jQuery插件

1、认识jQuery extend()和jQuery.fn.extend()

jQuery的API手册中，extend方法挂载在jQuery和jQuery.fn两个不同对象上方法，但在jQuery内部代码实现的是相同的，只是功能却不太一样；

且看官方给出解释：

jQuery.extend(): Merge the contents of two or more objects together into the first object.(把两个或者更多的对象合并到第一个文档中)；

jQuery.fn.extend():Merge the contents of an object onto the jQuery prototype to provide new jQuery instance methods.(把对象挂载到jQuery的prototype属性，来扩展一个新的jQuery实例方法)

2、理解jQuery.extend()

我们先把jQuery看成了一个类，这样好理解一些。jQuery.extend()，是扩展的jQuery这个类。

假设我们把jQuery这个类看成是人类，能吃饭能喝水能跑能跳，现在我们用jQuery.extend这个方法给这个类拓展一个能说话speak()的技能。这样的话，不论是男人，女人，xx人.....等能继承这个技能（方法）了。

a.添加一个新的全局函数

        jQuery.foo = function() {

            document.write("This is a test.");

        };

        $.foo();

b.增加多个全局函数

        jQuery.foo = function(){

            document.write("This is a foo.");

        };

        jQuery.bar = function(param){

            document.write("This is a bar.");

        }

        $.foo();

        $.bar();

c. 使用jQuery.extend(object)添加全局函数

       jQuery.extend({

             sayHello:function(str){

                 console.log(str+"在说话");

             }

         });

   $.sayHello("老王");

d.虽然在jQuery命名空间中，我们禁止使用javascript函数名和变量名。但仍然不可避免某些函数名或变量名将与其他jQuery插件冲突，

  因此我们习惯将一些方法封装到另一个自定义的命名空间。

        $.myFun = {

            sayHello:function(str){

                console.log(str+"在您好");

            },

            sayBye:function(str){

                console.log(str+"在再见");

            }

        }

        $.myFun.sayHello("小明");

        $.myFun.sayBye("蔡徐坤");

案例:实现加减乘除插件

        $.extend({

            option: function(a, b, opt) {

                switch (opt) {

                    case "+":

                        return a + b;

                        break;

                    case "-":

                        return a - b;

                        break;

                    case "\*":

                        return a \* b;

                        break;

                    case "/":

                        return a / b;

                        break;

                }

            },

            fun:function(){

                console.log("heihei");

            }

        });

        console.log($.option(3,5,"\*"));

        console.log($.option(3,5,"-"));

        console.log($.option(3,5,"\*"));

        console.log($.option(3,5,"/"));

        $.fun();

这个扩展也就是所谓的静态方法,只跟这个 类 本身有关。跟你具体的实例化对象是没关系的。

3、理解 jQuery.fn.extend()

从字面理解嘛，这个拓展的是jQuery.fn的方法。jQuery.fn是啥玩意呢？

jQuery.fn = jQuery.prototype = {

      init:funtion(selector,context){

            //.....

     }

}

所以jQuery.fn.extend拓展的是jQuery对象（原型的）的方法啊！

对象是啥？就是类的实例化嘛，例如$("#abc") ，$(div)

那就是说，jQuery.fn.extend拓展的方法，你得用在jQuery对象上面才行啊！他得是张三李四王五痳六这些实例化的对象才能用啊。

说白了就是得这么用（假设xyz()是拓展的方法）：

$('selector').xyz();

调用方法如下：

<!DOCTYPE html>

<html>

    <head>

        <meta charset="utf-8">

        <title></title>

        <style>

        </style>

    </head>

    <body>

        <ul>

            <li></li>

            <li></li>

            <li></li>

            <li></li>

            <li></li>

        </ul>

    </body>

</html>

<script src="myjQuery.js"></script>

<script>

    $(function(){

        $.fn.extend({

            changeStyle:function(){

                $(this).css({

                    background:"red",

                    width:25,

                    height:25

                });

            }

        });

        $("li").eq(1).changeStyle();

    });

</script>

弹出框插件:

<!DOCTYPE html>

<html>

    <head>

        <meta charset="utf-8">

        <title></title>

        <style>

            #box{

                width: 350px;

                height: 400px;

                background: skyblue;

                position:absolute;

                display: none;

