HTML5 Drag and Drop

[❮ Previous](http://www.w3schools.com/html/html5_geolocation.asp)[Next ❯](http://www.w3schools.com/html/html5_webstorage.asp)

W3Schools

Drag the W3Schools image into the rectangle.

Drag and Drop

Drag and drop is a very common feature. It is when you "grab" an object and drag it to a different location.

In HTML5, drag and drop is part of the standard: Any element can be draggable.

Browser Support

The numbers in the table specify the first browser version that fully supports Drag and Drop.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| API |  |  |  |  |  |
| Drag and Drop | 4.0 | 9.0 | 3.5 | 6.0 | 12.0 |

HTML Drag and Drop Example

The example below is a simple drag and drop example:

Example

<!DOCTYPE HTML>  
<html>  
<head>  
<script>  
function allowDrop(ev) {  
    ev.preventDefault();  
}  
  
function drag(ev) {  
    ev.dataTransfer.setData("text", ev.target.id);  
}  
  
function drop(ev) {  
    ev.preventDefault();  
    var data = ev.dataTransfer.getData("text");  
    ev.target.appendChild(document.getElementById(data));  
}  
</script>  
</head>  
<body>  
  
<div id="div1" ondrop="drop(event)" ondragover="allowDrop(event)"></div>  
  
<img id="drag1" src="img\_logo.gif" draggable="true"  
ondragstart="drag(event)" width="336" height="69">  
  
</body>  
</html>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml5_draganddrop)

It might seem complicated, but lets go through all the different parts of a drag and drop event.

Make an Element Draggable

First of all: To make an element draggable, set the draggable attribute to true:

<img draggable="true">

What to Drag - ondragstart and setData()

Then, specify what should happen when the element is dragged.

In the example above, the ondragstart attribute calls a function, drag(event), that specifies what data to be dragged.

The dataTransfer.setData() method sets the data type and the value of the dragged data:

function drag(ev) {  
    ev.dataTransfer.setData("text", ev.target.id);  
}

In this case, the data type is "text" and the value is the id of the draggable element ("drag1").

Where to Drop - ondragover

The ondragover event specifies where the dragged data can be dropped.

By default, data/elements cannot be dropped in other elements. To allow a drop, we must prevent the default handling of the element.

This is done by calling the event.preventDefault() method for the ondragover event:

*event*.preventDefault()

Do the Drop - ondrop

When the dragged data is dropped, a drop event occurs.

In the example above, the ondrop attribute calls a function, drop(event):

function drop(ev) {  
    ev.preventDefault();  
    var data = ev.dataTransfer.getData("text");  
    ev.target.appendChild(document.getElementById(data));  
}

Code explained:

* Call preventDefault() to prevent the browser default handling of the data (default is open as link on drop)
* Get the dragged data with the dataTransfer.getData() method. This method will return any data that was set to the same type in the setData() method
* The dragged data is the id of the dragged element ("drag1")
* Append the dragged element into the drop element

Arrastre HTML y soltar las interfaces permiten a las aplicaciones utilizar arrastrar y soltar características en Firefox y otros navegadores. Por ejemplo, con estas características, el usuario puede seleccionar *que se pueden arrastrar* elementos con un ratón, arrastre los elementos a un*lanzables* elemento, y eliminar estos elementos mediante la liberación el botón del ratón. Una representación transparente de los *que se pueden arrastrar* elementos sigue el puntero del ratón durante la operación de arrastre.

Para los sitios web, extensiones y aplicaciones XUL, puede personalizar los tipos de elementos que pueden llegar a ser *arrastrables* elementos, y el tipo de retroalimentación los *que se pueden arrastrar* elementos producen, y los *lanzables* elementos.

Este documento es un *resumen* de arrastre HTML y gota. Incluye una descripción de las interfaces, los pasos básicos para añadir arrastrar y soltar el apoyo a una aplicación y un resumen de la interoperabilidad de las interfaces.

