前端基础培训

jQuery Dom 事件绑定

- on
- bind
- delegate

.on(events [, selector] [, data], handler(eventObject))

添加的版本: 1.7

events

类型: String

一个或多个空格分隔的事件类型和可选的命名空间，或仅仅是命名空间，比如"click", "keydown.myPlugin", 或者 ".myPlugin"。

selector

类型: String

一个选择器字符串，用于过滤出被选中的元素中能触发事件的后代元素。如果选择器是 `null` 或者忽略了该选择器，那么被选中的元素总是能触发事件。

data

类型: Anything

当一个事件被触发时，要传递给事件处理函数的`event.data`。

handler(eventObject)

类型: Function()

事件被触发时，执行的函数。若该函数只是要执行`return false`的话，那么该参数位置可以直接简写成 `false`。

```
bind: function( types, data, fn ) {
        return this.on( types, null, data, fn );
},
unbind: function( types, fn ) {
        return this.off( types, null, fn );
},

delegate: function( selector, types, data, fn ) {
        return this.on( types, selector, data, fn );
},
undelegate: function( selector, types, fn ) {

        // ( namespace ) or ( selector, types [, fn] )
        return arguments.length === 1 ?
                this.off( selector, "**" ) :
                this.off( types, selector || "**", fn );
}
```

事件绑定技巧

- 事件绑定delegate 对动态渲染的内容delegate 其父容器，增强性能。
  使用时注意event.target 和 event.currentTarget 所代表的对象
- 事件绑定使用 namespace 防止绑定事件冲突，在off 事件时可以灵活remove事件。
- 事件中返回return false 会执行
  event.preventDefault();
  event.stopPropagation();

jQuery data attr

HTML5 dataset data-* 属性是 Attribute 的一个子集

```
<input type="text" id="tinp" value="" name="name" data-key="company">
```

```
var dataset = document.getElementById("tinp").dataset;
var data_key = $("#tinp").data("key")
console.log("data_key", data_key);
$("#tinp").data("id", "id123");
$("#tinp").data("obj", { txt: "hello word"});
console.log("datakey", dataset.key);
console.log("dataset",dataset);
```

jQuery remove vs detach

- remove 会 remove events data 并从dom树中移除。
- detach 只从DOM树中移除 , 并不remove events data .

jQuery $

- $(function(){ … })
  $.ready(function(){ … });
- $(selector[,context]) context 查找区域
- var a = $(selector)
  var b = a.find(selector2) (缩小查找范围，缓存已经查找的DOM)

$(htmltext)

原生创建元素绑定事件，在内存中创建元素

```
1  var dom = document.createElement("div")
2  var input = document.createElement("input");
3
4  dom.appendChild(input)
5
6  dom.addEventListener("click", function() {
7      .....
8  }, false)
```

$(htmltext) 2

```
11  var htmltext = "<div class=''><input type='text' name>
    </div>";
12
13  var dom = document.createElement("div");
14  dom.innerHTML = htmltext;
15  var dom1 = dom.getElementsbyTagName("div")[0]
16  dom1.addEventListener("click", function() {
17
18  }, false);
19
```

$(htmltext) 3

```
21
22  var $dom = $("<div class=''><input type='text' name>
    </div>")
23
24  $dom.click(function() {
25
26
27  })
```

通过$(htmltext)声明一个DOM 节点，然后对DOM节点进行操作，建立一个DOM操作的局部作用域。在这个作用域下进行DOM操作，性能会提高，selector冲突问题也会避免。

```
21
22  var $dom = $("<div class=''><input type='text' name>
    </div>")
23
24  $dom.click(function() {
25
26
27  })
```

jQuery 实用函数

- extend
- clone
- proxy
- type
- trim
- contains

$.extend

```
jQuery.extend( target [, object1 ] [, objectN ] )

jQuery.extend( [deep ], target, object1 [, objectN ] )

var object = $.extend({}, object1, object2);

$.extend( true, object1, object2 );
```

中文API
http://www.css88.com/jqapi-1.9/jQuery.extend/

$.clone

.clone([withDataAndEvents]) 添加的版本: 1.0

withDataAndEvents (默认: `false`)

类型: Boolean

一个Boolean值，表示是否会复制元素上的事件处理函数。从jQuery 1.4开始，元素数据也会被复制。

.clone([withDataAndEvents] [, deepWithDataAndEvents]) 添加的版本: 1.5

withDataAndEvents (默认: `false`)

类型: Boolean

一个Boolean值，表示是否会复制元素上的事件处理函数。 默认值是 `false`。*对于1.5.0的默认值被不适当地设置成了`true`，将在1.5.1以上改回`false`。

deepWithDataAndEvents (默认: `value of withDataAndEvents`)

类型: Boolean

一个布尔值，指示是否对事件处理程序和克隆的元素的所有子元素的数据应该被复制。默认情况下它的值相匹配的第一个参数的值（ 默认值是 `false`）

$.proxy

jQuery.proxy(function, context [, additionalArguments]) 添加的版本: 1.6

function

类型: Function()

将要改变上下文语境的函数。

context

类型: PlainObject

函数的上下文语境会被设置成这个 object 对象。

additionalArguments

类型: Anything

任何数目的参数传递给`function`参数的函数引用。

$.type

如果对象是undefined或null，则返回相应的“undefined”或“null”。

```
jQuery.type( undefined ) === "undefined"
jQuery.type() === "undefined"
jQuery.type( window.notDefined ) === "undefined"
jQuery.type( null ) === "null"
```

如果对象有一个内部的[[Class]]和一个浏览器的内置对象的 [[Class]] 相同，我们返回相应的 [[Class]] 名字。（有关此技术的更多细节。）

```
jQuery.type( true ) === "boolean"
jQuery.type( 3 ) === "number"
jQuery.type( "test" ) === "string"
jQuery.type( function(){} ) === "function"
jQuery.type( [] ) === "array"
jQuery.type( new Date() ) === "date"
jQuery.type( new Error() ) === "error" // as of jQuery 1.9
jQuery.type( /test/ ) === "regexp"
```

其他一切都将返回它的类型“object”。

$.contains

jQuery.contains(container, contained) 添加的版本: 1.4

container

类型: Element

DOM元素作为容器，可以包含其他元素

contained

类型: Element

DOM元素，可能被其他元素所包含

$.ajax

- cache : false (get 请求防缓存)
- dataType : json jsonp text html xml
- complete : 成功或者失败都会执行的函数
- contentType (default: 'application/x-www-form-urlencoded; charset=UTF-8’)
- timeout 超时时间
- context 回调函数this 指向context

$.fn

- $.fn = jQuery.prototype
- $.fn.xxx = function() {} //声明一个插件
- function 中的this 是jquery 对象，并不是 dom 原生对象
- function return this 可以继续链式调用
- function 中调用 通过 data 方式存储或者调用实例对象

The End