Javascript， cheer up。



Ajax：其通过在Web页面与服务器之间建立一个额外的处理层，这个处理层就被称为Ajax引擎，它解释来自用户的请求，在后台以异步的方式处理服务器通信，其结构如下图所示。



XMLHttpRequest对象：这个是Ajax的核心对象，其让Javascript构建的HTTP请求并提交给服务器，这样页面就能以异步方式在后台产生请求，让用户可以继续使用这个页面，而不必等待浏览器刷新或加载新的页面。接下来通过一个表格对XMLHttpRequest对象有个总体的了解。

|  |  |
| --- | --- |
| 属性与方法 | 描述 |
| onreadystatechange | 当对象的readyState属性改变时，调用相应的时间处理器 |
| readyState | 0:未初始化  1:正在加载  2：加载完成  3：交互  4：完成 |
| responseText/rsponseXML | 从服务器返回的字符串/文档对象形式数据 |
| status/statusText | 服务器返回的HTTP状态代码，解释短语 |
| setRequestHeader(x, y) | 设置请求头 |
| abort() | 停止当前请求 |
| getAllResponseHeaders()  getResponseHeader(x) | 以字符串形式返回全部响应头信息  返回指定响应头信息 |
| open(method, URL, a) | 指定HTTP方法GET/POST,目标URL和处理请求方法(默认a=true，表示异步) |
| send(content) | 发送请求，对于POST数据是可选的 |

最后通过一个简单的Ajax示例库来对其有个详细的了解。

|  |
| --- |
| function getXMLHttpRequest() {  try {  try {  return new ActiveXObject("Microsoft.XMLHTTP"); //IE  } catch (e) {  return new ActiveXObject("Msxml2.XMLHTTP");  }  } catch (e) {  return new XMLHttpRequest(); //Other  }  }  function doAjax(url, query, callback, reqtype, getxml) {  var myreq = getXMLHttpRequest();  myreq.onreadystatechange = function() {  if (myreq.readyState == 4) {  if (myreq.status == 200) {  var item = myreq.responseText;  if (getxml == 1) item = myreq.responseXML;  eval(callback + '(item)'); //注意这人的callback的使用方式  }  }  }  if (reqtype.toUpperCase() == "POST") {  requestPost(url, query, myreq);  } else {  requestGET(url, query, myreq);  }  }  function requestGET(url, query, req) {  var myRandom = parseInt(Math.random() \* 99999999);  if (query = '') {  var callUrl = url + '?rand=' + myRandom;  } else {  var callUrl = url + '?' + query + '&rand=' + myRandom;  }  req.open('GET', callUrl, true);  }  function requestPost(url, query, req) {  req.open("POST", url, true);  req.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');  req.send(query);  } |

Tip:

由于浏览器会缓存get请求的返回的数据，因此可以在请求参数中添加一个随机数来避免从缓存中读取页面，可以参考上例中的get请求。

此外，使用Ajax时还需要注意：“返回”按钮，书签和链接，注意添加用户反馈（mask），让Ajax平稳退化，应对搜索引擎嗅探，突出活跃页面元素，对用户输入的转义等。

Javacript库非常的多，例如最老的Prototype，常见的JQuery以及其移动版本Zepto，复杂一点的Angularjs，一次编写多处使用的ReactJs等，这部分将选择最常见的JQuery做一个简单的介绍。JQuery是最常见的Javascript库，其扩展性非常强，其语法中最具特色的当属$(document).ready(function() {});其省略形式为：$(function() {});

JQuery的常用选择器如下表所示。

|  |  |
| --- | --- |
| 选择器使用示例 | 描述 |
| .$("span") | 全部的span元素 |
| $("#elem") | Id为elem的元素 |
| $(".classname") | 类为classname的元素 |
| $("div#elem") | Id为elem的<div>元素 |
| $("ul li a.menu") | 类为menu且嵌套在列表项中的锚点 |
| $("p > span") | P的直接字元素span |
| $("input[type=password]") | 具有指定类型的输入元素 |
| $("p:first") | 页面上的第一个段落 |
| $("p:even") | 全部偶数段落 |
| JQuery自定义选择器 | $(":header"),$("p:button"),$("p:radio"),$("p:checkbox")  $("p:checked") |

