

目录

一、基本选择器	4
1. id 选择器 (指定 id 元素)	4
2. class 选择器 (遍历 css 类元素)	4
3. element 选择器 (遍历 html 元素)	4
4. * 选择器 (遍历所有元素)	4
5. 并列选择器	4
二、层次选择器	4
1. parent > child (直系子元素)	4
2. prev + next (下一个兄弟元素, 等同于 next()方法)	5
3. prev ~ siblings (prev 元素的所有兄弟元素, 等同于 nextAll()方法)	5
三、过滤选择器	5
1. 基本过滤选择器	5
1.1 :first 和:last (取第一个元素或最后一个元素)	5
1.2 :not (取非元素)	6
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)	6
1.4 :eq(x) (取指定索引的元素)	6
1.5 :gt(x)和:lt(x) (取大于 x 索引或小于 x 索引的元素)	7
1.6 :header (取 H1~H6 标题元素)	7
2. 内容过滤选择器	7
2.1 :contains(text) (取包含 text 文本的元素)	7
2.2 :empty (取不包含子元素或文本为空的元素)	8

2.3 :has(selector) (取选择器匹配的元素)	8
2.4 :parent (取包含子元素或文本的元素)	9
3. 可见性过滤选择器	9
3.1 :hidden (取不可见的元素)	9
3.2 :visible (取可见的元素)	11
4. 属性过滤选择器	11
4.1 [attribute] (取拥有 attribute 属性的元素)	11
4.2 [attribute = value]和[attribute != value] (取 attribute 属性值等于 value 或不等于 value 的元素)	12
4.3 [attribute ^= value], [attribute \$= value]和[attribute *= value] (attribute 属性值以 value 开始, 以 value 结束, 或包含 value 值)	12
4.4 [selector1][selector2] (复合型属性过滤器, 同时满足多个条件)	13
5. 子元素过滤选择器	13
5.1 :first-child 和:last-child.....	13
5.2 :only-child	15
5.3 :nth-child	16
6. 表单对象属性过滤选择器	18
6.1 :enabled 和:disabled (取可用或不可用元素)	18
6.2 :checked (取选中的单选框或复选框元素)	19
6.3 :selected (取下拉列表被选中的元素)	20
四、表单选择器.....	20
1. :input (取 input,textarea,select,button 元素)	20
2. :text (取单行文本框元素) 和:password (取密码框元素)	20
3. :radio (取单选框元素)	21
4. :checkbox (取复选框元素)	22

5. :submit (取提交按钮元素)	22
6. :reset (取重置按钮元素)	22
7. :button (取按钮元素)	23
8. :file (取上传域元素)	23
9. :hidden (取不可见元素)	23

JQuery 选择器大全

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）

A
B
C
D

```
$(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

- L1
- L2
- L3
- L4
- L5

```
<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

```
<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	

```
<dl>
  <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>
</dl>
```

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

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

A B

```
<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

- 1.
2.
- 3.
4.

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

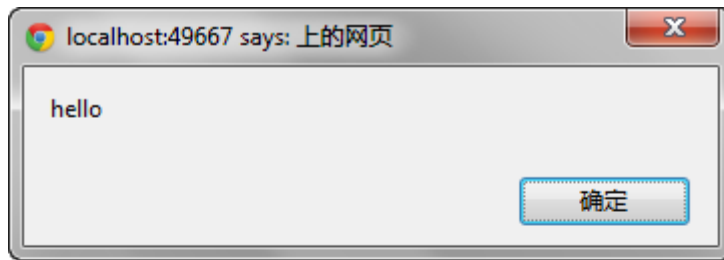
3. 可见性过滤选择器

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。



display: none

```
<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;
    }
    .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

```

<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 或不同于 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

```

<script type="text/javascript">
    $(document).ready(function() {
        $('a[class=item]').css('color', '#FF99CC');
        $('a[class!=item]').css('color', '#FF6600');
    });
</script>

