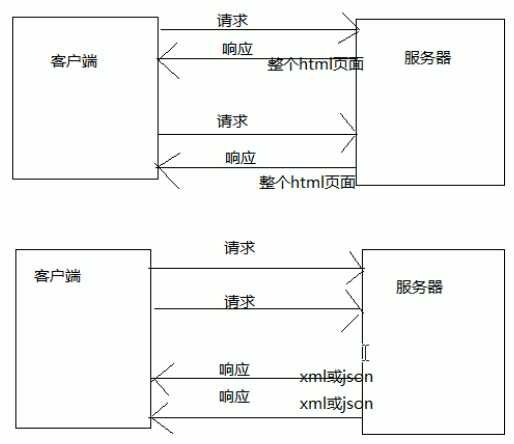
# Ajax

## Ajax是啥？

异步的 js 和 xml .

* 能使用 js 访问服务器，而且是异步访问；
* 服务器给客户端响应的一般是整个页面，一个 html 完整的页面！但是在 ajax 中因为是局部刷新，那么服务器就不用在响应整个页面！而只是数据！
* 数据交换格式：
* Text:纯文本；
* Xml:
* Json: js 提供的数据交互格式，在 ajax 中最受欢迎！
* 示意图



* 应用场景
* 百度的搜索框 下拉提示
* 用户注册时，实时校验用户名是否已被注册。
* 优点
* 异步交互，增强了用户体验；
* 性能：因为服务器无需在响应整个页面，只需要响应部分内容，所以服务器的压力减轻了；
* 缺点
* Ajax 不能应用在所有场景；
* Ajax 无端的增多了对服务器的访问次数，又给服务器增加了压力！

### 同步交互和异步交互

* 同步
* 发一个请求，要等待服务器的响应结束，然后才能发送第二个请求，中间的时间处于“卡”的阶段。
* 刷新的是整个页面。
* 异步
* 发送一个请求，无需等待服务器的响应，然后就可以发送第二个请求！
* 可以使用 js 接受服务器的响应，然后使用 js 进行局部刷新。

## Ajax 发送异步请求（四步）

### 第一步（得到 XMLHttpRequest）

* Ajax 只需要学习一个对象：XMLHttpRequest,如果掌握了它，就掌握了 ajax;
* 得到 XMLHttpRequest
* 大多数浏览器都支持 var XMLHttp = new XMLHttpRequest();
* IE6.0: var XMLHttp = new ActiveXObject(“Msxml2.XMLHTTP”);
* IE5.5以及更早版本的IE: var xmlHttp = new ActiveXObject(“Microsoft.XMLHTTP”);
* 编写创建 XMLHttpRequest 对象的函数：

Function createXMLRequest(){

Try{

Return new XMLHttpRequest();

}catch{

Try{

Return new ActiveXObject(“Msxml2.XMLHTTP”);

}catch{

Try{

Return new ActiveXObject(“Microsoft.XMLHTTP”);

}catch{

Alert(“您用的是啥浏览器啊？”)；

Throw e;

}

}

}

}

### 第二步（打开与服务器的连接）

* xmlHttp.open():用来打开与服务器的连接，需要三个参数：
* 请求的方式：GET/POST;
* 请求的URL:指定服务器端的资源，例如：/project/servlet;
* 请求是否为异步：如果为true表示发送异步请求，否则为同步请求；
* xmlHttp.open(“GET”,” /project/servlet”,true);

### 第三步（发送请求）

* xmlHttp.send(null)

如果不给参数可能会造成部分浏览器无法发送；

* 参数：就是请求体的内容！如果是 GET 请求，必须给出 null.