目录

[选择器 1](#_Toc532237359)

[Id选择器 1](#_Toc532237360)

[类选择器 1](#_Toc532237361)

[元素选择器 2](#_Toc532237362)

[全选择器（\*选择器） 3](#_Toc532237363)

[层级选择器 3](#_Toc532237364)

# 选择器

## Id选择器

<script type="text/javascript">

//通过原生方法处理

var div = document.getElementById('aaron');

div.style.border = "3px solid blue";

</script>

<script type="text/javascript">

$("#imooc2").css("border", "3px solid red");

</script>

## 类选择器

<script type="text/javascript">

//通过原生方法处理

//样式是可以多选的，所以得到的是一个合集

//需要通过循环给合集中每一个元素修改样式

var divs = document.getElementsByClassName('aaron');

for (var i = 0; i < divs.length; i++) {

divs[i].style.border = "3px solid blue";

}

</script>

<script type="text/javascript">

//通过jQuery直接传入class

//class选择器可以选择多个元素

$(".imooc").css("border", "3px solid red");

</script>

## 元素选择器

<script type="text/javascript">

//通过原生方法处理

//获取到所有的节点标记名为div的元素

var divs = document.getElementsByTagName('div');

for (var i = 0; i < divs.length; i++) {

divs[i].style.border = "3px solid blue";

}

</script>

<script type="text/javascript">

$("p").css("border", "3px solid red");

</script>

## 全选择器（\*选择器）

<script type="text/javascript">

//获取页面中所有的元素

var elements1 = document.getElementsByTagName('\*');

</script>

<script type="text/javascript">

//获取页面中所有的元素

var elements2 = $("\*") ;

//原生与jQuery方法比较

//===表示数据和类型都相等

if(elements2.length === elements1.length){

elements2.css("border","1px solid red");

}

</script>

## 层级选择器

<script type="text/javascript">

//子选择器

//$('div > p') 选择所有div元素里面的子元素P

$('div > p').css("border", "1px groove red");

</script>

//子代选择器只能用于下一代，而后代选择器可以用于所有代

<script type="text/javascript">

//后代选择器

//$('div p') 选择所有div元素里面的p元素

$('div p').css("border", "1px groove red");

</script>

<script type="text/javascript">

//相邻兄弟选择器

//选取prev后面的第一个的div兄弟节点

$(".prev + div").css("border", "3px groove blue");

</script>

<script type="text/javascript">

//一般相邻选择器

//选取prev后面的所有的div兄弟节点

$(".prev ~ div").css("border", "3px groove blue");

</script>

## 基本筛选选择器

<script type="text/javascript">

//找到第一个div

$(".div:first").css("color", "#CD00CD");

</script>

<script type="text/javascript">

//找到最后一个div

$(".div:last").css("color", "#CD00CD");

</script>

<script type="text/javascript">

//:even 选择所引值为偶数的元素，从 0 开始计数

$(".div:even").css("border", "3px groove red");

</script>

<script type="text/javascript">

//:odd 选择所引值为奇数的元素，从 0 开始计数

$(".div:odd").css("border", "3px groove blue");

</script>

<script type="text/javascript">

//:eq

//选择单个

$(".aaron:eq(2)").css("border", "3px groove blue");

</script>

<script type="text/javascript">

//:gt 选择匹配集合中所有索引值大于给定index参数的元素

$(".aaron:gt(3)").css("border", "3px groove blue");

</script>

<script type="text/javascript">

//:lt 选择匹配集合中所有索引值小于给定index参数的元素

//与:gt相反

$(".aaron:lt(2)").css("color", "#CD00CD");

</script>

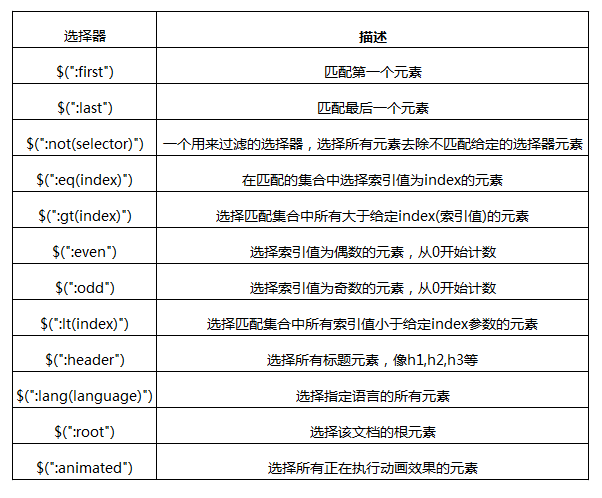
<script type="text/javascript">

//:not 选择所有元素去除不匹配给定的选择器的元素

//选中所有紧接着没有checked属性的input元素后的p元素，赋予颜色

$("input:not(:checked) + p").css("background-color", "#CD00CD");

</script>



## 内容筛选器

<script type="text/javascript">

//查找所有class='div'中DOM元素中包含"contains"的元素节点

$(".div:contains(':contains')").css("color", "#CD00CD");

</script>

<script type="text/javascript">

//查找所有class='div'中DOM元素中包含"span"的元素节点

$(".div:has(span)").css("color", "blue");

</script>

<script type="text/javascript">

//选择所有包含子元素或者文本的a元素

$("a:parent").css("border", "3px groove blue");

</script>

<script type="text/javascript">

//找到a元素下面的所有空节点(没有子元素)

$("a:empty").text(":empty").css("border", "3px groove red");

</script>