```

4.3 [attribute ^= value], [attribute \$= value]和[attribute *= value] (attribute 属性值以 value 开始, 以 value 结束, 或包含 value 值)

在属性选择器中, ^\$符号和正则表达式的开始结束符号表示的含义是一致的, *模糊匹配, 类似于 sql 中的 like '%str%'。

- 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">

```

```
// 识别大小写，输入字符串时可以输入引号，[title^=jQuery]和
[title^="jQuery"]是一样的
$('a[title^=jQuery]').css('font-weight', 'bold');
$('a[title$=jQuery]').css('font-size', '24px');
$('a[title*=jQuery]').css('text-decoration', 'line-through');
</script>
```

4.4 [selector1][selector2] (复合型属性过滤器，同时满足多个条件)

将 title 以"jQuery"开始，并且 class="item"的 a 标签隐藏，那么

`jQuery 事件大全`会被隐藏

```
<script type="text/javascript">
    $(document).ready(function() {
        $('a[title^=jQuery][class=item]').hide();
    });
</script>
```

5. 子元素过滤选择器

5.1 :first-child 和:last-child

:first-child 表示第一个子元素，:last-child 表示最后一个子元素。

需要大家注意的是，:first 和:last 返回的都是单个元素，而:first-child 和:last-child 返回的都是集合元素。举个例子：div:first 返回的是整个 DOM 文档中第一个 div 元素，而 div:first-child 是返回所有 div 元素下的第一个元素合并后的集合。

这里有个问题：如果一个元素没有子元素，:first-child 和:last-child 会返回 null 吗？

请看下面的代码：

```
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title></title>
    <script type="text/javascript" src="js/jquery.min.js"></script>
    <script type="text/javascript">
        $(document).ready(function() {
            var len1 = $('div:first-child').length;
            var len2 = $('div:last-child').length;
        });
    </script>
</head>
<body>
<div>
    <div>
        <div></div>
    </div>
</div>
</body>
</html>
```

也许你觉得这个答案，是不是太简单了？len1 = 2, len2 = 2。但实际确并不是，它们俩都等于 3。

把上面的代码稍微修改一下：

```
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
  <title></title>
  <script type="text/javascript" src="js/jquery.min.js"></script>
  <script type="text/javascript">
    $(document).ready(function() {
      var len1 = $('div:first-child').length;
      var len2 = $('div:last-child').length;
      $('div:first-child').each(function() {
        alert($(this).html());
      });
    });
  </script>
</head>
<body>
<div>123
  <div>456
    <div></div>
  </div>
</div>
</body>
</html>
```

结果却是弹出三个 alert , 只不过最后一个 alert 里面是空白的。



5.2 :only-child

当某个元素有且仅有一个子元素时，:only-child 才会生效。

```
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
  <title></title>
  <script type="text/javascript" src="js/jquery.min.js"></script>
  <script type="text/javascript">
    $(document).ready(function() {
      $('div:only-child').css('border', '1px solid
#FF0000').css('width', '200px');
    });
  </script>
</head>
<body>
<div>123
  <div>456
    <div></div>
  </div>
</div>
</body>
</html>
```

这里:only-child 也是三个元素，从最后一个很粗的红色边框（实际是两个元素的边框重叠了）也可以看出来。

123
456

5.3 :nth-child

看到这个就想起英文单词里的，fourth, fifth, sixth.....，nth 表示第 n 个，:nth-child 就表示第 n 个 child 元素。要注意的是，这儿的 n 不像 eq(x)、gt(x)或 lt(x) 是从 0 开始的，它是从 1 开始的，英文里好像也没有 zeroth 这样的序号词吧。

:nth-child 有三种用法：

- 1) :nth-child(x)，获取第 x 个子元素
- 2) :nth-child(even)和:nth-child(odd)，从 1 开始，获取第偶数个元素或第奇数个元素
- 3) :nth-child(xn+y)， $x \geq 0$ ， $y \geq 0$ 。例如 $x = 3$, $y = 0$ 时就是 $3n$ ，表示取第 $3n$ 个元素 ($n \geq 0$)。实际上 $xn+y$ 是上面两种的通项式。（当 $x=0, y \geq 0$ 时，等同于:nth-child(y)；当 $x=2, y=0$ 时，等同于:nth-child(even)；当 $x=2, y=1$ 时，等同于:nth-child(odd)）

下面的两个例子是针对 2)和 3)的，1)的例子我就不列举了。

例 2：

- | |
|----------------|
| 1. NBA 2012季后赛 |
| 2. NBA 2011季后赛 |
| 3. NBA 2010季后赛 |
| 4. NBA 2009季后赛 |
| 5. NBA 2008季后赛 |
| 6. NBA 2007季后赛 |

