# 第三章HTML5中的文件与拖放

1.掌握files对象

2. FileReader对象的使用方法

3.FileReader对象事件

4.掌握HTML5新增拖拽的api

5.掌握dataTransfer的使用方法

在HTML5中，从Web网页上访问本地文件系统非常简单，那就是使用它新增的API，files对象。

一) files对象的使用方法

|  |  |
| --- | --- |
| 属性 | 描述 |
| name | 文件名称 |
| size | 文件大小 |
| type | 文件类型 |

注意:用户每个选择的文件都是一个files对象，而fileList对象则为这些files对象的列表，代表用户选择的所有文件。

**练习1:打印出上传文件的详细信息**

**练习2:进行上传文件后缀判断**

二) FileReader接口读取文件

FileReader接口主要用来把文件读入内存，并且读取文件中的数据。FileReader接口提供了一个异步的api，使用该api可以在浏览器主线程中异步访问文件系统，读取文件中的数据。

|  |  |  |
| --- | --- | --- |
| 方法 | 参数 | 描述 |
| readAsDataUrl | file | 将文件读取为DataURL |
| readAsText | file | 将文件读取为文本 |

2.1：readAsDataUrl:该方法将文件读取为一段以data开头的字符串，这段字符串实际上是dataURL.，并且读取结果都将存储在result属性中。

2.2:readAsText该方法会读取文件中的文本元素，并且读取结果都将存储在result属性中。

注意:使用FileReader的时候，必须使用new运算符，创建出对象，并且调用readAsDaraUrl方法读取文件，在实例的onload事件中，获取到成功读取到的文件内容。

**练习3：使用readAsDataUrl预览图片**

**思考题;如何实现多个图片预览上传？**

**练习4：使用readAsText读取文本文件**

1. **FileReader事件**

FileReader接口中包含了一套完整的事件模型，用于捕获读取文件的状态。

|  |  |
| --- | --- |
| 事件 | 描述 |
| onabort | 中断时触发 |
| onerror | 出错时触发 |
| onload | 文件读取成功完成时触发 |
| onloadstart | 读取完成触发，无论成功或失败 |
| onloadend | 读取时触发 |
| onprogress | 读取中（接受事件对象） |

**练习5：ajax异步上传文件(带进度条)**

**注意：文件上传是属于数据流的格式，那么我们就要注意，就要使用HTML5中的FormData对象**

FormData对象用以将数据编译成键值对，以便用[XMLHttpRequest](https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest)来发送数据。其主要用于发送表单数据，但亦可用于发送带键数据(keyed data)，而独立于表单使用。

jquery中ajax方法，用法如下:

|  |
| --- |
| var fd = new FormData(document.querySelector("form"));  fd.append("CustomField", "This is some extra data");  $.ajax({  url: "stash.php",  type: "POST",  data: fd,  processData: false, // 不处理数据  contentType: false // 不设置内容类型  )} |

四) 常见的拖放api

在HTL5中，支持直接拖放操作的api。虽然HTML5之前已经可以使用mousedown,mousemove和mouseup来进行操作，但只支持在浏览器内部的拖放；而在HTML5中，已经支持在浏览器与其它应用程序之间的数据互相拖动。

**实现拖拽的步骤**

1. **把需要拖放的对象元素的draggable属性设为true(draggable=true)。这样才能将该元素进行拖放。**
2. **编写与拖放有关的事件处理代码。**

**练习6:原生js实现拖拽**

拖拽api



**练习7:拖拽显示拖放信息**

**练习8：拖拽实现购物车效果**

五) 总结

主要介绍了文件api和拖放api。在文件api中主要介绍了fileList对象与file对象，以及FileReader接口。通过这些文件的对象和接口，可以实现文件上传与文件预览等操作。介绍了dataTransfer对象的属性与方法的介绍，通过使用文件的api与拖放API，对于web页面上访问本地文件系统的相关操作，将会变得很简单。