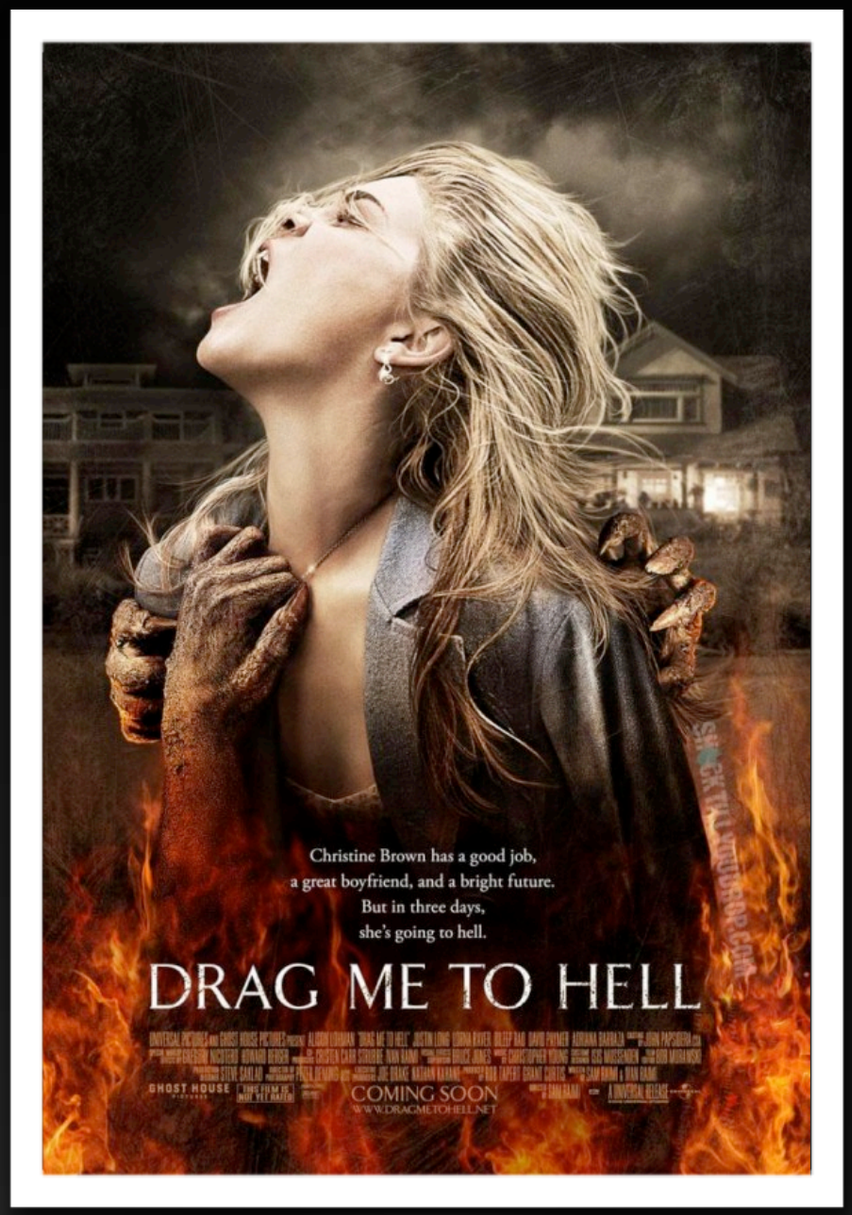
Drag and Drop



...I am forced to conclude that the HTML5 drag and drop module is not just a disaster, it's a ***fucking*** *disaster.*

-Peter-Paul Koch

```
<div id="can-be-dragged" draggable></div>
```

```
div[draggable] {
    -khtml-user-drag: element;
}
```

```
<p id="drop-area">
    Drag and drop files here
</p>
```

```
var someImg = document.getElementById("some-image"),
    dropArea = document.getElementById("drop-area");

someImg.ondragstart = function (evt) {
    var event = evt || window.event;
    event.dataTransfer.setData("Text", this.getAttribute("alt"));
    return false;
};

dropArea.ondragenter = function (evt) {
    return false;
};

dropArea.ondragover = function (evt) {
    return false;
};

dropArea.ondrop = function (evt) {
    var text = event.dataTransfer.getData("Text");
    event.cancelBubble = true; // For IE
    return false;
};
```

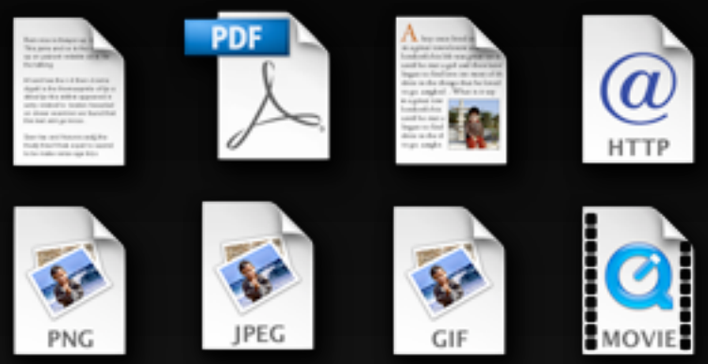
```
// Needed for web browser compatibility
dropArea.addEventListener("dragleave", function (evt) {
    evt.preventDefault();
    evt.stopPropagation();
}, false);

// Needed for web browser compatibility
dropArea.addEventListener("dragenter", function (evt) {
    evt.preventDefault();
    evt.stopPropagation();
}, false);

// Needed for web browser compatibility
dropArea.addEventListener("dragover", function (evt) {
    evt.preventDefault();
    evt.stopPropagation();
}, false);

dropArea.addEventListener("drop", function (evt) {
    // Get reference to dropped files
    var files = evt.dataTransfer.files;
    evt.preventDefault();
    evt.stopPropagation();
}, false);
```