**Arrastre y Eventos**[EDITAR](https://developer.mozilla.org/en-US/docs/Web/API/HTML_Drag_and_Drop_API$edit#Drag_Events)

Arrastrar y soltar HTML utiliza el [DOM event model](https://developer.mozilla.org/en-US/docs/Web/API/Event)y *[drag events](https://developer.mozilla.org/en-US/docs/Web/API/DragEvent" \o "La interfaz DragEvent es un evento DOM que representa un arrastrar y soltar interacción.  El usuario inicia una resistencia, colocando un dispositivo de señalización (por ejemplo, un ratón) sobre la superficie táctil y luego arrastrando el puntero a una nueva ubicación (tal como otro elemento DOM).  Las aplicaciones son libres de interpretar un arrastrar y soltar interacción de una forma específica de la aplicación.)*heredan de [mouse events](https://developer.mozilla.org/en-US/docs/Web/API/MouseEvent). Una operación típica de arrastre empieza cuando un usuario selecciona un *arrastrable* elemento con un ratón, mueve el ratón a una *lanzables* elemento y luego suelta el ratón. Durante las operaciones, varios tipos de eventos son disparados y algunos tipos de eventos pueden ser despedidos muchas veces (por ejemplo, el [drag](https://developer.mozilla.org/en-US/docs/Web/Events/drag" \o "/en-US/docs/Web/Events/drag)y [dragover](https://developer.mozilla.org/en-US/docs/Web/Events/dragover" \o "/en-US/docs/Web/Events/dragover)los tipos de eventos).

Todos los [tipos de eventos de arrastrar](https://developer.mozilla.org/en-US/docs/Web/API/DragEvent#Event_types) tienen una asociada [controlador de eventos globales](https://developer.mozilla.org/en-US/docs/Web/API/DragEvent#GlobalEventHandlers) .Cada tipo de evento de arrastre y arrastre atributo global tiene un documento de referencia que describe el evento. La siguiente tabla ofrece una breve descripción de los tipos de eventos y un enlace a los documentos de referencia.

|  |  |  |
| --- | --- | --- |
| **Evento** | **El Gestor de Eventos** | **Descripción** |
| [drag](https://developer.mozilla.org/en-US/docs/Web/Events/drag) | [ondrag](https://developer.mozilla.org/en-US/docs/Web/API/GlobalEventHandlers/ondrag) | Se activa cuando se está arrastrando un elemento o selección de texto. |
| [dragend](https://developer.mozilla.org/en-US/docs/Web/Events/dragend) | [ondragend](https://developer.mozilla.org/en-US/docs/Web/API/GlobalEventHandlers/ondragend) | Se activa cuando se puso fin a una operación de arrastre (por ejemplo, al soltar el botón del ratón o pulsando la tecla de escape). (Ver [El acabado de un Drag](https://developer.mozilla.org/en-US/docs/DragDrop/Drag_Operations#dragend) ). |
| [dragenter](https://developer.mozilla.org/en-US/docs/Web/Events/dragenter) | [ondragenter](https://developer.mozilla.org/en-US/docs/Web/API/GlobalEventHandlers/ondragenter) | Se activa cuando un elemento arrastrado o selección de texto entra en un destino de colocación válida. (Ver [Objetivos Especificación de caída](https://developer.mozilla.org/en-US/docs/DragDrop/Drag_Operations" \l "droptargets" \o "Especificación de destinos de colocación) ). |
| [dragexit](https://developer.mozilla.org/en-US/docs/Web/Events/dragexit) | [ondragexit](https://developer.mozilla.org/en-US/docs/Web/API/GlobalEventHandlers/ondragexit) | Se activa cuando un elemento ya no es objetivo de la selección inmediata de la operación de arrastre. |
| [dragleave](https://developer.mozilla.org/en-US/docs/Web/Events/dragleave) | [ondragleave](https://developer.mozilla.org/en-US/docs/Web/API/GlobalEventHandlers/ondragleave) | Se activa cuando un elemento arrastrado o selección de texto deja un destino de colocación válida. |
| [dragover](https://developer.mozilla.org/en-US/docs/Web/Events/dragover) | [ondragover](https://developer.mozilla.org/en-US/docs/Web/API/GlobalEventHandlers/ondragover) | Se activa cuando un elemento o selección de texto está siendo arrastrado sobre un objetivo de colocación válido (cada pocos cientos de milisegundos). |
| [dragstart](https://developer.mozilla.org/en-US/docs/Web/Events/dragstart) | [ondragstart](https://developer.mozilla.org/en-US/docs/Web/API/GlobalEventHandlers/ondragstart) | Se activa cuando el usuario empieza a arrastrar una selección de elementos o texto. (Consulte [Inicio de una operación de arrastre](https://developer.mozilla.org/en-US/docs/DragDrop/Drag_Operations" \l "dragstart" \o "Inicio de una operación de arrastre) .) |
| [drop](https://developer.mozilla.org/en-US/docs/Web/Events/drop) | [ondrop](https://developer.mozilla.org/en-US/docs/Web/API/GlobalEventHandlers/ondrop) | Se activa cuando un elemento de selección o el texto se deja caer en un destino de colocación válida. (Véase [Realización de una gota](https://developer.mozilla.org/en-US/docs/DragDrop/Drag_Operations" \l "drop" \o "Realización de una gota) ). |