            }

        </style>

    </head>

    <body>

        <button>开始</button>

        <div id="box">

        </div>

    </body>

</html>

<script src="myjQuery.js"></script>

<script type="text/javascript">

    $(function() {

        $.fn.extend({

            Popplugin: function() {

                let box = $(this)[0];

                let divbtn = null;

                function setBoxPostion() {

                    box.style.display = "block";

                    box.style.left = window.innerWidth / 2 - box.offsetWidth / 2 + "px";

                    box.style.top = window.innerHeight / 2 - box.offsetHeight / 2 + "px";

                    createBtn();

                }

                function createBtn() {

                    divbtn = document.createElement("button");

                    //按钮一定要先加定位

                    divbtn.style.position = "absolute";

                    divbtn.style.width = 50 + "px";

                    divbtn.style.height = 25 + "px";

                    divbtn.innerHTML = "关闭";

                    box.appendChild(divbtn);

                    //如果要修改元素的位置,一定要让元素存在

                    divbtn.style.left = box.offsetWidth - divbtn.offsetWidth + "px";

                    closeBox();

                }

                function closeBox() {

                    divbtn.onclick = function() {

                        box.style.display = "none";

                    }

                }

                setBoxPostion();

            }

        });

        $("button").click(function(){

            $("#box").Popplugin();

        });

    });

</script>

 4、两者区别总结：

4.1、两者调用方式不同：

       jQuery.extend(),一般由传入的全局函数来调用，主要是用来拓展个全局函数，如$.init()，$.ajax();

       jQuery.fn.extend(),一般由具体的实例对象来调用，可以用来拓展个选择器，例如$.fn.each();

4.2、两者的主要功能作用不同：

        jQuery.extend(object); 为扩展jQuery类本身，为自身添加新的方法。

        jQuery.fn.extend(object);给jQuery对象添加方法

4.3、大部分插件都是用jQuery.fn.extend() 99%

5.引入第三方插件

jquery之cookie操作

Cookies

首先:jQuery cookie插件下载

定义：让网站服务器把少量数据储存到客户端的硬盘或内存，从客户端的硬盘读取数据的一种技术；

下载与引入:jquery.cookie.js基于jquery；先引入jquery，再引入：jquery.cookie.js；下载：http://plugins.jquery.com/cookie/

<script type="text/javascript" src="js/jquery.min.js"></script><script type="text/javascript" src="js/jquery.cookie.js"></script>

1.添加一个"会话cookie"

$.cookie('the\_cookie', 'the\_value');

这里没有指明 cookie有效时间，所创建的cookie有效期默认到用户关闭浏览器为止，所以被称为 “会话cookie（session cookie）”。

2.创建一个cookie并设置有效时间为 7天

$.cookie('the\_cookie', 'the\_value', { expires: 7 });

这里指明了cookie有效时间，所创建的cookie被称为“持久 cookie （persistent cookie）”。注意单位是：天；

3.创建一个cookie并设置 cookie的有效路径

$.cookie('the\_cookie', 'the\_value', { expires: 7, path: '/' });

在默认情况下，只有设置 cookie的网页才能读取该 cookie。如果想让一个页面读取另一个页面设置的cookie，必须设置cookie的路径。cookie的路径用于设置能够读取 cookie的顶级目录。将这个路径设置为网站的根目录，可以让所有网页都能互相读取 cookie （一般不要这样设置，防止出现冲突）。

4.读取cookie

$.cookie('the\_cookie');

5.删除cookie

$.cookie('the\_cookie', null, -1);   //通过传递null作为cookie的值即可

6.可选参数

$.cookie('the\_cookie','the\_value',{

    expires:7,

    path:'/',

    domain:'jquery.com',

    secure:true

})

expires：（Number|Date）有效期；设置一个整数时，单位是天；也可以设置一个日期对象作为Cookie的过期日期；

path：（String）创建该Cookie的页面路径；

domain：（String）创建该Cookie的页面域名；

secure：（Booblean）如果设为true，那么此Cookie的传输会要求一个安全协议，例如：HTTPS；

使用第三方插件的精髓：

求Demo

作业:

1.整理笔记,课堂案例

2.项目需求

    a.七个页面

    b.主页

    c.登录

    d.注册

    e.详情（放大镜）

    f.购物车

    g.前后端交互

    项目日志:项目进度

            当前能够运行的最大功能模块