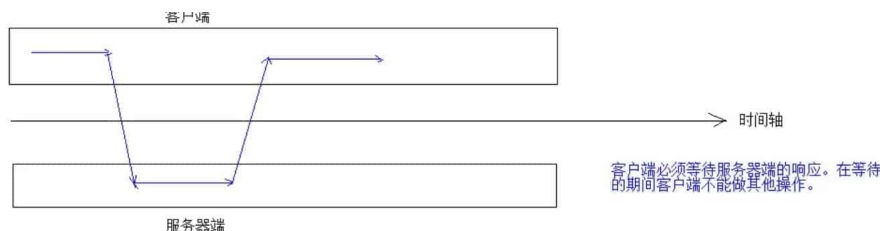


AJAX

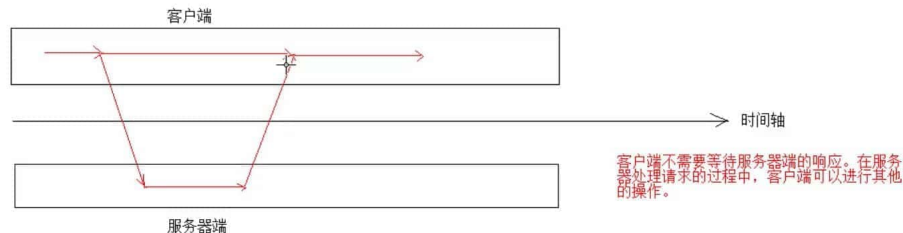
一、简介

1. AJAX 的含义：异步的 JavaScript 和 XML
2. 对异步的解释

同步



异步



3. 在操作上的不同：无需加载整个页面，只更新部分数据；传统网页需要加载额外的资源

二、原生 JS 方式

1. 定义方法，并创建相关对象

```
//定义方法
function fun() {
    //发送异步请求
    //1.创建核心对象
    var xmlhttp;
    if (window.XMLHttpRequest)
    {
        // code for IE7+, Firefox, Chrome, Opera, Safari
        xmlhttp=new XMLHttpRequest();
    }
    else
    {
        // code for IE6, IE5
        xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
    }
}
```

2. 建立连接

```
//2. 建立连接
/*
    参数：
    1. 请求方式：GET、POST
        * get方式，请求参数在URL后边拼接。send方法为空参
        * post方式，请求参数在send方法中定义
    2. 请求的URL：
    3. 同步或异步请求：true（异步）或 false（同步）
*/
xmlhttp.open("GET","ajaxServlet?username=tom",true);
```

3. 发送请求

```
//3.发送请求
xmlhttp.send();
```

4. 接收响应

```
//4.接受并处理来自服务器的响应结果
//获取方式：xmlhttp.responseText
//什么时候获取？当服务器响应成功后再获取

//当xmlhttp对象的就绪状态改变时，触发事件onreadystatechange。
xmlhttp.onreadystatechange=function()
{
    //判断readyState就绪状态是否为4，判断status响应状态码是否为200
    if (xmlhttp.readyState==4 && xmlhttp.status==200)
    {
        //获取服务器的响应结果
        var responseText = xmlhttp.responseText;
        alert(responseText);
    }
}
```

二、JQuery 方式

1. ajax

```
//定义方法
function fun() {
    //使用$.ajax() 发送异步请求

    $.ajax({
        url: "ajaxServlet", // 请求路径
        type: "POST", // 请求方式
        //data: "username=jack&age=23", // 请求参数
        data: {"username": "jack", "age": 23},
        success: function (data) {
            alert(data);
        }, // 响应成功后的回调函数
    });
}
```

2. get

```
function fun() {

    $.get("ajaxServlet", {username: "rose"}, function (data) {
        alert(data);
    }, "text");
}
```

3. post

```
$.post("ajaxServlet", {username: "rose"}, function (data) {
    alert(data);
}, "text");
```