器

下是一些常见的 CSS 选择器：

**元素选择器 (Element Selector)**：通过元素名称选择 HTML 元素。

如下代码，p 选择器将选择所有 <p> 元素：

```
p {
  color: blue;
}
```

**类选择器 (Class Selector)**：通过类别名称选择具有特定类别的 HTML 元素。

类选择器以 . 开头，后面跟着类别名称。

如下代码，.highlight 选择器将选择所有具有类别为 "highlight" 的元素。

```
.highlight {
  background-color: yellow;
}
```

**ID 选择器 (ID Selector)**：通过元素的唯一标识符 (ID) 选择 HTML 元素。

ID 选择器以 # 开头，后面跟着 ID 名称。

如下代码，#runoob 选择器将选择具有 ID 为 "runoob" 的元素。

```
#runoob {
  width: 200px;
}
```

[反馈/建议](#)

animation-iteration-count
animation-name
animation-play-state
animation-timing-function
appearance
backface-visibility
background
background-attachment
background-blend-mode
background-clip
background-color
background-image
background-origin
background-position
background-repeat
background-size
border
border-bottom
border-bottom-color
border-bottom-left-radius
border-bottom-right-radius
border-bottom-style

}

**属性选择器 (Attribute Selector)**：通过元素的属性选择 HTML 元素。属性选择器可以根据属性名和属性值进行选择。

如下代码，`input[type="text"]` 选择器将选择所有 `type` 属性为 `"text"` 的 `<input>` 元素。

```
input[type="text"] {
  border: 1px solid gray;
}
```

**后代选择器 (Descendant Selector)**：通过指定元素的后代关系选择 HTML 元素。

后代选择器使用空格分隔元素名称。

如下代码，`div p` 选择器将选择所有在 `<div>` 元素内的 `<p>` 元素。

```
div p {
  font-weight: bold;
}
```

更多选择器参考下列表格：

选择器	示例	示例说明	CSS
<code>.class</code>	<code>.intro</code>	选择所有 <code>class="intro"</code> 的元素	1
<code>#id</code>	<code>#firstname</code>	选择所有 <code>id="firstname"</code> 的元素	1
<code>*</code>	<code>*</code>	选择所有元素	2
<code>element</code>	<code>p</code>	选择所有 <code>&lt;p&gt;</code> 元素	1
<code>element,element</code>	<code>div,p</code>	选择所有 <code>&lt;div&gt;</code> 元素和 <code>&lt;p&gt;</code> 元素	1
<code>element.class</code>	<code>p.hometown</code>	选择所有 <code>class="hometown"</code> 的 <code>&lt;p&gt;</code> 元素	1
<code>element element</code>	<code>div p</code>	选择 <code>&lt;div&gt;</code> 元素内的所有 <code>&lt;p&gt;</code> 元素	1
<code>element&gt;element</code>	<code>div&gt;p</code>	选择所有父级是 <code>&lt;div&gt;</code> 元素的 <code>&lt;p&gt;</code> 元素	2
<code>element+element</code>	<code>div+p</code>	选择所有紧跟在 <code>&lt;div&gt;</code> 元素之后的第一个 <code>&lt;p&gt;</code> 元素	2
<code>[attribute]</code>	<code>[target]</code>	选择所有带有 <code>target</code> 属性元素	2



border-bottom-width	<code>[attribute=value]</code>	<code>[target=-blank]</code>	选择所有使用target="-blank"的元素	2
border-collapse	<code>[attribute~=value]</code>	<code>[title~=flower]</code>	选择标题属性包含单词"flower"的所有元素	2
border-color	<code>[attribute]=language]</code>	<code>[lang]=en]</code>	选择 lang 属性等于 en，或者以 en- 为开头的所有元素	2
border-image				
border-image-outset				
border-image-repeat	<code>:link</code>	<code>a:link</code>	选择所有未访问链接	1
border-image-slice	<code>:visited</code>	<code>a:visited</code>	选择所有访问过的链接	1
border-image-source	<code>:active</code>	<code>a:active</code>	选择活动链接	1
border-image-width	<code>:hover</code>	<code>a:hover</code>	选择鼠标在链接上面时	1
border-left	<code>:focus</code>	<code>input:focus</code>	选择具有焦点的输入元素	2
border-left-color	<code>:first-letter</code>	<code>p:first-letter</code>	选择每一个<p>元素的第一个字母	1
border-left-style	<code>:first-line</code>	<code>p:first-line</code>	选择每一个<p>元素的第一行	1
border-left-width	<code>:first-child</code>	<code>p:first-child</code>	指定只有当<p>元素是其父级的第一个子级的样式。	2
border-radius				
border-right	<code>:before</code>	<code>p:before</code>	在每个<p>元素之前插入内容	2
border-right-color	<code>:after</code>	<code>p:after</code>	在每个<p>元素之后插入内容	2
border-right-style	<code>:lang(language)</code>	<code>p:lang(it)</code>	选择一个lang属性的起始值="it"的所有<p>元素	2
border-right-width	<code>element1~element2</code>	<code>p~ul</code>	选择p元素之后的每一个ul元素	3
border-spacing	<code>[attribute^=value]</code>	<code>a[src^="https"]</code>	选择每一个src属性的值以"https"开头的元素	3
border-style	<code>[attribute\$=value]</code>	<code>a[src\$=".pdf"]</code>	选择每一个src属性的值以".pdf"结尾的元素	3
border-top	<code>[attribute*=value]</code>	<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
border-top-color				
border-top-left-radius	<code>:first-of-type</code>	<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
border-top-right-radius				
border-top-style				