Tenga en cuenta que dragstarty dragendacontecimientos no se disparan cuando se arrastra un archivo en el navegador del sistema operativo.

**Interfaces**[EDITAR](https://developer.mozilla.org/en-US/docs/Web/API/HTML_Drag_and_Drop_API$edit#Interfaces)

Las interfaces de arrastrar y soltar HTML son [DragEvent](https://developer.mozilla.org/en-US/docs/Web/API/DragEvent), [DataTransfer](https://developer.mozilla.org/en-US/docs/Web/API/DataTransfer), [DataTransferItem](https://developer.mozilla.org/en-US/docs/Web/API/DataTransferItem)y[DataTransferItemList](https://developer.mozilla.org/en-US/docs/Web/API/DataTransferItemList).

La [DragEvent](https://developer.mozilla.org/en-US/docs/Web/API/DragEvent" \o "La interfaz DragEvent es un evento DOM que representa un arrastrar y soltar interacción.  El usuario inicia una resistencia, colocando un dispositivo de señalización (por ejemplo, un ratón) sobre la superficie táctil y luego arrastrando el puntero a una nueva ubicación (tal como otro elemento DOM).  Las aplicaciones son libres de interpretar un arrastrar y soltar interacción de una forma específica de la aplicación.)interfaz tiene un constructor y una propiedad, la [dataTransfer](https://developer.mozilla.org/en-US/docs/Web/API/DragEvent/dataTransfer" \o "revisión editorial completó.)propiedad, que es un [DataTransfer](https://developer.mozilla.org/en-US/docs/Web/API/DataTransfer" \o "El objeto dataTransfer se utiliza para contener los datos que se está arrastrando durante una operación de arrastrar y soltar.  Puede contener uno o más elementos de datos, cada uno de uno o más tipos de datos.  Para obtener más información acerca de arrastrar y soltar, consulte Arrastre HTML y API gota.)objeto. [DataTransfer](https://developer.mozilla.org/en-US/docs/Web/API/DataTransfer" \o "El objeto dataTransfer se utiliza para contener los datos que se está arrastrando durante una operación de arrastrar y soltar.  Puede contener uno o más elementos de datos, cada uno de uno o más tipos de datos.  Para obtener más información acerca de arrastrar y soltar, consulte Arrastre HTML y API gota.)Objetos incluyen el estado del evento de arrastre, tales como el tipo de arrastre está haciendo (por ejemplo, copyo move), los datos del arrastre (uno o más elementos) y el tipo de cada *elemento de arrastre* (un tipo MIME). [DataTransfer](https://developer.mozilla.org/en-US/docs/Web/API/DataTransfer" \o "El objeto dataTransfer se utiliza para contener los datos que se está arrastrando durante una operación de arrastrar y soltar.  Puede contener uno o más elementos de datos, cada uno de uno o más tipos de datos.  Para obtener más información acerca de arrastrar y soltar, consulte Arrastre HTML y API gota.)objetos también tienen métodos para agregar elementos a los datos del arrastre y para eliminar un elemento de arrastre. El [DragEvent](https://developer.mozilla.org/en-US/docs/Web/API/DragEvent" \o "La interfaz DragEvent es un evento DOM que representa un arrastrar y soltar interacción.  El usuario inicia una resistencia, colocando un dispositivo de señalización (por ejemplo, un ratón) sobre la superficie táctil y luego arrastrando el puntero a una nueva ubicación (tal como otro elemento DOM).  Las aplicaciones son libres de interpretar un arrastrar y soltar interacción de una forma específica de la aplicación.)e [DataTransfer](https://developer.mozilla.org/en-US/docs/Web/API/DataTransfer" \o "El objeto dataTransfer se utiliza para contener los datos que se está arrastrando durante una operación de arrastrar y soltar.  