# **CSS选择器**

CSS选择器是用来将HTML的元素拣选出来，然后套用对应的CSS样式。例如你想改变网页内超链接文字的颜色，调整不同段落的文字大小，又或者为图片加上圆角效果等等，可以透过内嵌CSS的写法将样式直接套用到相关的HTML元素里面。

栗子：

<img src=”某个图片” style=”border-radius:10px; width：20px; height:20px”>

用img加一张图片，然后设定style属性，里面直接编写CSS样式,加个border-radius为图片加上圆角。

但这样过于麻烦，所以我们一般会将CSS与html分开来写，将CSS统一写在一个档案当中，然后再透过一种途径将这个CSS样式套用到对应的HTML元素上，这个途径就是今天要讲的CSS选择器。

添加CSS的方法：

1. 是在HTML中<head>里面添加<style>标签。
2. 在<head>中添加<link>标签，新建一个.css文件；

<link rel=”stylesheet” type=”text/css” href=”文件名.css” />

**CSS语法：**

选择器{

属性名:属性值

}

举十种CSS选择器栗子：

1.ID选取器

用#号开头然后自定一个名称，命名规则是：第一个字元一定要是英文字母（大小写皆可），第二个字元开始就可以是英文、数字、横线、底线等。

栗子:

#the-id-selector{

font-size:24px;

background-color:yellow;

padding:8px;

}

即是选取拥有id这个属性。并且它的值是the-id-selector的HTML元素,ID选取器有一个特点，就是它在整个HTML页面内必须是唯一的，一个ID只能用在一个HTML元素上，而一个HTML元素也只能有一个id。

同时我们在HTML里加一个<div>，将它的id设定为the-id-selector,这样他们就关联在一起了！

<div id=”the-id-selector”>

Coding Start

</div>

1. Class选取器

这是最常用的选取器，使用.号开头，然后自定义一个名称，命名规则与ID选取器相同，第一个字元一定要是英文字母（大小写皆可），第二个字元开始就可以是英文、数字、横线、底线等。

给个栗子:

.my-class{

font-family: Helvetica;

font-size: 24px;

color: yellow;

}

一个class可以套用到多个HTML元素上，一个HTML也可以有很多个class。

栗子：

加一个叫bg-blue的class

.bg-blue{

background-color:blue;

}

此时在HTML中我们只需要将两个样式空格隔开即可同时运用

<div class=”my-class bg-blue”>

Coding Startup

</div>

1. Tag选取器

在CSS中直接使用HTML标签（Tag）的名称，例如<h1><p><div>等等，会直接对应到HTML所有对应的标签中。

如果在Tag名称后加上ID选取器或Class选取器

栗子：

div.container{

background-color:yellow;

}

即表示container这个class只能够套用到div上，如果我将它套用到其他HTML标签中则不会生效。

1. 空格

直接上栗子：

.container div{

background-color: yellow;

}

既是选取拥有container这个class的HTML元素里所有div，注意并不包括container本身。

1. 大于符号

栗子：

.container > div {

background-color: yellow;

}

即是选取拥有container这个class的HTML元素里面的第一层的所有div，我们可以透过画面中颜色的变化去理解这个层级关系。

1. 加号

栗子：

.container + div {

background-color: yellow

}

即是选取拥有coontainer这个class的HTML元素与它身处同一层紧接着的div，如果紧接着的不是div，那此选择器就不会选取到任何东西。

1. 波浪符号

栗子：

.container ~ div {

background-color: yellow

}

即是选取拥有container这个class的HTML元素与它身处同一层之后的所有div，尽管其中有其它HTML元素隔开，只要是与container处于同一层都会套用得到。

1. 星号

\*是任何HTML标签的意思，如果只用一个星号的话即是选取所有HTML元素，我们一般会配合刚才提到的空格、加号或者波浪符号去做进一步的筛选，缩小选区范围。

栗子：

会将background-color红色套用到container之后的div与p标签

.container ~ \*{

background-color: red!important;

}

1. 属性选择器

是透过HTML元素的属性去选取。

栗子：

HTML中有三两个链接，分别连去Apple、Dlmu网站。

<a href=”https://www.apple.com” title=”Apple”>Apple</a>

<a href=”https://www.dlmu.edu.cn” title=”Dlmu”>Dlmu</a>

透过a[title]即可选取带有title属性的<a>Tag:

a[title]{

color:red

};

除此之外，我们还可以根据属性里面的值去做配对

栗子：

a[href=”https://www.apple.com”]