border-top-width	:last-of-type	p:last-of-type	选择每个p元素是其父级的最后一个p元素	3
border-width				
bottom	:only-of-type	p:only-of-type	选择每个p元素是其父级的唯一p元素	3
box-align				
box-direction				
box-flex	:only-child	p:only-child	选择每个p元素是其父级的唯一子元素	3
box-flex-group				
box-lines	:nth-child( <i>n</i> )	p:nth-child(2)	选择每个p元素是其父级的第二个子元素	3
box-ordinal-group	:nth-last-child( <i>n</i> )	p:nth-last-child(2)	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
box-orient				
box-pack				
box-shadow	:nth-of-type( <i>n</i> )	p:nth-of-type(2)	选择每个p元素是其父级的第二个p元素	3
box-sizing				
caption-side	:nth-last-of-type( <i>n</i> )	p:nth-last-of-type(2)	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
clear				
clip				
color	:last-child	p:last-child	选择每个p元素是其父级的最后一个子级。	3
column-count				
column-fill	:root	:root	选择文档的根元素	3
column-gap	:empty	p:empty	选择每个没有任何子级的p元素（包括文本节点）	3
column-rule	:target	#news:target	选择当前活动的#news元素（包含该锚名称的点击的URL）	3
column-rule-color				
column-rule-style				
column-rule-width	:enabled	input:enabled	选择每一个已启用的输入元素	3
column-span	:disabled	input:disabled	选择每一个禁用的输入元素	3
column-width	:checked	input:checked	选择每个选中的输入元素	3
columns	:not( <i>selector</i> )	:not(p)	选择每个并非p元素的元素	3
content	::selection	::selection	匹配元素中被用户选中或处于高亮状态的部分	3
counter-increment				
counter-reset				



cursor
direction
display
empty-cells
filter
flex
flex-basis
flex-direction
flex-flow
flex-grow
flex-shrink
flex-wrap
float
font
@font-face
font-family
font-size
font-size-adjust
font-stretch
font-style
font-variant
font-weight
@import
grid-rows
hanging-punctuation
height
icon
justify-content
@keyframes
left
letter-spacing
line-height
list-style

:out-of-range	:out-of-range	匹配值在指定区间之外的input元素	3
:in-range	:in-range	匹配值在指定区间之内的input元素	3
:read-write	:read-write	用于匹配可读及可写的元素	3
:read-only	:read-only	用于匹配设置 "readonly" (只读) 属性的元素	3
:optional	:optional	用于匹配可选的输入元素	3
:required	:required	用于匹配设置了 "required" 属性的元素	3
:valid	:valid	用于匹配输入值为合法的元素	3
:invalid	:invalid	用于匹配输入值为非法的元素	3

← CSS 参考手册

CSS 听觉参考手册 →

1TB 海外流量仅需 9.9美元

额外免费获取500万次WAF和1000万次http/https请求

阿里云

打开



2 篇笔记



写笔记



### CSS1、CSS2和CSS3

CSS1、CSS2 和 CSS3 是层叠样式表 (Cascading Style Sheets) 的不同版本, 每个版本引入了新的特性和功能, 以改进网页样式的控制和设计。

5

- **CSS1 (1996年推出)** : CSS1是第一版的CSS规范, 它引入了一套基本的样式属性和选择器, 使得开发者可以通过样式表将样式应用于HTML文档。CSS1提供了对字体、颜色、背



list-style-image

list-style-  
position

list-style-type

margin

margin-bottom

margin-left

margin-right

margin-top

max-height

max-width

@media

min-height

min-width

mix-blend-mode

object-fit

object-position

nav-down

nav-index

nav-left

nav-right

nav-up

opacity

order

outline

outline-color

outline-offset

outline-style

outline-width

overflow

overflow-x

overflow-y

padding

padding-bottom

景、边框等基本样式的控制。它为网页布局和样式提供了一些基础，但功能相对有限。

- **CSS2 (1998年推出)**：CSS2是CSS的第二版规范，引入了更多的功能和选择器，增强了样式控制的能力。它引入了浮动、定位和层叠等特性，使得开发者能够更好地控制和布局网页元素。CSS2还引入了伪类和伪元素选择器，允许开发者根据元素的状态或位置选择应用样式。

- **CSS3 (2001年开始，持续发展)**：CSS3是CSS的下一代规范，由一系列模块组成，每个模块引入了新的特性和功能。CSS3引入了更多的选择器、盒模型属性、颜色和背景属性、文本效果、过渡和动画效果、多列布局等。它还引入了媒体查询 (Media Queries)，允许开发者根据设备的属性和特征应用不同的样式，从而实现响应式设计。CSS3的发展是渐进式的，新的模块和特性不断被添加和标准化。

