伴随着web的强大而出现的是等待，等待服务器响应，等待屏幕刷新，等待请求返回和生成新的页面，以及，用户的咆哮。那Ajax的作用呢，就是最大化的减小这种等待。

使用Ajax需要了解多门基础技术：

* HTML表单
* JavaScript代码
* 动态HTML
* DOM

不过大部分内容你估计都了解，甚至熟练。从这一点看，Ajax就是将老技术整合，然后绽放出新的光辉。（也不光是Ajax吧，很多框架都是这样。以前看别人黑程序员，说是技术更新极快，日新月异。但本质上都是换汤不换药的，底子打的劳，上手框架分分钟的事。）

////////////////////////////////////////////////////////////////////////////////////////////

~Ajax流程

1. XMLHttpRequest

这是一个JavaScript对象，是处理所有服务器通信的对象，创建它很简单。

var xmlHttp = null;

if(window.XMLHttpRequest){

xmlHttp = new XMLHttpRequest();

}else{

// 这个else是为IE5/6准备的，唉，也就中国还在用着些鬼东西了。

xmlHttp = new ActiveXObject(“Microsoft.XMLHTTP”); // 字符串

}

Ajax把JavaScript和XMLHttpRequest对象放在web表单与服务器之间。当用户填写表单时，数据会传送给JavaScript代码而不是服务器。之后，JavaScript代码接收数据并向服务器提出请求，此时，用户屏幕上的表单不会闪烁，消失或延迟。

然后，服务器将数据返回给JavaScript代码，后者决定如何处理这些数据。可以直接更新表单，或者做些处理，再度发出一个请求。（所以呀，要熟练JavaScript哟）

这便是XMLHttpRequest对象的强大之处，它可以根据需要自行与服务器进行交互，用户可以完全不知道幕后发生的一切。

1. 请求/响应

（JavaScript还是很强大的，有很多功能，像用户名，密码判空都可以直接交给JavaScript做，减轻servlet的负担。）

以下是一个在所有Ajax应用程序中都雷同的流程：

1. 从web表单获取数据
2. 建立要连接的URL
3. 打开到服务器的连接
4. 设置服务器在完成后要调用的函数
5. 发送请求

那个获取数据和建立URL就不讲了，几个字符串。3-5步一般以以下的形式展现。

xmlHttp.open(“GET”, url, true);

xmlHttp.onreadystatechange = ...; // 某函数

xmlHttp.send(null);

分别代表，打开连接，设置完成后调用函数，与发送请求。这里的例子用GET方法，我个人偏好POST，但GET传数据确实方便。url是要建立的连接，true代表异步处理。

onreadystatechange完成后要调用的函数，比如更新数据什么的。Send，发送请求。这里参数为null是因为上面已经用了GET方法，不需要传参了。

关于响应暂时只需要知道两点

1. 等待xmlHttp.readyState的值变为4
2. 服务器将响应填充至xmlHttp.responseText

以下是一个JavaScript小函数

function change(){

if(xmlHttp.readyState == 4){

var response = xmlHttp.responseText;

// 然后用response做爱做的事，比如赋给其他引用。

}

}

眼睛瞪大了嘿，readyState的S是大写，他奶奶的大写！

////////////////////////////////////////////////////////////////////////////////////////////

~XMLHttpRequest详解

1. 常用函数及属性

* open()
* send()
* abort()
* readystate
* responseText

以上只是Ajax中很小的一部分（爷的心，深似大海）

\* 2. 创建带有错误检查的XMLHttpRequest对象，这里还包括处理IE浏览器的异常问题。唉，东西一大堆，日后再说吧。下面看个小例子，其实就是加了个try-catch

var request = false;

try{

request = new XMLHttpRequest();

}catch(failed){

request = false;

}

if(!request){

alert(“错误信息”);

}

1. HTTP就绪状态及状态码

0：请求没有发出（open()之前）

1：请求已经建立但还没有发出（send()之前）

2：请求已经发出，正在处理

3：请求已经处理，但服务器还没有完成响应

4：响应已完成，可以访问服务器响应并使用

之前提到，服务器在完成请求之后会在XMLHttpRequest.onreadystatechange中寻找要调用的函数，但你看，上面有那么多就绪状态，其实每改变一个都会尝试调用方法。另外web各平台之间的兼容是很混乱的，所以你不能指望浏览器每次都按照你要的顺序来提示就绪状态。

对于Ajax来讲，唯一需要直接处理的状态就是4

状态码，就是标志状态的编码（这话是我说的，不知道对不对）。生活中经历比较多的状态码有404：页面不存在、403：访问数据受到保护、401：禁止访问。下面是比较具体的。

* 1xx：提示性信息
* 2xx：请求成功
* 3xx：重定向
* 4xx：客户端错误
* 5xx：服务器端错误

（这里的1,2,3和就绪状态没什么关系）

我们期望的状态码是200，表示一切顺利。因此更完美的处理服务器响应的方式就是

在就绪状态为4，且状态码为200时写下你的JavaScript代码。