

ajax0523

JSON.parse(); 正常情况下用这个就可以了但是在IE低版本不行。

那还可以用这个eval() 但是不是特别安全会被植入病毒的危险;

最后的new Function()的方式;

eval() 能够将字符串尽可能的转成JS代码执行

ie:10-11 textContent;

ie 9 一下 text

chrome: inner HTML;

还有一个就是onload: 在IE9一下浏览器不兼容;

onreadystatechange存储函数或者函数名字, 每当readyState属性改变的时候, 就会调用该函数

0-4数字 只要数字到了4 说明请求过程完毕

0监控不到的 因为0是创建;

如果onreadystatechange 放在send的前面可以多监听一步请求的步骤, 其实对结果没影响 因为请求结束是四;

http状态码:

200-207成功

4开头的全是错误;

5开头找后端给你没关系

ajax.status可以查看本次请求的状态码

拿到接口之后先看看数据是什么格式的;

同步异步:

页面上边所有的静态资源都是同步的;

同步的意思是: 代码从上倒下, 依次解读, 只要中间无法运行, 下面就不执行;

异步: 从上而下, 如果中间运行不了, 下面依然运行;

```
user.onblur = function() {  
    ajax({  
        url: 'php/get.php',  
        // method: 'GET',  
        data: {  
            user: user.value  
        },  
        success: function(data) {  
            console.log(data)  
            span.innerHTML = data;  
        },  
        fail: function(status) {  
            alert(status);  
        }  
    });  
}
```

```

function ajax(json) {
    //默认配置
    var settings = {
        url:'',
        method:'get',
        success:function() {},
        fail:function() {},
        data:{}
    }
    //有配置走配置，没配置走默认
    for(var attr in json) {
        settings[attr] = json[attr];
    }

    //将一个对象转成字符串 user=momo&sex=男&id=2
    var arr = [];
    for(var i in settings.data) {
        arr.push(i+'='+settings.data[i]);
    }

    settings.data = arr.join('&');

    var ajax = new XMLHttpRequest();

    //如果请求方式是get，在open下的第二个参数就要拼接字段
    //else就是post。要设置请求头信息并且把字段拼接到send中
    if(settings.method.toLowerCase() === 'get') {
        ajax.open(settings.method, settings.url+'?' + settings.data, true);
        ajax.send();
    } else {
        ajax.open(settings.method, settings.url, true);
        ajax.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
        ajax.send(settings.data);
    }

    ajax.onload = function() {
        //当请求成功的时候执行settings.success，失败的时候执行settings.fail
        if(ajax.status >= 200 && ajax.status <= 207) {
            settings.success(ajax.responseText);
        } else {
            settings.fail(ajax.status);
        }
    }
}

```

```

user.onblur = function() {

    ajax({
        url:'php/get.php',
        dataType:'string',
//        dataType:'xml',
//        method:'GET',
        data:{
            user:user.value
        },
        success:function(data) {
            var doc = data;

            console.log(data);
            //var persons = doc.getElementsByTagName('person');

//            //console.log(doc, persons)
//            span.innerHTML = data.msg;

            span.innerHTML = data;
        },
        fail:function(status) {
            alert(status);
        }
    });
}

```

```

function ajax(json) {
    //默认配置
    var settings = {
        url:'',
        method:'get',
        success:function() {},
        fail:function() {},
        data:{},
        dataType:'json' //返回的数据格式，默认为json
    }
    //有配置走配置，没配置走默认
    for(var attr in json) {
        settings[attr] = json[attr];
    }

    //将一个对象转成字符串  user=momo&sex=男&id=2
    var arr = [];
    for(var i in settings.data) {
        arr.push(i+'='+settings.data[i]);
    }
}

```

```

settings.data = arr.join('&');

var ajax = new XMLHttpRequest();

//如果请求方式是get，在open下的第二个参数就要拼接字段
//else就是post。要设置请求头信息并且把字段拼接到send中
if(settings.method.toLowerCase() === 'get') {
    ajax.open(settings.method, settings.url+'?' +settings.data, true);
    ajax.send();
} else {
    ajax.open(settings.method, settings.url, true);
    ajax.setRequestHeader('Content-Type', 'application/x-www-form-
urlencoded');
    ajax.send(settings.data);
}

ajax.onload = function() {
    //当请求成功的时候执行settings.success，失败的时候执行settings.fail
    if(ajax.status >= 200 && ajax.status <= 207) {

        //如果传入json，那么就返回对象格式的数据
        if(settings.dataType === 'json') {
            settings.success(eval('(' +ajax.responseText+')'));
        } else if(settings.dataType === 'xml') {
            //如果传入的是xml，那么返回xml格式的数据
            settings.success(ajax.responseXML);
        } else {
            //如果传入的既不是json也不是xml，那么返回json格式（字符串）的
            settings.success(ajax.responseText);
        }
    } else {
        settings.fail(ajax.status);
    }
}
}

```

数据