需要注意的是，CSS3并不是一个完整的标准版本，而是一个包含多个模块的集合。各个模块的发展和支持程度可能有所不同。不过，通常人们将从CSS3引入的新功能和特性统称为CSS3，以区别于较早的CSS版本。开发者可以根据目标浏览器和需求选择使用CSS3的特性，并通过特性检测和前缀来确保在不同浏览器中的兼容性。

DFEwvs 10个月前 (06-30)



CSS 提供了多种选择器，用于选择要应用样式的 HTML 元素。

以下是一些常见的 CSS 选择器：

22

- **元素选择器 (Element Selector)**：通过元素名称选择 HTML 元素。例如，`p` 选择器将选择所有 `<p>` 元素。

- **类选择器 (Class Selector)**：通过类别名称选择具有特定类别的 HTML 元素。类选择器以 `.` 开头，后面跟着类别名称。例如，`.highlight` 选择器将选择所有具有类别为 "highlight" 的元素。

- **ID 选择器 (ID Selector)**：通过元素的唯一标识符 (ID) 选择 HTML 元素。ID 选择器以 `#` 开头，后面跟着 ID 名称。例如，`#logo` 选择器将选择具有 ID 为 "logo" 的元素。

- **属性选择器 (Attribute Selector)**：通过元素的属性选择 HTML 元素。属性选择器可以根据属性名和属性值进行选择。例如，`input[type="text"]` 选择器将选择所有 `type` 属性为 "text" 的 `<input>` 元素。



padding-left
padding-right
padding-top
page-break-after
page-break-before
page-break-inside
perspective
perspective-origin
position
punctuation-trim
quotes
resize
right
rotation
tab-size
table-layout
target
target-name
target-new
target-position
text-align
text-align-last
text-decoration
text-decoration-color
text-decoration-line
text-decoration-style
text-indent
text-justify

- **后代选择器 (Descendant Selector)** : 通过指定元素的后代关系选择 HTML 元素。后代选择器使用空格分隔元素名称。例如, `div p` 选择器将选择所有在 `<div>` 元素内的 `<p>` 元素。
- **子元素选择器 (Child Selector)** : 通过指定元素的直接子元素关系选择 HTML 元素。子元素选择器使用 `>` 符号分隔父元素和子元素。例如, `ul > li` 选择器将选择所有 `<ul>` 元素的直接子元素中的 `<li>` 元素。
- **相邻兄弟选择器 (Adjacent Sibling Selector)** : 通过指定元素之间的相邻兄弟关系选择 HTML 元素。相邻兄弟选择器使用 `+` 符号分隔两个相邻元素。例如, `h2 + p` 选择器将选择紧接在 `<h2>` 元素后的 `<p>` 元素。
- **通用选择器 (Universal Selector)** : 选择所有 HTML 元素。通用选择器使用 `*` 符号。例如, `*` 选择器将选择页面上的所有元素。

除了上述常见的选择器, 还有伪类选择器 (Pseudo-Class Selector) 和伪元素选择器 (Pseudo-Element Selector) , 用于选择元素的特定状态或位置。

这只是一小部分 CSS 选择器的示例, CSS 还提供了更多的选择器和组合方式, 使开发者可以更精确地选择和应用样式于 HTML 元素。开发者可以根据需要选择适当的选择器来实现所需的样式效果。

DFEwvs 10个月前 (06-30)



text-outline

text-overflow

text-shadow

text-transform

text-wrap

top

transform

transform-origin

transform-style

transition

transition-delay

transition-  
duration

transition-  
property

transition-  
timing-function

unicode-bidi

vertical-align

visibility

white-space

width

word-break

word-spacing

word-wrap

z-index

writing-mode

gird

gap

row-gap

grid-area

grid-auto-  
columns

grid-auto-rows





grid-auto-flow
grid-column
grid-column-start
grid-column-end
grid-column-gap
grid-gap
grid-row-gap
grid-row-end
grid-row-start
grid-template
grid-template-areas
grid-template-columns
grid-template-rows
grid-row
@charset
clip-path
pointer-events
backdrop-filte

在线实例

- [HTML 实例](#)
- [CSS 实例](#)
- [JavaScript 实例](#)
- [Ajax 实例](#)
- [jQuery 实例](#)
- [XML 实例](#)
- [Java 实例](#)

字符集&工具

- [HTML 字符集设置](#)
- [HTML ASCII 字符集](#)
- [JS 混淆/加密](#)
- [PNG/JPEG 图片压缩](#)
- [HTML 拾色器](#)
- [JSON 格式化工具](#)
- [随机数生成器](#)

最新更新

- [SVG VScode](#)
- [HTML AI 编程助手](#)
- [C++ vector 容器](#)
- [免费开源, 无需 ...](#)
- [C VScode](#)
- [JavaScript VScode](#)
- [Java AI 编程助手](#)

站点信息

- [意见反馈](#)
- [免责声明](#)
- [关于我们](#)
- [文章归档](#)

关注微信



 [Privacy and cookie settings](#)

Managed by Google. Complies with IAB TCF. CMP ID: 300