```
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
  <title></title>
  <style type="text/css">

    td {
      width: 200px;
      height: 32px;
      line-height: 32px;
    }

  </style>
  <script type="text/javascript" src="js/jquery.min.js"></script>
  <script type="text/javascript">
    $(document).ready(function() {
      // 偶数行背景红色
      $('tr:nth-child(even)').css('background', '#FF0000');
      // 奇数行背景蓝色
      $('tr:nth-child(odd)').css('background', '#0000FF');
    });
  </script>
</head>
<body>
  <table>
    <tr><td>1. NBA 2012 季后赛</td></tr>
    <tr><td>2. NBA 2011 季后赛</td></tr>
    <tr><td>3. NBA 2010 季后赛</td></tr>
    <tr><td>4. NBA 2009 季后赛</td></tr>
    <tr><td>5. NBA 2008 季后赛</td></tr>
    <tr><td>6. NBA 2007 季后赛</td></tr>
  </table>
```

```
</body>
</html>
```

例 3 (html 代码和例 2 是一样的) :

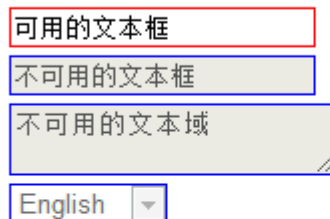
1. NBA 2012季后赛
2. NBA 2011季后赛
3. NBA 2010季后赛
4. NBA 2009季后赛
5. NBA 2008季后赛
6. NBA 2007季后赛

```
<script type="text/javascript">
$(document).ready(function() {
    $('tr:nth-child(3n)').css('background', '#0000FF');
});
</script>
```

6. 表单对象属性过滤选择器

6.1 :enabled 和:disabled (取可用或不可用元素)

:enabled 和:disabled 的匹配范围包括 input, select, textarea。



```
<script type="text/javascript">
$(document).ready(function() {
    $(':enabled').css('border', '1px solid #FF0000');
    $(':disabled').css('border', '1px solid #0000FF');
});
```

```

    });
</script>
<div>
    <input type="text" value="可用的文本框" />
</div>
<div>
    <input type="text" disabled="disabled" value="不可用的文本框" />
</div>
<div>
    <textarea disabled="disabled">不可用的文本域</textarea>
</div>
<div>
<select disabled="disabled">
    <option>English</option>
    <option>简体中文</option>
</select>
</div>

```

6.2 :checked (取选中的单选框或复选框元素)

下面的代码，更改边框或背景色仅在 IE 下有效果，chrome 和 firefox 不会改变，但是 alert 都会弹出来。



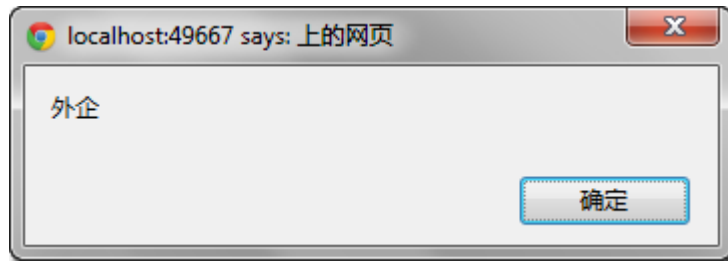
```

