

选择器是jQuery最基础的东西，本文中列举的选择器基本上囊括了所有的jQuery选择器，也许各位通过这篇文章能够加深对jQuery选择器的理解，它们本身用法就非常简单，我更希望的是它能够提升个人编写jQuery代码的效率。本文配合截图、代码和简单的概括对所有jQuery选择器进行了介绍，也列举出了一些需要注意和区分的地方。

## 一、基本选择器

### 1. id选择器（指定id元素）

将id="one"的元素背景色设置为黑色。（id选择器返单个元素）

```
$(document).ready(function () {  
    $('#one').css('background', '#000');  
});
```

### 2. class选择器（遍历css类元素）

将class="cube"的元素背景色设为黑色

```
$(document).ready(function () {  
    $('.cube').css('background', '#000');  
});
```

### 3. element选择器（遍历html元素）

将p元素的文字大小设置为12px

```
$(document).ready(function () {  
    $('p').css('font-size', '12px');  
});
```

### 4. \* 选择器（遍历所有元素）

```
$(document).ready(function () {  
    // 遍历form下的所有元素，将字体颜色设置为红色  
    $('form *').css('color', '#FF0000');  
});
```

### 5. 并列选择器

```
$(document).ready(function () {  
    // 将p元素和div元素的margin设为0  
    $('p, div').css('margin', '0');  
});
```

## 二、层次选择器

### 1. parent > child（直系子元素）

```
$(document).ready(function () {
    // 选取div下的第一代span元素，将字体颜色设为红色
    $('div > span').css('color', '#FF0000');
});
```

下面的代码，只有第一个span会变色，第二个span不属于div的一代子元素，颜色保持不变。

```
<div>
    <span>123</span>
    <p>
        <span>456</span>
    </p>
</div>
```

## 2. prev + next（下一个兄弟元素，等同于next()方法）

```
$(document).ready(function () {
    // 选取class为item的下一个div兄弟元素
    $('.item + div').css('color', '#FF0000');
    // 等价代码
    //$('.item').next('div').css('color', '#FF0000');
});
```

下面的代码，只有123和789会变色

```
<p class="item"></p>
<div>123</div>
<div>456</div>
<span class="item"></span>
<div>789</div>
```

## 3. prev ~ siblings（prev元素的所有兄弟元素，等同于nextAll()方法）

```
$(document).ready(function () {
    // 选取class为inside之后的所有div兄弟元素
    $('.inside ~ div').css('color', '#FF0000');
    // 等价代码
    //$('.inside').nextAll('div').css('color', '#FF0000');
});
```

下面的代码，G2和G4会变色

```
<div class="inside">G1</div>
<div>G2</div>
<span>G3</span>
<div>G4</div>
```

# 三、过滤选择器

## 1. 基本过滤选择器

——1.1 :first和:last（取第一个元素或最后一个元素）

```
$(document).ready(function () {
    $('span:first').css('color', '#FF0000');
```

```
    $('span:last').css('color', '#FF0000');  
});
```

下面的代码，G1（first元素）和G3（last元素）会变色

```
<span>G1</span>  
<span>G2</span>  
<span>G3</span>
```

——1.2 :not（取非元素）

```
$(document).ready(function () {  
    $('div:not(.wrap)').css('color', '#FF0000');  
});
```

下面的代码，G1会变色

```
<div>G1</div>  
<div class="wrap">G2</div>
```

但是，请注意下面的代码：

```
<div>  
    G1  
    <div class="wrap">G2</div>  
</div>
```

当G1所在div和G2所在div是父子关系时，G1和G2都会变色。

——1.3 :even和:odd（取偶数索引或奇数索引元素，索引从0开始，even表示偶数，odd表示奇数）

```
$(document).ready(function () {  
    $('tr:even').css('background', '#EEE'); // 偶数行颜色  
    $('tr:odd').css('background', '#DADADA'); // 奇数行颜色  
});
```

A、C行颜色#EEE（第一行的索引为0），B、D行颜色#DADADA

A
B
C
D

```
<table width="200" cellpadding="0" cellspacing="0">  
    <tbody>  
        <tr><td>A</td></tr>  
        <tr><td>B</td></tr>  
        <tr><td>C</td></tr>  
        <tr><td>D</td></tr>  
    </tbody>  
</table>
```

——1.4 :eq(x)（取指定索引的元素）



```
$(document).ready(function () {  
    $('tr:eq(2)').css('background', '#FF0000');  
});
```

