

/\*

xml数据格式

元数据：描述数据的数据

这种数据格式的弊端：

- 1、元数据占用的数据量比较大，不利于大量数据的网络传输
- 2、解析不太方便

\*/

```
xhr.onreadystatechange = function() {  
    if(xhr.readyState == 4) {  
        if(xhr.status == 200) {  
            var data = xhr.responseXML;  
            var bookList = data.getElementsByTagName('booklist');  
            // var books = bookList[0].childNodes;  
            var books = bookList[0].children;  
  
            var book1 = books[0];  
            // console.log(books[0].nodeType);  
            // console.log(books[1].nodeType);  
            // console.log(book1);  
            // var name = book1.children[0].innerHTML;  
            // var author = book1.children[1].innerHTML;  
            // var desc = book1.children[2].innerHTML;  
  
            var name = book1.children[0].firstChild.wholeText;  
            var author = book1.children[1].firstChild.wholeText;  
            var desc = book1.children[2].firstChild.wholeText;  
            console.log(name, author, desc);  
        }  
    }  
}
```

// 对象

/\*

json数据和普通的js对象的区别:

- 1、json数据没有变量
- 2、json形式的数据结尾没有分号
- 3、json数据中的键必须用双引号包住

\*/

// var obj = {

// name: 'zhangsan',

// age : 12,

// lover : ['coding', 'singing', 'dancing'],

// friend : {

// high : '180cm',

// weight : '80kg'

// }

// };

**var str** = ' {"name": "zhangsan", "age": "12"} ';

**var obj** = **JSON.parse(str)**; //把json形式的字符串转成对象

**var str1** = **JSON.stringify(obj)**; //把对象转成字符串

**console.log(str1)**; // {"name": "zhangsan", "age": "12"}