{

color:red;

}

即可选取到Apple的超链接，它还支援类似搜寻的功能；

* 运用^=符号可以搜寻到以什么开头的值a[href^=”https”]
* 运用$=符号可以搜寻到以什么结尾的值a[href$=”edu.cn”]
* \*=符号就当作关键字去搜寻a[href\*=”apple”]

1. Pseudo Classes（伪类）

用于选取普通选取器不能表达的讯息。比如当鼠标移到一个HTML元素上的时候更改一个浏览过的超链接的颜色，又比如在十个<div>中选取第13579个或选取第6个。这些都需要在普通的选取器上加以修饰才能表达得到的情况。

写法是在选取器后面加一个冒号，要设定超链接在浏览过后的颜色时可以用a:visited去定义。

a:visited{

color:red;

}

设定当鼠标移到HTML元素上的时候的样式可以用：hover去定义；

a:hover{

color:yellow;

}

而在一组多个HTML元素中选取特定的某几个<div>可以使用nth-child()

栗子：

container div:nth-child(2){

color: red;

}

即是选取第二个<div>

若括号里是even，即是选取双数；

若括号里是3n+0.即是选取3的倍数，以此类推。

附件一：

## CSS 选择器

|  |  |  |
| --- | --- | --- |
| **选择器** | **例子** | **例子描述** |
| [.*class*](https://www.w3school.com.cn/cssref/selector_class.asp) | .intro | 选择 class="intro" 的所有元素。 |
| .*class1*.*class2* | .name1.name2 | 选择 class 属性中同时有 name1 和 name2 的所有元素。 |
| .*class1* .*class2* | .name1 .name2 | 选择作为类名 name1 元素后代的所有类名 name2 元素。 |
| [#*id*](https://www.w3school.com.cn/cssref/selector_id.asp) | #firstname | 选择 id="firstname" 的元素。 |
| [\*](https://www.w3school.com.cn/cssref/selector_all.asp) | \* | 选择所有元素。 |
| [*element*](https://www.w3school.com.cn/cssref/selector_element.asp) | p | 选择所有 <p> 元素。 |
| [*element*.*class*](https://www.w3school.com.cn/cssref/selector_element_class.asp) | p.intro | 选择 class="intro" 的所有 <p> 元素。 |
| [*element*,*element*](https://www.w3school.com.cn/cssref/selector_element_comma.asp) | div, p | 选择所有 <div> 元素和所有 <p> 元素。 |
| [*element* *element*](https://www.w3school.com.cn/cssref/selector_element_element.asp) | div p | 选择 <div> 元素内的所有 <p> 元素。 |
| [*element*>*element*](https://www.w3school.com.cn/cssref/selector_element_gt.asp) | div > p | 选择父元素是 <div> 的所有 <p> 元素。 |
| [*element*+*element*](https://www.w3school.com.cn/cssref/selector_element_plus.asp) | div + p | 选择紧跟 <div> 元素的首个 <p> 元素。 |
| [*element1*~*element2*](https://www.w3school.com.cn/cssref/selector_gen_sibling.asp) | p ~ ul | 选择前面有 <p> 元素的每个 <ul> 元素。 |
| [[*attribute*]](https://www.w3school.com.cn/cssref/selector_attribute.asp) | [target] | 选择带有 target 属性的所有元素。 |
| [[*attribute*=*value*]](https://www.w3school.com.cn/cssref/selector_attribute_value.asp) | [target=\_blank] | 选择带有 target="\_blank" 属性的所有元素。 |
| [[*attribute*~=*value*]](https://www.w3school.com.cn/cssref/selector_attribute_value_contain.asp) | [title~=flower] | 选择 title 属性包含单词 "flower" 的所有元素。 |
| [[*attribute*|=*value*]](https://www.w3school.com.cn/cssref/selector_attribute_value_start.asp) | [lang|=en] | 选择 lang 属性值以 "en" 开头的所有元素。 |
| [[*attribute*^=*value*]](https://www.w3school.com.cn/cssref/selector_attr_begin.asp) | a[href^="https"] | 选择其 src 属性值以 "https" 开头的每个 <a> 元素。 |
| [[*attribute*$=*value*]](https://www.w3school.com.cn/cssref/selector_attr_end.asp) | a[href$=".pdf"] | 选择其 src 属性以 ".pdf" 结尾的所有 <a> 元素。 |
| [[*attribute*\*=*value*]](https://www.w3school.com.cn/cssref/selector_attr_contain.asp) | a[href\*="w3schools"] | 选择其 href 属性值中包含 "abc" 子串的每个 <a> 元素。 |
| [:active](https://www.w3school.com.cn/cssref/selector_active.asp) | a:active | 选择活动链接。 |
| [::after](https://www.w3school.com.cn/cssref/selector_after.asp) | p::after | 在每个 <p> 的内容之后插入内容。 |
| [::before](https://www.w3school.com.cn/cssref/selector_before.asp) | p::before | 在每个 <p> 的内容之前插入内容。 |
| [:checked](https://www.w3school.com.cn/cssref/selector_checked.asp) | input:checked | 选择每个被选中的 <input> 元素。 |
| [:default](https://www.w3school.com.cn/cssref/selector_default.asp) | input:default | 选择默认的 <input> 元素。 |
| [:disabled](https://www.w3school.com.cn/cssref/selector_disabled.asp) | input:disabled | 选择每个被禁用的 <input> 元素。 |
| [:empty](https://www.w3school.com.cn/cssref/selector_empty.asp) | p:empty | 选择没有子元素的每个 <p> 元素（包括文本节点）。 |
| [:enabled](https://www.w3school.com.cn/cssref/selector_enabled.asp) | input:enabled | 选择每个启用的 <input> 元素。 |