Puede contener uno o más elementos de datos, cada uno de uno o más tipos de datos.  Para obtener más información acerca de arrastrar y soltar, consulte Arrastre HTML y API gota.)interfaces deben ser los únicos necesarios para agregar arrastrar y soltar HTML capacidades para una aplicación. Sin embargo, tenga en cuenta Firefox soporta algunas [extensiones Gecko-específicos](https://developer.mozilla.org/en-US/docs/Web/API/HTML_Drag_and_Drop_API#Gecko_specific_interfaces) al [DataTransfer](https://developer.mozilla.org/en-US/docs/Web/API/DataTransfer" \o "El objeto dataTransfer se utiliza para contener los datos que se está arrastrando durante una operación de arrastrar y soltar.  Puede contener uno o más elementos de datos, cada uno de uno o más tipos de datos.  Para obtener más información acerca de arrastrar y soltar, consulte Arrastre HTML y API gota.)objeto que se puede utilizar, aunque esas extensiones sólo funcionarán en Firefox.

Cada [DataTransfer](https://developer.mozilla.org/en-US/docs/Web/API/DataTransfer" \o "El objeto dataTransfer se utiliza para contener los datos que se está arrastrando durante una operación de arrastrar y soltar.  Puede contener uno o más elementos de datos, cada uno de uno o más tipos de datos.  Para obtener más información acerca de arrastrar y soltar, consulte Arrastre HTML y API gota.)objeto contiene una [items](https://developer.mozilla.org/en-US/docs/Web/API/DataTransfer/items" \o "La propiedad DataTransfer.items es una lista de los elementos de transferencia de datos en una operación de arrastre.  La lista incluye un elemento para cada elemento de la operación y si la operación no tenía elementos, la lista está vacía.)propiedad que es una [list](https://developer.mozilla.org/en-US/docs/Web/API/DataTransferItemList" \o "El objeto DataTransferItemList es una lista de objetos que representan elementos DataTransferItem ser arrastrado.  Durante una operación de arrastre, cada uno tiene una propiedad DragEvent dataTransfer y que la propiedad es un DataTransferItemList.)de[DataTransferItem](https://developer.mozilla.org/en-US/docs/Web/API/DataTransferItem)objetos. Cada [DataTransferItem](https://developer.mozilla.org/en-US/docs/Web/API/DataTransferItem" \o "El objeto DataTransferItem representa un elemento de datos de arrastre.  Durante una operación de arrastre, arrastre cada evento tiene una propiedad dataTransfer que contiene una lista de elementos de datos de arrastre.  Cada elemento de la lista es un objeto DataTransferItem.)objeto representa un solo *elemento de arrastre* y cada artículo tiene una [kind](https://developer.mozilla.org/en-US/docs/Web/API/DataTransferItem/kind" \o "El sólo lectura DataTransferItem.kind propiedad devuelve un DataTransferItem que representa el tipo de arrastre elemento de datos: un poco de texto o algún archivo.)propiedad que es el *tipo* de datos (ya sea stringo file) y una [type](https://developer.mozilla.org/en-US/docs/Web/API/DataTransferItem/type" \o "El sólo lectura DataTransferItem.type propiedad devuelve el tipo (formato) del objeto DataTransferItem que representa el elemento de datos de arrastre.  El tipo es una cadena Unicode generalmente dado por un tipo MIME, aunque no se requiere un tipo MIME.)propiedad que es el tipo de elemento de datos (es decir, tipo MIME). El[DataTransferItem](https://developer.mozilla.org/en-US/docs/Web/API/DataTransferItem)objeto también tiene métodos para obtener los datos del elemento de arrastre.