<script type="text/javascript">
    $(document).ready(function() {
        $(':checked').css('background', '#FF0000').each(function() {
            alert($(this).val());
        });
    });
</script>
<div>
    <input type="checkbox" checked="checked" value="must"/>必须勾选
</div>
<div>
    你现在工作的企业属于：

```

```
<input type="radio" name="radio" checked="checked" value="外企"/>外企  
<input type="radio" name="radio" value="国企"/>国企  
<input type="radio" name="radio" value="民企"/>民企  
</div>
```

6.3 :selected (取下拉列表被选中的元素)



```
<script type="text/javascript">  
    $(document).ready(function() {  
        alert($('':selected').val());  
    });  
</script>  
<select>  
    <option value="外企">外企</option>  
    <option value="国企">国企</option>  
    <option value="私企">私企</option>  
</select>
```

四、表单选择器

1. :input (取 input,textarea,select,button 元素)

:input 元素这里就不再多说了，前面的一些例子中也已经囊括了。

2. :text (取单行文本框元素) 和:password (取密码框元素)

这两个选择器分别和属性选择器\$('input[type=text]')、
\$('input[type=password]')等同。

```
<script type="text/javascript">  
$(document).ready(function() {  
    $(':text').css('border', '1px solid #FF0000');  
    $(':password').css('border', '1px solid #0000FF');  
  
    // 等效代码  
    //$('input[type=text]').css("border", '1px solid #FF0000');  
    //$('input[type=password]').css('border', '1px solid #0000FF');  
});  
</script>  
<fieldset style="width: 300px;">  
    <legend>账户登录</legend>  
    <div>  
        <label>用户名 : </label><input type="text"/>  
    </div>  
    <div>  
        <label>密&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&码 : </label><input type="password"/>  
    </div>  
</fieldset>
```

:radio 选择器和属性选择器\$('input[type=radio]')等同

```
$(document).ready(function() {
    $(':radio').each(function() {
        alert($(this).val());
    });
});
```

// 等效代码

```
/*
$('input[type=radio]').each(function() {
    alert($(this).val());
});
*/
```

```
});  
</script>
```

你现在工作的企业属于：

☒ 外企

☐ 国企

☐ 民企

4. :checkbox (取复选框元素)

:checkbox 选择器和属性选择器\$('input[type=checkbox]')等同

```
<script type="text/javascript">  
$(document).ready(function() {  
    $(':checkbox').each(function() {  
        alert($(this).val());  
    });  
    // 等效代码  
    /*  
    $('input[type=checkbox]').each(function() {  
        alert($(this).val());  
    });  
    */  
});  
</script>
```

您的兴趣爱好：

☐ 游泳

☐ 看书

☒ 打篮球

☒ 电脑游

戏

上面的代码，会将所有 checkbox 的 value 输出出来。若你想选择选中项，

有三种写法：

```
$(':checkbox:checked').each(function() {  
    alert($(this).val());  
});
```

```
$('input[type=checkbox][checked]').each(function() {  
    alert($(this).val());  
});
```

```
$(':checked').each(function() {  
    alert($(this).val());  
});
```

5. :submit (取提交按钮元素)

:submit 选择器和属性选择器\$('input[type=submit]')等同

6. :reset (取重置按钮元素)

:reset 选择器和属性选择器\$('input[type=reset]')等同

7. :button (取按钮元素)

:button 选择器和属性选择器\$('input[type=button]')等同

8. :file (取上传域元素)

:file 选择器和属性选择器\$('input[type=file]')等同

9. :hidden (取不可见元素)

:hidden 选择器和属性选择器\$('input[type=hidden]')等同