

## 第三课 jquery 基本原理介绍

### 学习目录

- jquery 核心对象
- jquery 与原生 js 的关系

### 一 . jquery 核心对象

在我们之前课程中跟大家演示 jquery 的时候我们在获取选择页面元素的时候，用到了一个\$符号，这个符号实际上就是 jquery 的核心对象的简写形式。这个\$符号，实际上相当于是一个变量，这个变量接收到的是 jquery 的核心对象。

```
window.jQuery = window.$ = jQuery;
```

由上面的代码我们可以看出来，除了\$符号，我们好可以使用 jQuery 这个全局变量来使用 jquery，不管是\$还是 jQuery，他们是一个全局变量，他们都接收 jQuery 这个核心对象，为了使用方便，绝大多数情况下我们都是用\$来使用 jquery 的各种属性和方法。

实际上真实的 jQuery 核心对象的实现原理也是构造函数来做到的，在它的原型对象身上也有许多属性和方法，这也是为什么我们在使用\$选择到某些页面元素之后可以使用 jQuery 各种方法的原因，她实际上和我们开发的普通组件的基础原理差不多。

只不过 jQuery 核心对象并不是在 jQuery 这个对象上，而是在 jQuery.fn.init 上来做到的。

```
var jQuery = function( selector, context ) {  
  
    return new jQuery.fn.init( selector, context );  
  
}
```

由上面的代码我们能够看出来 jQuery 核心对象实际上来自他的 init 对象，为什么 init 对象可以这样被实例化呢？因为 init 对象的原型也被赋值了 jQuery 核心对象的原型了，中间作为过渡的属性是 jQuery.fn 这个属性。这样做的好处是可以不需要我们手动实例化 jQuery 对象就可以直接用 \$ 来使用 jQuery 核心对象上的各种实例属性和实例方法了。

```
jQuery.fn = jQuery.prototype;
```

```
jQuery.fn.init.prototype = jQuery.fn;
```

**总结一下：**

1. jQuery 也是构造函数模式实现的组件化库 其中 jQuery 是这个库的核心对象 \$ 是 jQuery 实例对象的简写形式。
2. 所有的 \$ 选择元素之后的各种属性和方法的调用都来自于 jQuery.prototype 原型对象上。
3. 在使用 jquery 的时候不需要显示的 new 出来一个 jQuery 的实例对象，因为实例化的过程已经在 jquery 的源码里实现了，通过 \$ 选择一个页面元素的时候实际上就是新建了一个 jQuery 的实例对象，因此可以直接调用 jQuery 的各种属性和方法。

## 二 . jquery 与原生 js 的关系

jquery 选择到的元素会得到一个实例对象，很明显这个对象不是 js 的 DOM 对象，因此这个实例对象无法直接使用原生 js 中 DOM 的各种属性和操作方法，反之原生 js 的 DOM 对象也无法直接使用 jquery 的各种属性和方法，因此他们直接就需要一个转换方法。这样就可以保证在开发过程中 js 与 jquery 可以相关转换，使用更多方法灵活开发应用。

## js 转 jquery

这个转换实际上很简单，就是用\$包裹 js 的 DOM 对象即可。

```
var jsDiv1 = document.getElementById( 'div1' );
```

```
var jqueryDiv1 = $(jsDiv1);
```

## jquery 转 js

这个转换有两种方法，两种方式都可以使用，主要根据大家个人使用情况来选择。

第一种方式是下标获取，因为 jquery 对象是一个数据对象的形式，可以使用类数组的 [index]形式来直接得到想要的原生 js 对象。

```
var jqueryDiv1 = $( 'div1' );
```

```
var jsDiv1= jqueryDiv1[0];
```

另一种方法是使用 jquery 提供的 get(index)方法来得到对应的原生 js 对象。

```
var jqueryDiv1 = $( 'div1' );
```

```
var jsDiv1= jqueryDiv1.get(0);
```

## 谢谢观看！

我是星星课堂老师：周小周