JQuery的常用函数如下表所示。

|  |  |
| --- | --- |
| 函数 | 描述 |
| .html() | 获取元素的HTML内容（包括文本），类似js中的innerHTML |
| .text() | 获取元素的文本内容 |
| .attr(x) | 获取特定属性的值 |
| .show(200, function(){})  .hide, .toggle | 显示或隐藏元素，第一个参数为显示延迟的毫秒数  Toggle可以方便切换显示和隐藏状态 |
| .fadeOut(),.fadeIn()  .fadeTo(300, 0.5) | 淡出/淡入操作，fadeTo可以指定最终的不透明度 |
| .slidedown(),.slideUp()  .slideToggle() | 元素向下或向上滑动 |
| 动画 | .animate({width:'400px', height:'500px'}, 1500) |
| 命令链 | $('x').fadeout().fadeIn(); |
| 处理事件 | 原来: $().click(function(event) {});  现在: $().on('click', function(event) {});  此外，该方法还支持blur,focus,keypress,change  ,mousemove,resize,scroll,submit,select等事件 |
| Ajax | $('div:first').**load**("content.html #info");  #info表示选取该html中指定容器中的内容  $.get(url,{par1:'val1', par2:'val2'},function () {});  Post方法与get方法格式基本相同  .ajax()参数很多，可以参考文档 |
| 表单元素序列化 | var formdata=$('#form01').serialize(); |

Javascript与多媒体：多媒体内容通常保存在媒体文件中，由于这类文件都非常的，通常通过不同的编码方式压缩文件大小，比如对于音频最常见的就是MP3，对于视频来说，最常见的就是MPEG，Flash，mp4等格式。在页面中加载和播放多媒体内容的方式有如下几种。

|  |  |
| --- | --- |
| 方式 | 描述 |
| 锚点标签(不推荐) | <a href="sound.mp3">Play Sound</a> |
| <embed> | <embed src='music.mp3' autostart='true' loop='false'> |
| <object> | <object type="audio/x-wav" data="music.wav" width="200" height="75">  <param name="src" value="music.wav">  <param name="autostart" value="true">  </object> |
| Flash(ShockWave) | Play(),StopPlay(),IsPlaying(),播放,停止播放,是否正在播放  GotoFrame(x),跳过第x帧  TotalFrames(x),统计影片总帧数  Rewind(),跳过第一帧，并且停止播放  Zoom(pecent),缩放视窗，与习惯思维有点差异，50表示两倍,0表示原尺寸，200表示2分之1  PercentLoaded(),获取影片下载的百分比 |
| <video>(推荐) | <video id='vd01' width="400" height="300" controls="controls">  <source src="movie.mp4" type="video/mp4">  <source src="movie.ogg" type="video/ogg">  </video>  属性:loop循环播放,autoplay自动播放,controls显示回放控件,ended/paused判断状态,poster设置影片加载时显示的图像,volume音量  Tip:  可以使用canPlayType(type)来判断浏览器是否支持该格式，比如video/webm  常见方法: .play(), .pause() |
| <audio> | <audio src='music.mp3' autostart loop/>  常见格式:audio/ogg,xxx.mp3(audio/mpeg)  常见属性;.duration, .currentTime(跳到xx秒), .playbackRate, .muted |

通过一个示例来了解flash的调用

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8">  <meta http-equiv="X-UA-Compatible" content="IE=edge">  <meta name="viewport" content="width=device-width, initial-scale=1">  <title>flash示例</title>  <link href="../public/css/bootstrap.css" rel="stylesheet">  <link href="../public/css/bootstrap-theme.css" rel="stylesheet">  <script type="text/javascript" src="../public/js/jquery-2.1.4.js"></script>  <script type="text/javascript">  function flashLoaded(theMovie) {  if (typeof(theMovie) != 'undefined') {  return theMovie.PercentLoaded() == 100;  } else {  return false;  }  }  function play() {  if (flashLoaded(demo) && !demo.IsPlaying()) {  demo.Play();  }  }  function stop() {  if (flashLoaded(demo) && demo.IsPlaying()) {  demo.StopPlay();  }  }  function rewind() {  stop();  if (demo.Rewind()) {  demo.Rewind();  }  }  $(function() {  document.getElementById('play').onclick = play;  document.getElementById('stop').onclick = stop;  document.getElementById('rewind').onclick = rewind;  //var demo = $('#demo');  var demo = document.getElementById('demo');  });  </script>  </head>  <body>  <article>  <!-- <embed id='demo' src='example.swf' width="318" height="300" play='false' loop='false' pluginspage='http://www.macromedia.com/go/getflashplayer' swliveconnect='true'></embed> -->  <embed id='demo' src='flash4611.swf' width="800" height="600" play='false' loop='false'></embed>  <form id='form'>  <button id="play" type="button" class="btn btn-success" data-dismiss="modal">Start</button>  <button id="stop" type="button" class="btn btn-warning" data-dismiss="modal">Stop</button>  <button id="rewind" type="button" class="btn btn-danger" data-dismiss="modal">Rewind</button>  </form>  </article>  <script type="text/javascript" src="../public/js/bootstrap.js"></script>  </body>  </html> |

