

# HTML5程序设计基础

## 第九章 拖放



# 主要内容

---

01

DataTransfer对象

02

实现拖放的步骤

01

# DataTransfer对象

# 拖放

---

拖放是一种常见的特性，即抓取对象以后拖到另一个位置。  
在 HTML5 中，拖放（drag 和 drop）是标准的一部分，任何元素都能够拖放。

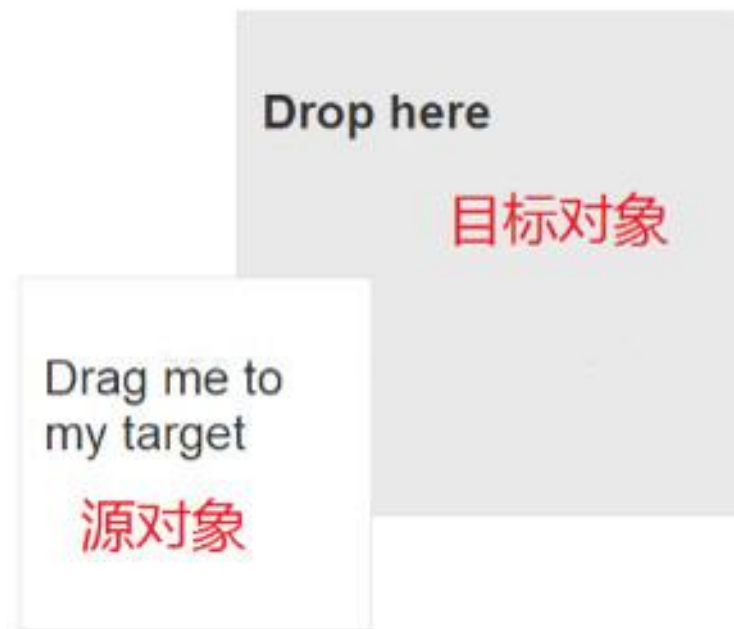
浏览器支持：

Internet Explorer 9、Firefox、Opera 12、Chrome 以及 Safari 5 支持拖放。



# 拖放

---



- **源对象**：指鼠标点击的一个事物，可以是图片，DIV，一段文本等等。
- **目标对象**：指拖动源对象后移动到一块区域，源对象可以进入这个区域，可以在这个区域上方悬停(未松手)，可以释松手释放将源对象放置此处(已松手)，也可以悬停后离开该区域。

# DataTransfer对象

---

- HTML5原生拖放
- DataTransfer , 拖拽数据传递对象。
- 为事件对象的一个属性，用于在源对象和目标对象的事件间传递字符串格式的数据。只能在拖放事件的事件处理程序中访问，可以使用此对象的属性和方法来完善拖放功能。

event.dataTransfer

# 事件对象

---

- 在触发DOM上的某个事件时，会产生一个事件对象Event。
- 事件对象中包含所有与事件有关的信息。包括导致事件的元素、事件的类型、键盘按键状态、鼠标的位置等。
- 在事件处理函数执行时，事件对象将会由浏览器自动传递给事件处理函数。
- 事件处理函数中，声明形参接收该参数。
- preventDefault( )取消事件的默认动作。

这个对象是隐藏在函数中的

# DataTransfer对象的方法

---

- setData(format, data) 向DataTransfer对象存入数据

```
ev.dataTransfer.setData("text", ev.target.innerHTML);
```

- getData(format) 从DataTransfer对象读数据

```
ev.dataTransfer.getData("text");
```

现在支持拖动处理的MIME的类型有：“text/plain（文本文字）”、“text/html”、“text/xml”、“text/uri-list（URL列表，每个URL为一行）”



# DataTransfer对象的属性

---

- types 属性 存入数据的种类。
- dropEffect 属性 表示拖放操作的视觉效果。



02

## 实现拖放的步骤

# 实现拖放的步骤

---

实现拖放操作，至少要经过两个步骤：

- （1）将需要拖放的对象元素的draggable属性设为true，即  
draggable="true"

注：img元素与a元素（必须指定href），默认允许拖放。

- （2）编写与拖放有关的事件处理代码。

# 拖放的相关事件

---

事件	产生事件的元素	描述
dragstart	被拖放的元素	开始拖放操作
drag	被拖放的元素	拖放过程中
dragenter	拖放过程中鼠标经过的元素	被拖放的元素开始进入本元素的范围内
dragover	拖放过程中鼠标经过的元素	被拖放的元素正在本元素范围内移动
dragleave	拖放过程中鼠标经过的元素	被拖放的元素离开本元素的范围
drop	拖放的目标元素	有其他元素被拖放到了本元素中
dragend	拖放的目标元素	拖放操作结束