El [DataTransferItemList](https://developer.mozilla.org/en-US/docs/Web/API/DataTransferItemList" \o "El objeto DataTransferItemList es una lista de objetos que representan elementos DataTransferItem ser arrastrado.  Durante una operación de arrastre, cada uno tiene una propiedad DragEvent dataTransfer y que la propiedad es un DataTransferItemList.)objeto es una lista de [DataTransferItem](https://developer.mozilla.org/en-US/docs/Web/API/DataTransferItem" \o "El objeto DataTransferItem representa un elemento de datos de arrastre.  Durante una operación de arrastre, arrastre cada evento tiene una propiedad dataTransfer que contiene una lista de elementos de datos de arrastre.  Cada elemento de la lista es un objeto DataTransferItem.)objetos. El objeto de la lista tiene métodos para: añadir un elemento de arrastre a la lista, retirar un elemento de arrastre de la lista y borrar la lista de todos los elementos de arrastre.

Una diferencia clave entre los [DataTransfer](https://developer.mozilla.org/en-US/docs/Web/API/DataTransfer" \o "El objeto dataTransfer se utiliza para contener los datos que se está arrastrando durante una operación de arrastrar y soltar.  Puede contener uno o más elementos de datos, cada uno de uno o más tipos de datos.  Para obtener más información acerca de arrastrar y soltar, consulte Arrastre HTML y API gota.)y [DataTransferItem](https://developer.mozilla.org/en-US/docs/Web/API/DataTransferItem" \o "El objeto DataTransferItem representa un elemento de datos de arrastre.  Durante una operación de arrastre, arrastre cada evento tiene una propiedad dataTransfer que contiene una lista de elementos de datos de arrastre.  Cada elemento de la lista es un objeto DataTransferItem.)las interfaces es que el primero utiliza la síncrono [getData()](https://developer.mozilla.org/en-US/docs/Web/API/DataTransfer/getData" \o "El método dataTransfer.getData () recupera datos de arrastre (como DOMString) para el tipo especificado.  Si la operación de arrastre no incluye datos, este método devuelve una cadena vacía.)método para acceder a los datos de un elemento de arrastre, mientras que la tarde utiliza el asíncrona [getAsString()](https://developer.mozilla.org/en-US/docs/Web/API/DataTransferItem/getAsString" \o "El método DataTransferItem.getAsString () invoca la devolución de llamada dada con los datos de cadena de arrastre del elemento de datos como el argumento si la especie de que el artículo es una cadena Unicode sin formato (es decir, tipo es string).)método para acceder a los datos de un elemento de arrastre.

