

## 第二章 HTML5 拖放

### 2.1 拖放

#### 2.1.1 什么是拖放

一个元素上点击并按住鼠标按钮，拖动它到别的位置上，然后松开鼠标按钮将元素放到那儿。在 HTML5 中，拖放是标准的一部分，任何元素都能够拖放。

问题：哪里会用到？

#### 2.1.2 拖放实现

1. 传统的拖放是根据鼠标按下 松开事件完成的。
2. HTML5对于拖放只需要在元素上面设置draggable = "true" 属性

```
<div class="mydiv" draggable="true"></div>
```

draggable属性有三个值：

- true 元素可以被拖放
- false 元素不能被拖放
- auto 浏览器自己判断元素是否能被拖放

#### 2.1.3 浏览器兼容性

Internet Explorer 9+, Firefox, Opera, Chrome, 和 Safari 支持拖动。

**注意:**Safari 5.1.2不支持拖动

### 2.2 拖放事件

#### 拖元素

dragsatrt	当一个元素开始被拖拽的时候触发
drag	这个事件在拖拽源触发
dragend	拖拽源在拖拽操作结束dragend事件对象，不管操作成功与否

#### 放元素

dragenter	当拖拽中的鼠标第一次进入一个元素（目标元素）的时候触发
dragover	当拖拽中的鼠标移动经过一个元素的时候触发
dragleave	当拖拽中的鼠标离开元素时触发
drop	这个事件在拖拽操作结束释放时于释放元素上触发

注意：

- 在拖拽的时候只会触发拖拽的相关事件，鼠标事件，例如mousemove，是不会触发的

- drop事件需要阻止原有的dragover默认事件,不然不会触发

```
//第一种方式
事件源.ondragover=function(){
    return false;
}

//第二种方式
事件源.ondragover=function(e){
    e.preventDefault();
}
// event 对象代表事件的状态，比如键盘按键的状态、鼠标的位置、鼠标按钮的状态等，也叫事件对象
// 简单理解：事件发生后，跟事件有关的一系列信息数据的集合都放到整个对象里面，这个对象就是事件对象event，它有很多属性和方法。
```

一般在一个元素开始被拖拽的时候，触发dragstart时候设置与这次拖拽相关的信息，例如拖动的数据和图像。所有拖拽事件都有一个属性-----dataTransfer，它包含着拖拽数据。

dataTransfer的属性

属性	说明
files	其属性返回和放置相关的所有文件
types	属性使用数组的形式返回当前注册的格式
effectAllowed	此属性的意思为通知浏览器当前可被用户选用的操作，如果只是copy，则只允许执行copy操作
dropEffect	拖放的操作类型，决定了浏览器如何显示鼠标形状，值有：copy、move、link和none
items	属性返回所有项与相关格式的所有文件

dataTransfer的方法

方法	说明
setData(format,date)	在dragstart事件调用此函数在dataTransfer对象上存储数据
getData(format)	从dataTransfer对象上取出数据
setDragImage(element,x,y)	使用存在的图像元素作为拖动图像
addElement(element)	调用此函数需要提供一个页面元素作为参考，同时使用此参数作为拖放反馈图像
clearData()	不传参数，表示清空所有已注册数据，带参则清除指定的注册数据

## 2.3 上传外部文件

注意：在拖拽外部文件的时候，为了避免原有的打开文件、下载文件等事件，需要阻止原有的默认行为

```
//1. 获取页面（获取HTML元素）
var html = document.documentElement;

//2. 取消HTML元素的默认事件
html.ondragover=function(e){
    e.preventDefault();
}

//3. 放下之后
html.ondrop=function(e){
    //取消默认事件
    e.preventDefault();
    console.log(e.dataTransfer.files)
    //创建图片节点
    var img = new Image();
    //设置图片url
    img.src=window.URL.createObjectURL(e.dataTransfer.files[0])

    //设置图片放置的位置（坐标）为鼠标的坐标
    img.style.top=e.pageY + 'px';
    img.style.left=e.pageX + 'px';

    //将创建的图片节点添加到页面中去
    document.body.appendChild(img)
}
```