在HTML5中，新增了很多的HTML元素，其中有一部分比较常用，在此进行一个简单介绍。

|  |  |
| --- | --- |
| 方式 | 描述 |
| <canvas> | 提供页面上一个矩形区域，用于绘制图形  常见函数: canvas.getContext('2d)获取2d画布  fillRect(x,y,width,height)绘制填充矩形,strokeRect()绘制矩形框，  clearRect()清除矩形,moveTO(x,y)移动到指定位置,lineTo(x,y)绘制直线,  arc(x,y,r,startAngle,endAngle,anti)绘制弧线  object.beginPath(); object.clothPath()封闭形状,object.stroke()绘制轮廓  object.fill()填充形状 |
| 拖放 | <div draggable='true'>  对dragstart,drag,dragenter,dragleave,dragover,drop,dragend等事件进行监控，此外，相应的数据会包含在evnet.dataTransfer属性中,可以通过.setData(), .getData()对其进行操作  ev.preventDefault();  ev.dataTransfer.serData('Text',en.target.id); |
| 本地存储 | 是cookie的替代者，包括localStorage和sessionStorage  localStorage.setItem(key,value);localStorage.getItem(key),数组形式也行 |
| 操作本地文件 | File:提供名称、大小、MIME类型、文件句柄等信息  FileList:类似数据的文件对象列表  FileReader:使用File异步读取文件,可以查看读取进程、捕获错误、判断文件加载状态 |

Canvas示例如下所示（旋转的小球）。

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8">  <meta http-equiv="X-UA-Compatible" content="IE=edge">  <meta name="viewport" content="width=device-width, initial-scale=1">  <title>canvas示例</title>  <link href="../public/css/bootstrap.css" rel="stylesheet">  <link href="../public/css/bootstrap-theme.css" rel="stylesheet">  <script type="text/javascript" src="../public/js/jquery-2.1.4.js"></script>  <script type="text/javascript">  var context;  var x = 50;  var y = 50;  var counter = 0;  function paint() {  context.beginPath();  context.fillStyle = '#ff0000';  context.arc(x, y, 15, 0, Math.PI \* 2, false);  context.closePath();  context.fill();  }  function animate() {  context.clearRect(0, 0, 400, 300);  counter++;  x += 20 \* Math.sin(counter);  y += 20 \* Math.cos(counter);  paint();  }  $(function() {  context = canvas1.getContext('2d');  setInterval(animate, 100);  });  </script>  </head>  <body>  <article>  <canvas id='canvas1' width='400' height="300">  <p>你的浏览器不支持canvas</p>  </canvas>  </article>  <script type="text/javascript" src="../public/js/bootstrap.js"></script>  </body>  </html> |

拖放操作结合文件API的示例如下所示

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8">  <meta http-equiv="X-UA-Compatible" content="IE=edge">  <meta name="viewport" content="width=device-width, initial-scale=1">  <title>HTML5 文件API</title>  <link href="../public/css/bootstrap.css" rel="stylesheet">  <link href="../public/css/bootstrap-theme.css" rel="stylesheet">  <style type="text/css">  body {  background-color: #ddd;  font-family: arial, verdana, sans-serif  }  #drop1 {  width: 400px;  height: 20px;  border: 1px solid black;  background-color: white;  padding: 10px;  }  </style>  <script type="text/javascript" src="../public/js/jquery-2.1.4.js"></script>  <script type="text/javascript">  function allowDrop(ev) {  ev.preventDefault();  }  function drop(ev) {  var files = ev.dataTransfer.files;  for (var i = 0; i < files.length; i++) {  var f = files[i];  var pnode = document.createElement("p");  var tnode = document.createTextNode(f.name + ' (' + f.type + ') ' + f.size + ' bytes');  pnode.appendChild(tnode);  ev.target.appendChild(pnode);  }  ev.preventDefault();  }  $(function() {  var drophere = $('#drop01')[0];  drophere.ondragover = allowDrop;  drophere.ondrop = drop;  });  </script>  </head>  <body>  <div id='drop01'></div>  <output id='text'></output>  <script type="text/javascript" src="../public/js/bootstrap.js"></script>  </body>  </html> |

最后介绍一个比较有意思的，在chrome中添加自己的扩展，新建一个文件夹，在其中放入一个html页面和一个icon，然后编辑如下manifest.json文件。就可以在chrome的Tools -> Extensions ->Load unpacked extension来加载自己的扩展程序了。

|  |
| --- |
| {  'name': 'MyExtension',  'version': '1.0',  'description': 'xionger is handsome',  'browser\_action': {  'default\_icon': 'icon.png',  'popup': 'demo.html'  }  } |

参考资料：

1. Phil, Ballard. JavaScript入门经典（第五版）[M]. 北京:人民邮电出版社, 2013.