更改第三行的背景色，在上面的代码中C的背景会变色。

——1.5 :gt(x)和:lt(x)（取大于x索引或小于x索引的元素）

```
$(document).ready(function () {  
    $('ul li:gt(2)').css('color', '#FF0000');  
    $('ul li:lt(2)').css('color', '#0000FF');  
});
```

L4和L5会是红色，L1和L2会是蓝色，L3是默认颜色



```
<ul>  
    <li>L1</li>  
    <li>L2</li>  
    <li>L3</li>  
    <li>L4</li>  
    <li>L5</li>  
</ul>
```

——1.6 :header（取H1~H6标题元素）

```
$(document).ready(function () {  
    $(':header').css('background', '#EFEFEF');  
});
```

下面的代码，H1~H6的背景色都会变

# H1

## H2

### H3

#### H4

##### H5

###### H6

中文网  
PHP100.com

```
<h1>H1</h1>
<h2>H2</h2>
<h3>H3</h3>
<h4>H4</h4>
<h5>H5</h5>
<h6>H6</h6>
```

## 2. 内容过滤选择器

——2.1 :contains(text)（取包含text文本的元素）

```
$(document).ready(function () {
    // dd元素中包含"jQuery"文本的会变色
    $('dd:contains("jQuery")').css('color', '#FF0000');
});
```

下面的代码，第一个dd会变色

技术

jQuery, .NET, CLR

SEO

关键字排名

其他

中文网  
PHP100.com

```
<d1>
  <dt>技术</dt>
  <dd>jQuery, .NET, CLR</dd>
  <dt>SEO</dt>
  <dd>关键字排名</dd>
  <dt>其他</dt>
  <dd></dd>
</d1>
```

——2.2 :empty（取不包含子元素或文本为空的元素）

```
$(document).ready(function () {
```

```
$( 'dd:empty' ).html( '没有内容' );  
});
```

技术

jQuery, .NET, CLR

SEO

关键字排名

其他

没有内容

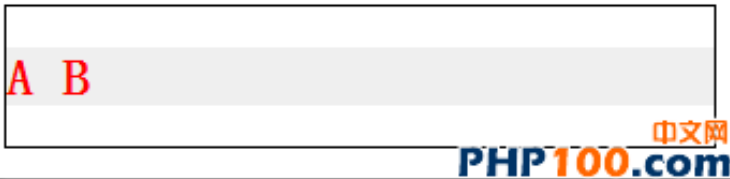
中文网  
PHP100.com

上面第三个dd会显示"没有内容"文本

### ——2.3 :has(selector)（取选择器匹配的元素）

```
$(document).ready(function () {  
    // 为包含span元素的div添加边框  
    $('div:has(span)').css('border', '1px solid #000');  
});
```

即使span不是div的直系子元素，也会生效

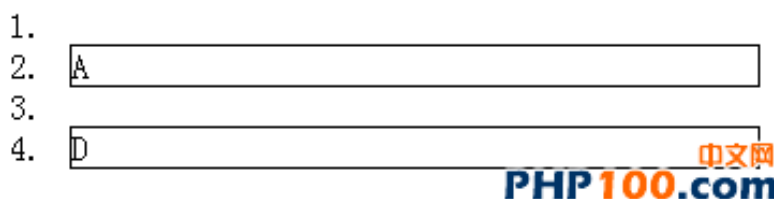


```
<div>  
    <h2>  
        A  
        <span>B</span>  
    </h2>  
</div>
```

### ——2.4 :parent（取包含子元素或文本的元素）

```
$(document).ready(function () {  
    $('ol li:parent').css('border', '1px solid #000');  
});
```

下面的代码，A和D所在的li会有边框



```
<ol>  
    <li></li>
```

```
<li>A</li>
<li></li>
<li>D</li>
</ol>
```