| [:first-child](https://www.w3school.com.cn/cssref/selector_first-child.asp) | p:first-child | 选择属于父元素的第一个子元素的每个 <p> 元素。 |
| [::first-letter](https://www.w3school.com.cn/cssref/selector_first-letter.asp) | p::first-letter | 选择每个 <p> 元素的首字母。 |
| [::first-line](https://www.w3school.com.cn/cssref/selector_first-line.asp) | p::first-line | 选择每个 <p> 元素的首行。 |
| [:first-of-type](https://www.w3school.com.cn/cssref/selector_first-of-type.asp) | p:first-of-type | 选择属于其父元素的首个 <p> 元素的每个 <p> 元素。 |
| [:focus](https://www.w3school.com.cn/cssref/selector_focus.asp) | input:focus | 选择获得焦点的 input 元素。 |
| [:fullscreen](https://www.w3school.com.cn/cssref/selector_fullscreen.asp) | :fullscreen | 选择处于全屏模式的元素。 |
| [:hover](https://www.w3school.com.cn/cssref/selector_hover.asp) | a:hover | 选择鼠标指针位于其上的链接。 |
| [:in-range](https://www.w3school.com.cn/cssref/selector_in-range.asp) | input:in-range | 选择其值在指定范围内的 input 元素。 |
| [:indeterminate](https://www.w3school.com.cn/cssref/selector_indeterminate.asp) | input:indeterminate | 选择处于不确定状态的 input 元素。 |
| [:invalid](https://www.w3school.com.cn/cssref/selector_invalid.asp) | input:invalid | 选择具有无效值的所有 input 元素。 |
| [:lang(*language*)](https://www.w3school.com.cn/cssref/selector_lang.asp) | p:lang(it) | 选择 lang 属性等于 "it"（意大利）的每个 <p> 元素。 |
| [:last-child](https://www.w3school.com.cn/cssref/selector_last-child.asp) | p:last-child | 选择属于其父元素最后一个子元素每个 <p> 元素。 |
| [:last-of-type](https://www.w3school.com.cn/cssref/selector_last-of-type.asp) | p:last-of-type | 选择属于其父元素的最后 <p> 元素的每个 <p> 元素。 |
| [:link](https://www.w3school.com.cn/cssref/selector_link.asp) | a:link | 选择所有未访问过的链接。 |
| [:not(*selector*)](https://www.w3school.com.cn/cssref/selector_not.asp) | :not(p) | 选择非 <p> 元素的每个元素。 |
| [:nth-child(*n*)](https://www.w3school.com.cn/cssref/selector_nth-child.asp) | 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。 |
| [:nth-last-child(*n*)](https://www.w3school.com.cn/cssref/selector_nth-last-child.asp) | p:nth-last-child(2) | 同上，从最后一个子元素开始计数。 |
| [:nth-of-type(*n*)](https://www.w3school.com.cn/cssref/selector_nth-of-type.asp) | 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。 |
| [:nth-last-of-type(*n*)](https://www.w3school.com.cn/cssref/selector_nth-last-of-type.asp) | p:nth-last-of-type(2) | 同上，但是从最后一个子元素开始计数。 |
| [:only-of-type](https://www.w3school.com.cn/cssref/selector_only-of-type.asp) | p:only-of-type | 选择属于其父元素唯一的 <p> 元素的每个 <p> 元素。 |
| [:only-child](https://www.w3school.com.cn/cssref/selector_only-child.asp) | p:only-child | 选择属于其父元素的唯一子元素的每个 <p> 元素。 |
| [:optional](https://www.w3school.com.cn/cssref/selector_optional.asp) | input:optional | 选择不带 "required" 属性的 input 元素。 |
| [:out-of-range](https://www.w3school.com.cn/cssref/selector_out-of-range.asp) | input:out-of-range | 选择值超出指定范围的 input 元素。 |
| [::placeholder](https://www.w3school.com.cn/cssref/selector_placeholder.asp) | input::placeholder | 选择已规定 "placeholder" 属性的 input 元素。 |
| [:read-only](https://www.w3school.com.cn/cssref/selector_read-only.asp) | input:read-only | 选择已规定 "readonly" 属性的 input 元素。 |
| [:read-write](https://www.w3school.com.cn/cssref/selector_read-write.asp) | input:read-write | 选择未规定 "readonly" 属性的 input 元素。 |
| [:required](https://www.w3school.com.cn/cssref/selector_required.asp) | input:required | 选择已规定 "required" 属性的 input 元素。 |
| [:root](https://www.w3school.com.cn/cssref/selector_root.asp) | :root | 选择文档的根元素。 |
| [::selection](https://www.w3school.com.cn/cssref/selector_selection.asp) | ::selection | 选择用户已选取的元素部分。 |
| [:target](https://www.w3school.com.cn/cssref/selector_target.asp) | #news:target | 选择当前活动的 #news 元素。 |
| [:valid](https://www.w3school.com.cn/cssref/selector_valid.asp) | input:valid | 选择带有有效值的所有 input 元素。 |
| [:visited](https://www.w3school.com.cn/cssref/selector_visited.asp) | a:visited |  |