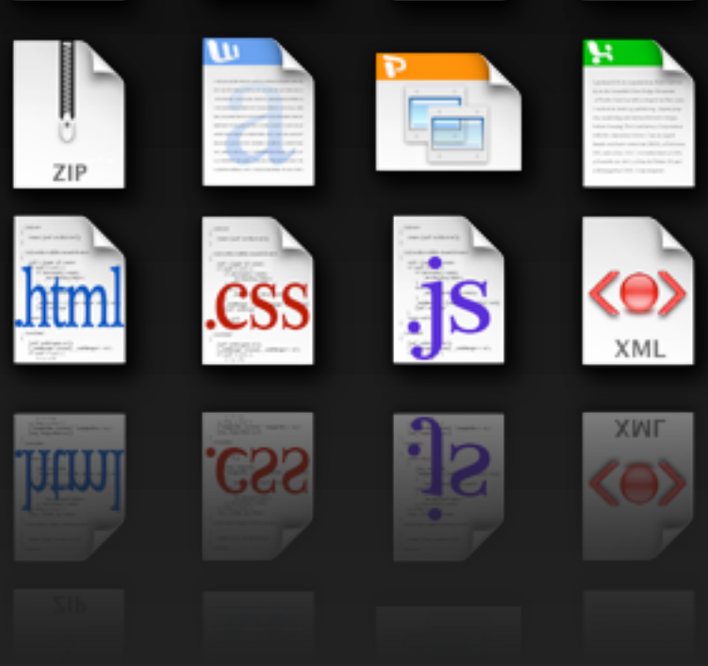
```
// Needed for web browser compatibility
dropArea.addEventListener("dragleave", function (evt) {
    evt.preventDefault();
    evt.stopPropagation();
}, false);

// Needed for web browser compatibility
dropArea.addEventListener("dragenter", function (evt) {
    evt.preventDefault();
    evt.stopPropagation();
}, false);

// Needed for web browser compatibility
dropArea.addEventListener("dragover", function (evt) {
    evt.preventDefault();
    evt.stopPropagation();
}, false);

dropArea.addEventListener("drop", function (evt) {
    // Get reference to dropped files
    var files = evt.dataTransfer.files;
    evt.preventDefault();
    evt.stopPropagation();
}, false);
```

> “If the drop is to be accepted, then this event (dragover) has to be canceled.”

```
someImg.ondragstart = function (evt) {
    var event = evt || window.event;
    event.dataTransfer.setDragImage(dragIcon, -10, -10);
    return false;
};
```

File API





```
<!--
    The multiple attribute allows for
    uploading of multiple files
-->
<input id="files-upload" type="file" multiple>
```

```
var filesUpload = document.getElementById("files-upload");
filesUpload.onchange = function () {
    // Access to data about all files
    var files = this.files;
    for (var i=0, il=files.length; i<il; i++) {
        file.name; // Get the name of the file
        file.size; // Get the size of the file, in bytes
        file.type; // Get the type of the file
    };
};
```

Uploaded files

Name: berlin.jpg

Size: 8631 bytes

Type: image/jpeg

Name: cv.rtf

Size: 1309 bytes

Type: text/rtf

Name: Screen Recording.mov

Size: 3108303 bytes

Type: video/quicktime

```
for (var i=0, il=files.length, file, img; i<il; i++) {
    file = files[i];

    if (typeof FileReader !== "undefined") {
        img = document.createElement("img");
        reader = new FileReader();
        reader.onload = (function (theImg) {
            return function (evt) {
                theImg.src = evt.target.result;
            };
        }(img));
        reader.readAsDataURL(file);
    }
}
```

```
// For Firefox, Chrome and Safari
var xhr = new XMLHttpRequest();
xhr.open("post", "upload/upload.php", true);
xhr.onreadystatechange = function() {
  if (this.readyState === 4) {
        // File uploaded
    }
};

// Upload file: Firefox, Google Chrome and Safari
xhr.setRequestHeader("Content-Type", "multipart/form-data");
xhr.setRequestHeader("X-File-Name", file.fileName);
xhr.setRequestHeader("X-File-Size", file.fileSize);
xhr.setRequestHeader("X-File-Type", file.type);

xhr.send(file);
```

About Plupload

The developers of TinyMCE brings you Plupload, a highly usable upload handler for your Content Management Systems or similar. Plupload is currently separated into a Core API and a jQuery upload queue widget this enables you to either use it out of the box or write your own custom implementation.

Features

This table shows the availability of the different features in each available runtime.

Feature	Flash	Gears	HTML 5	Silverlight	BrowserPlus	HTML 4
Chunking	✓	✓	✗	✓	✓	✗
Drag/Drop	✗	✓	✓ [1]	✗	✓	✗
PNG Resize	✓	✓	✓ [2]	✓	✓	✗
JPEG Resize	✓	✓	✓ [2]	✓	✓	✗
Type filtering	✓	✓	✗ [3]	✓	✓	✗
Stream upload	✓	✓	✓	✓	✗	✗
Multipart upload	✓	✓	✓ [4]	✓	✓	✓
File size restriction	✓	✓	✓	✓	✓	✗
Upload progress	✓	✓	✓	✓	✓	✗