这个this 真是惹事啊。这里做出总结，可能不全面。但在发现新用法还会整理进来。

this是Javascript语言的一个关键字。

它代表**函数运行时，自动生成的一个内部对象，只能在函数内部使用**。比如，

　(function test(){

alert(this===window);

})() 解释：上面是一个全局函数**window.test().因此返回的是true**。

随着函数使用场合的不同，this的值会发生变化。但是有一个**总的原则，那就是this指的是，调用函数的那个对象。**

下面分四种情况，详细讨论this的用法。

**情况一：纯粹的函数调用**

这是函数的最通常用法，属于全局性调用，因此this就代表全局对象Global。

请看下面这段代码，它的运行结果是1。

　　function test(){

　　　　this.x = 1;

　　　　alert(this.x);

　　}

　　test(); // 1

为了证明this就是全局对象，我对代码做一些改变：

　　var x = 1;

　　function test(){

　　　　alert(this.x);

　　}

　　test(); // 1

运行结果还是1。再变一下：

　　var x = 1;

　　function test(){

　　　　this.x = 0;

　　}

　　test();

　　alert(x); //0

**情况二：作为对象方法的调用**

函数还可以作为某个对象的方法调用，这时this就指这个上级对象。

　　function test(){

　　　　alert(this.x);

　　}

　　var o = {};

　　o.x = 1;

　　o.m = test;

　　o.m(); // 1

**情况三 作为构造函数调用**

所谓构造函数，就是通过这个函数生成一个新对象（object）。这时，this就指这个新对象。

　　function test(){

　　　　this.x = 1;

　　}

　　var o = new test();

　　alert(o.x); // 1

运行结果为1。为了表明这时this不是全局对象，我对代码做一些改变：

　　var x = 2;

　　function test(){

　　　　this.x = 1;

　　}

　　var o = new test();

　　alert(x); //2

运行结果为2，表明全局变量x的值根本没变。

**情况四 apply调用（由于不太懂apply造成这里不理解）**

apply()是函数对象的一个方法，它的作用是改变函数的调用对象，它的第一个参数就表示改变后的调用这个函数的对象。因此，this指的就是这第一个参数。

　　var x = 0;

　　function test(){

　　　　alert(this.x);

　　}

　　var o={};

　　o.x = 1;

　　o.m = test;

　　o.m.apply(); //0

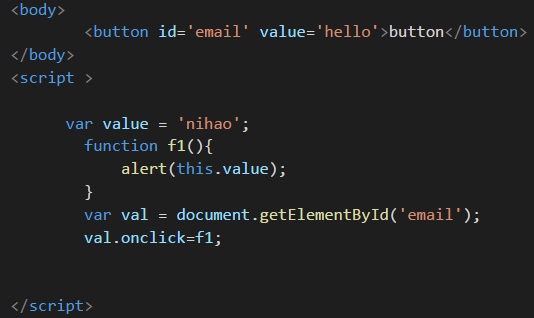
apply()的参数为空时，默认调用全局对象。因此，这时的运行结果为0，证明this指的是全局对象。

如果把最后一行代码修改为

　　o.m.apply(o); //1

运行结果就变成了1，证明了这时this代表的是对象o。

五，作为元素的节点。



当点击button时，返回的是‘hello’ 而不是‘nihao’ 。这是 因为每一个获取的DOM元素也是一个对象；