**二十五 、JQuery设计模式**

**一、单例设计模式**

**1.alert**

alert("我们是最棒的");

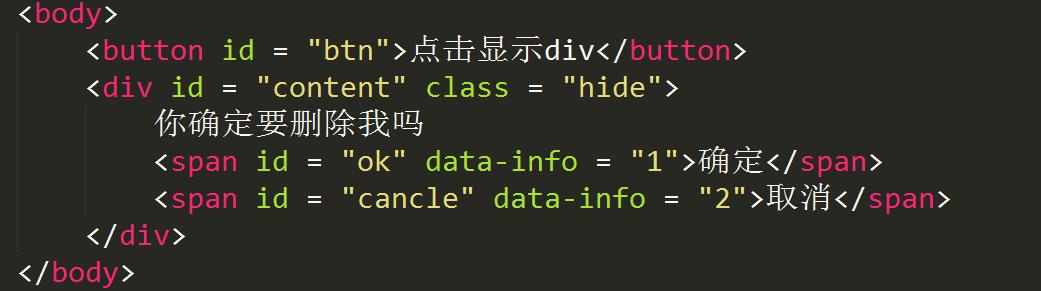
alert("青岛校区最棒的");

【注】所有网页都只使用一个alert。

**2.点击Btn显示和创建唯一的div**

（1）**点击Btn显示div**





**【思考】如果我们要在不同页面或者不同的事件中进行触发，需要是个弹出框，那么我要复制10次吗？**

**【答】将弹出框进行封装**

（2）**点击Btn创建div**



**【注】【缺点】重复点击会一直创建div弹出框**

（3）**点击Btn创建唯一div**

需要在创建该节点之前在页面上找一次，如果存在就不执行后续操作，直接return。



**【缺陷】每次使用的时候 都要在页面上去找。是否能够将该弹出框直接存储在内存中。使用闭包。**

（4）**点击Btn创建唯一div（闭包）**



**二、代理模式**

**1.表白送信**

表白 男孩向女孩表白 但是没有勇气 所以找中间代理人帮忙

对象：一个女孩 一个男孩 中间代理人



**【程序逻辑】**

**（1）声明了三个变量BeautGirl、Boy、proxyObj**

**（2）执行程序new BeautGirl("刘诗诗")，创建了{name：“刘诗诗”}的对象；**

**（3）执行程序new proxyObj(new BeautGirl("刘诗诗"));，创建proxyObj对象**

**（4）执行程序proxyObj.sendGift("李森说他喜欢你")，执行对象的方法sendGift，创建new Boy(this.girl)，将girl对象当做参数传进去。**

**（5）执行new Boy(this.girl).sendEmail(gift)，将李森说他喜欢你作为gift参数传进去，最后alert("hi" + this.girl.name + ", 您有一封新消息：" + email);**

**【结果】**



**2.使用代理加载图片**



**【程序逻辑】**

**（1）声明了一个变量myImg，调用匿名函数后，创建了一个img节点，返回结果：一个对象**

**{**

**setSrc: function(src){**

**alert(3)**

**imgNode.src = src;**

**}**

**}**

**将这个对象赋值给myImg；**

**（2）声明了一个变量proxyImg，调用匿名函数后，创建了一个img对象，然后执行alert（1），给img绑定了onload事件函数，在图片加载完成购才会执行，返回结果：一个对象**

**{**

**setSrc: function(src){**

**myImg.setSrc("img/waite.gif");**

**img.src = src;**

**}**

**}**

**将这个对象赋值给proxyImg；**

**（3）a.执行程序proxyImg.setSrc("img/show.png");**

**b.执行程序myImg.setSrc("img/waite.gif");执行alert(3)，imgNode.src = src;**

**c.此时图片加载完成，执行img的onload函数,执行alert(2)，myImg.setSrc(this.src);**

**d.执行程序myImg.setSrc(this.src);执行alert(3)**

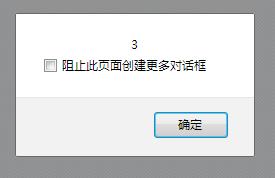
**imgNode.src = src;图片最终加载完成。**

**【执行结果】**

**1.alert（1）；**



**2.alert（3）；**



**3. ①imgNode.src = src; ② alert（2）；**



**4. alert（3）；此时imgNode.src = src;的图片仍然存在**



**5. imgNode.src = src; 此时imgNode.src = src;的图片被替换**

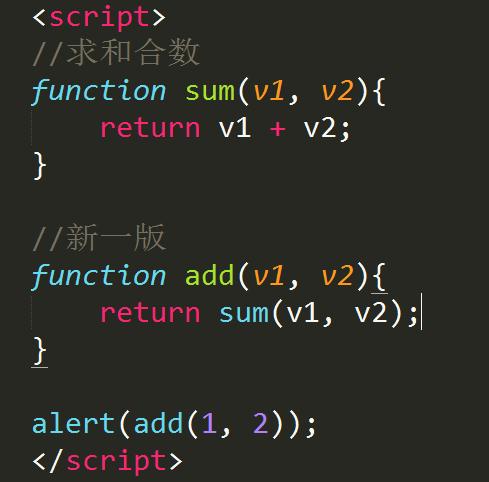


**三、适配器模式**

**1.求和**

【注】原来定义了一个求和函数，但现在想重新定义一个新版本，但又想兼容老版本

**在新函数中，将函数作为返回值**

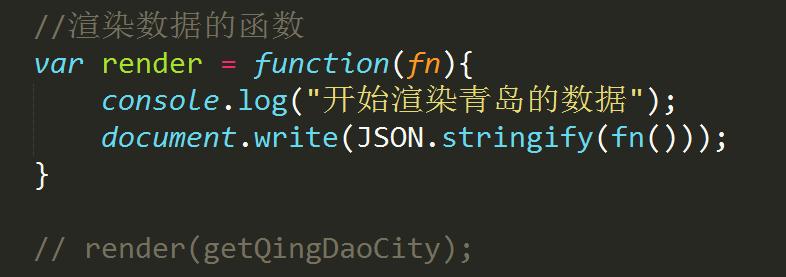


**2.数据渲染**

（1）旧数据



（2）渲染数据的函数



（2）新数据

a.将旧数据遍历一遍

b.将旧数据