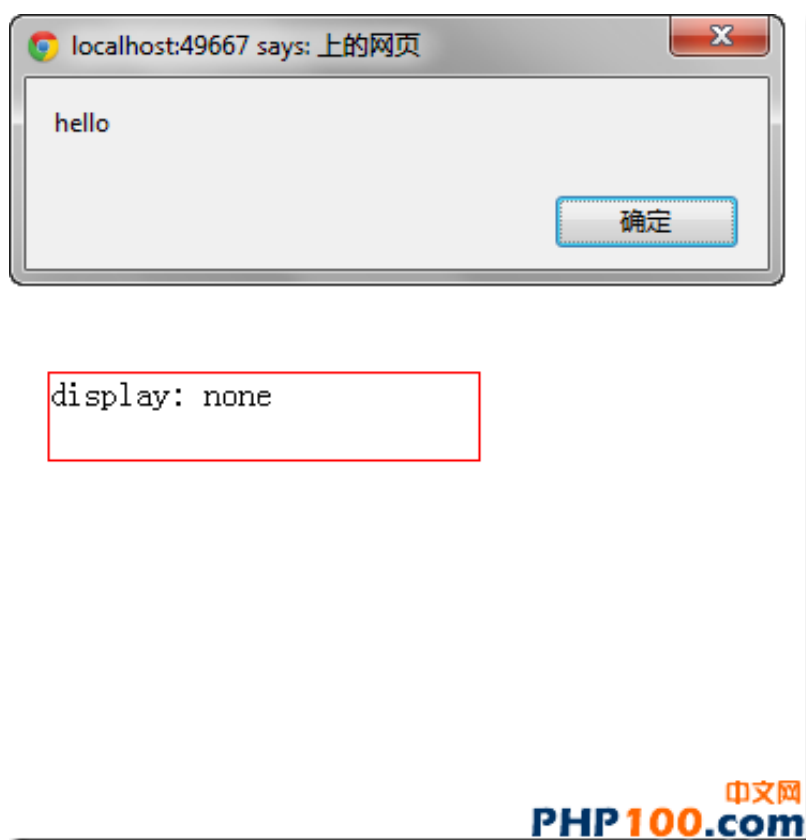
### 3. 可见性过滤选择器

——3.1 :hidden（取不可见的元素）

jQuery至1.3.2之后的:hidden选择器仅匹配display:none或<input type="hidden" />的元素，而不匹配visibility: hidden或opacity:0的元素。这也意味着hidden只匹配那些“隐藏的”并且不占空间的元素，像visibility:hidden或opacity:0的元素占据了空间，会被排除在外。

参照：<http://www.jquerysdk.com/api/hidden-selector>

下面的代码，先弹出"hello"对话框，然后hid-1会显示，hid-2仍然是不可见的。



```
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
  <title></title>
  <style type="text/css">
    div
    {
      margin: 10px;
      width: 200px;
      height: 40px;
      border: 1px solid #FF0000;
      display: block;
    }
    .hid-1
    {
      display: none;
    }
  </style>
</head>
<body>
```

```

        .hid-2
        {
            visibility: hidden;
        }
    </style>
    <script type="text/javascript" src="js/jquery.min.js"></script>
    <script type="text/javascript">
        $(document).ready(function() {
            $('div:hidden').show(500);
            alert($('input:hidden').val());
        });
    </script>
</head>
<body>
    <div class="hid-1">display: none</div>
    <div class="hid-2">visibility: hidden</div>
    <input type="hidden" value="hello"/>
</body>
</html>

```

### ——3.2 :visible（取可见的元素）

下面的代码，最后一个div会有背景色

jQuery选择器大全

中文网  
PHP100.com

```

<script type="text/javascript">
    $(document).ready(function() {
        $('div:visible').css('background', '#EEADBB');
    });
</script>
<div class="hid-1">display: none</div>
<div class="hid-2">visibility: hidden</div>
<input type="hidden" value="hello"/>
<div>
    jQuery选择器大全
</div>

```

## 4. 属性过滤选择器

### ——4.1 [attribute]（取拥有attribute属性的元素）

下面的代码，最后一个a标签没有title属性，所以它仍然会带下划线



- DOM对象和jQuery对象
- jQuery选择器大全
- jQuery事件大全
- 基于jQuery的插件开发
- Wordpress & jQuery
- 其他

```
<script type="text/javascript">
    $(document).ready(function() {
        $('a[title]').css('text-decoration', 'none');
    });
</script>
<ul>
    <li><a href="#" title="DOM对象和jQuery对象" class="item">DOM对象和jQuery对象</a></li>
    <li><a href="#" title="jQuery选择器大全" class="item-selected">jQuery选择器大全</a></li>
    <li><a href="#" title="jQuery事件大全" class="item">jQuery事件大全</a></li>
    <li><a href="#" title="基于jQuery的插件开发" class="item">基于jQuery的插件开发</a></li>
    <li><a href="#" title="Wordpress & jQuery" class="item">Wordpress & jQuery</a></li>
    <li><a href="#" class="item">其他</a></li>
</ul>
```

——4.2 **[attribute = value]**和**[attribute != value]**（取**attribute**属性值等于**value**或不等于**value**的元素）

分别为class="item"和class!=item的a标签指定文字颜色

- DOM对象和jQuery对象
- jQuery选择器大全
- jQuery事件大全
- 基于jQuery的插件开发
- Wordpress & jQuery
- 其他