

# jQuery—

jQuery [2]

## 一下载jQuery

jQuery下载地址：<http://jquery.com/>

jQuery 是一个功能强大的 **JavaScript 库**。  
jQuery 极大地简化了 JavaScript 编程。

- 避免了浏览器的兼容问题。

## 一、jQuery()函数

jQuery 另一个名称 **\$**, jQuery() 等同于 **\$()**

### jQuery设计精髓

- 强大的 **函数式编程**
- **链式操作**
- **getter** **setter** 是合体的

**getter** 是获取中集合中的一个的值，**setter** 是设置集合中的每一个

## 二、**\$()** 的四大调用方式

### 1、**\$( function(){} )**

文档加载完毕执行函数，等同于 `DOMContentLoaded` ,旧的写法  
`$(document).ready(function(){})`

## 2、\$( `selector` ,start )

`selector` 是一个CSS选择器,  
`start` 可选 在该元素下或该子元素下面中去匹配

- `size()` / `length` 长度

jQuery 对象中元素的个数。  
这个函数的返回值与 jQuery 对象的`size()`属性一致。

- `css( )` 获取/设置样式

`css( key,val )` 参数形式

`css( {key: val} )` 对象形式 ,可以一次赋值多个属性,val可以是函数——很多值都可以用 `return` 函数

- 可以使用连字符: `background-color` 驼峰名属性: `backgroundColor`
- 获取计算后的css样式
- 不能获取复合样式

```
1. $('.box').css('font-size')
2. $('.box').css({font-size: function( i,curval){
3.     console.log( i )
4.     return 2*parseInt(curval );
5.
6. }})
```

- `click( )` 点击事件

`click( function(){} )` 参数是一个函数

- `selector` 属性

创建jquery对象时 `css`选择器字符串

- `jquery` jquery版本号

所有jquery对象下都有 `jquery` 属性

```
1. $(' .box' ).jquery //
```

### 3、\$( `element` )

传递给一个 `Element` , `document` , `window` 会把对象转为jQuery对象

```
1. $(document)
2. $(this)
```

### 4、\$( `tag` ,object )

`tag` 创建元素，是字符串标签，得带上 `<>` 括号

`object` 是一个对象，对象的属性可以

是 `csss,html,text,width,height,offset,val` 或 `data`,或者是事件类型，

```
1. $('<img/>', {
2.     src:url,
3.     css:{borderWidth:5},
4.     click:handlerClick //添加事件
5. })
```

---

## 三、筛选

1、`first()` ————— 第一个

2、`last()` ————— 最后一个

3、`filter(selector)` ————— 有

4、`not( selector )` ————— 没有

5、`next()` ————— 上一个兄弟

6、`prev()` ————— 下一个兄弟

6 `siblings( [selector] )` —— 除自身外的兄弟节点

用于筛选**同辈元素**的表达式

7、`children()` ————— 子元素

`children()` **只考虑子元素**而不考虑所有后代元素

`children( selector )` 通过可选的表达式来**过滤**所匹配的子元素

8、`has()` ————— 子元素包含

`has( selector )` 匹配子元素中包含selector元素的父级

```
1. $( '.box' ).filter( 'span' ) //
```

9、`find()` ————— 查找指定子元素

```
1. obj.`find(` selector`)` // obj 下的selector元素
```

10、`eq()` ————— 下标

集合中的第几个

11、`index()` ————— 索引

在**索引**就在所有兄弟中的位置

```
1. <div>
2.     <h3>h3</h3>
3.     <h2>h2</h2>
4.     <h3>h3</h3>
5.     <h3 id="obj">h3</h3>
6.     <h2>h2</h2>
7.     <h2>h2</h2>
8. </div>
9. <script>
10. alert( $('#obj').index() ); //索引就是当前元素在所有兄弟节点中的位置
11. </script>
```

## 12、add( selector ) 添加jQuery对象

把与表达式匹配的元素添加到jQuery对象中。形成两个表达式匹配的元素结果集

---

# 三、HTML

## 1、class操作

hasClass() - 是否包含某一个class，不能用函数  
addClass() - 添加 一个或多个类  
removeClass() - 删除 一个或多个类  
toggleClass() - 添加/删除 类的切换操作

---

## 2、设置/获取 元素内容

text() - 设置或返回所选元素的文本内容——会获取所有元素的文本（取值赋值合体特例）  
html() - 设置或返回所选元素的内容（包括 HTML 标记）  
val() - 设置或返回表单字段的值

---

## 2、HTML属性

attr() ——设置/获取属性

```

1. attr( key,val )
2. /////
3. attr( key,function(){ return val;} )
4. //////////
5. attr( { key:val } )
6. ////////////
7. attr( {
8.     key1:val1;
9.     key2:val2;
10.    key3:val3;
11. } )
12. /////
13. attr( {
14.     key1:function(){ return val;};
15.
16. } )

```

`removeAttr()` ———— 设置/获取属性

- `data()` 获取/设置 元素数据——不可传递函数
- `removeData()` 移除数据

```

1. $( '.box' ).data('x',1)           //设置
2.
3.
4. $( '.box' ).data('x')             //获取
5. $( '.box' ).removeData('x')       //移除
6. //////////////////////////////////
7. //可以获取元素自定义data-xxx属性
8. <div id='box' data-scroll='100'></div>
9. <script>
10. $( '.box' ).data('scroll');//100
11. </script>

```

### 3、盒模型

`width()` ———— 获取/设置 宽度

border-box 下，获取的（包含padding border），设置的仅仅content的值（不包含padding border）

`innerWidth()` ———— 获取宽度+padding

`outerWidth()` ———— 获取宽度+padding+border

`outerWidth( true )` ———— 获取宽度+padding+border+margin

## 4、位置

### 1、`offset()`

以文档为参考来计算位置，是一个对象，带有`top`,`left`值。相对于是固定定位

```
offset().top  
offset().left
```

### 设置

```
1. var pos = $(' .box').offset();  
2. pos.top +=100;  
3. $(' .box').offset( pos ) //设置新坐标
```

### 2、`parent()` 一定位父级

### 3、`offsetParent()` 一定位父级

### 4、`position()` ——到定位父级的偏移量

`position()`也是一个对象，包含`top`,`left`。是到定位父级的偏移量——只读

### 5、`scrollTop()` / `scrollLeft()` — 设置/获取 元素滚动条位置

可以用于 `window` , `document` 上，不传递函数

### 分屏滚动

```
1. function page(n){  
2.     var w = $(window);  
3.     var pagesize = w.height();           //得到页面高度  
4.     var current= w.scrollTop();          //得到当前滚动的高度  
5.     w.scrollTop( current + n*pagesize )  //设置新的滚动高度  
6.  
7. }
```