# 拖放

---

( 1 ) 设置元素为可拖放

```

```

( 2 ) 设置拖动什么 ondragstart 和 setData()

```
<body>
  
</body>
<script type="text/javascript">
  function drag(ev){
    ev.dataTransfer.setData("Text",ev.target.id);
  }
</script>
```

拖动时触发拖动事件ondragstart，调用drag(event)。在该函数设定了被拖动元素的数据。

# 拖放

## ( 3 ) 放置元素到何处 ondragover

```
<body>
  <div ondragover="allowDrop(event)"></div>
  
</body>
<script type="text/javascript">
  function drag(ev){
    ev.dataTransfer.setData("Text",ev.target.id);
  }
  function allowDrop(ev)
  {
    ev.preventDefault();
  }
</script>
```

取消默认事件处理

默认无法将元素放置到其它元素中，如果需要设置允许放置，我们必须阻止对元素的默认处理方式。

# 拖放

## ( 4 ) 进行放置 ondrop 和 getData()

```
<body>
  <div ondrop="drop(event)" ondragover="allowDrop(event)"></div>
  
</body>
<script type="text/javascript">
  function drag(ev){
    ev.dataTransfer.setData("Text",ev.target.id);
  }
  function allowDrop(ev)
  {
    ev.preventDefault();
  }

  function drop(ev){
    ev.preventDefault();
    var data=ev.dataTransfer.getData("Text");
    ev.target.appendChild(document.getElementById(data));
  }
</script>
```

获取到要放置的元素 将其添加到当前元素的内部

示例 : 9-1.html

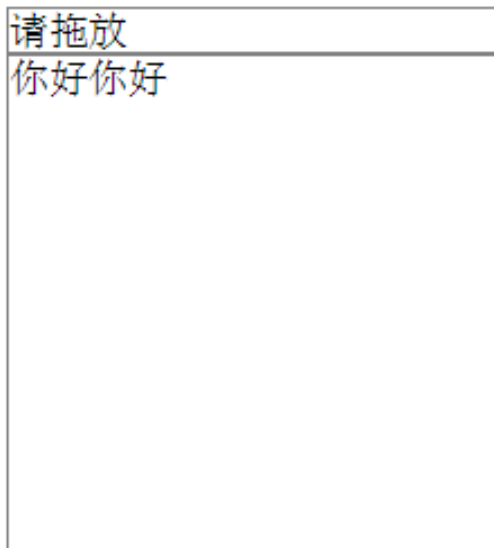
# 拖放

---

## 实例：简单拖放示例

有一个显示“拖放”文字的div元素，可以把它拖放到位于其下面的div元素中，每次被拖放时，在下面的div元素中会追加一次“你好”文字。

### 简单拖放示例



示例：9-2.html

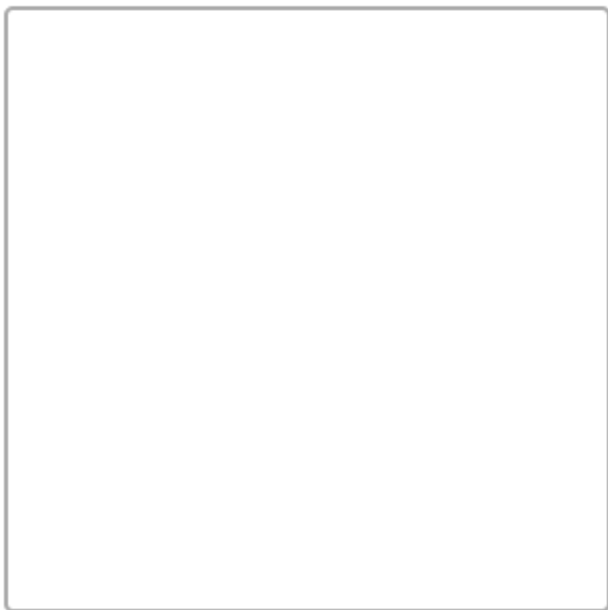


# 拖放

---

实例：拖动图片在指定区域显示

上传头像



上传头像



**THANKYOU**