附件二：

**CSS Selector**

|  |  |  |
| --- | --- | --- |
| # | id="" |  |
| .class | class="" |  |
| .class1.class2 | class="class1 class2" | 选择class1和class2同时具备的元素（无空格） |
| .class1 .class2 | class="class1"  class="class2" | class1下的class2 |
| tag{} | <tag></tag> |  |
| 空格 | 后代选择器：标签下的子标签  例如 .a b指a下的b | html 写法 <tag1 class="container" id="a"><tag id="b"><tag id="c"></tag></tag><tag id="d"></tag></tag1>  css 写法 .container tag{} |
| p.marked{} | 为所有class="marked"的p指定样式 |  |
| , | 和 | div , span 选中<div>和<span> |
| > | A > B  选中所有父级是A的B | html 写法  <tag class="container"id="a">  <tag id="b">  <tag id="c"></tag>  </tag>  <tag id="d"></tag>  </tag>  css 写法 .container > tag |
| + | A + B  紧跟着A的B | html 写法  <tag class="class-name" id="a">  <tag id="b"></tag>  <tag id="c"></tag>  </tag>  css 写法 tag + .b |
| ~ | 某一层同一层之后的所有同种元素 | html 写法  <tag1 class="container" id="a">  <tag id="b"></tag>  </tag>  <tag id="c"></tag>  <tag id="d"></tag>  css 写法 .container ~ tag{}  作用：修饰带有class属性值为container的容器之后的同一层的 tag 标签元素 |
| \* | 所有元素  （一般会配合其他符号组合使用） | html 写法 <tag1 class="container" id="a"><tag id="b"></tag></tag><tag id="c"></tag><tag id="d"></tag>  css 写法 .container \*{}  作用：修饰class值为container的所有子元素，不包括元素本身。 |

**属性选择器**

|  |  |
| --- | --- |
| 元素[属性] | 选择具有指定属性的指定元素，如 a[title]{} 选取具有title属性的<a> |
| 正则表达式 |  |
| ^="a" | 选择以a开头的内容 |
| $="b" | 选择以b结尾的内容 |
| \*="c" | 选择含有c的内容 |

**伪类(Pesudo Class)**

|  |  |
| --- | --- |
| :visited | 访问过的超链接 |
| :hover | 鼠标悬停 |
| :nth-child(<表达式>){} | 括号内表达式:  选择第几个/even双数/odd单数..?/3n+0 |