Nota: el [DragEvent](https://developer.mozilla.org/en-US/docs/Web/API/DragEvent" \o "La interfaz DragEvent es un evento DOM que representa un arrastrar y soltar interacción.  El usuario inicia una resistencia, colocando un dispositivo de señalización (por ejemplo, un ratón) sobre la superficie táctil y luego arrastrando el puntero a una nueva ubicación (tal como otro elemento DOM).  Las aplicaciones son libres de interpretar un arrastrar y soltar interacción de una forma específica de la aplicación.)e [DataTransfer](https://developer.mozilla.org/en-US/docs/Web/API/DataTransfer" \o "El objeto dataTransfer se utiliza para contener los datos que se está arrastrando durante una operación de arrastrar y soltar.  Puede contener uno o más elementos de datos, cada uno de uno o más tipos de datos.  Para obtener más información acerca de arrastrar y soltar, consulte Arrastre HTML y API gota.)interfaces son ampliamente interoperable con los navegadores de escritorio. Sin embargo, los [DataTransferItem](https://developer.mozilla.org/en-US/docs/Web/API/DataTransferItem" \o "El objeto DataTransferItem representa un elemento de datos de arrastre.  Durante una operación de arrastre, arrastre cada evento tiene una propiedad dataTransfer que contiene una lista de elementos de datos de arrastre.  Cada elemento de la lista es un objeto DataTransferItem.)y [DataTransferItemList](https://developer.mozilla.org/en-US/docs/Web/API/DataTransferItemList" \o "El objeto DataTransferItemList es una lista de objetos que representan elementos DataTransferItem ser arrastrado.  Durante una operación de arrastre, cada uno tiene una propiedad DragEvent dataTransfer y que la propiedad es un DataTransferItemList.)las interfaces han limitado el apoyo del navegador. Ver [Interoperabilidad](https://developer.mozilla.org/en-US/docs/Web/API/HTML_Drag_and_Drop_API#Interoperability) para obtener más información acerca de la fricción y la interoperabilidad gota.

interfaces de Gecko-específicos

Mozilla Firefox y apoyan algunas de las características no en el modelo estándar de arrastrar y soltar. Estas son *funciones de conveniencia* para facilitar arrastrando varios artículos y arrastrando los datos no son cadenas (como archivos). Para obtener más información, consulte[Arrastrar y soltar los diversos elementos](https://developer.mozilla.org/en-US/docs/DragDrop/Dragging_and_Dropping_Multiple_Items) . Además, consulte la [DataTransfer](https://developer.mozilla.org/en-US/docs/Web/API/DataTransfer" \o "El objeto dataTransfer se utiliza para contener los datos que se está arrastrando durante una operación de arrastrar y soltar.  Puede contener uno o más elementos de datos, cada uno de uno o más tipos de datos.  Para obtener más información acerca de arrastrar y soltar, consulte Arrastre HTML y API gota.)página de referencia para todas las [propiedades Gecko-específicos](https://developer.mozilla.org/en-US/docs/Web/API/DataTransfer#Gecko_properties) y [métodos Gecko-específicas](https://developer.mozilla.org/en-US/docs/Web/API/DataTransfer#Gecko_methods) .

**Los basicos**[EDITAR](https://developer.mozilla.org/en-US/docs/Web/API/HTML_Drag_and_Drop_API$edit#The_basics)

En esta sección se ofrece un resumen de los pasos básicos para agregar funcionalidad de arrastrar y soltar a una aplicación. Cada sección incluye una descripción de la etapa, un ejemplo de código corto, y enlaces a información adicional.

Identificar lo que es *posible arrastrar*

Para hacer que un elemento *arrastrable* requiere añadir el [draggable](https://developer.mozilla.org/en-US/docs/Web/HTML/Global_attributes" \l "attr-draggable)atributo más el[ondragstart](https://developer.mozilla.org/en-US/docs/Web/API/GlobalEventHandlers/ondragstart)gestor de eventos global, como se muestra en el siguiente ejemplo de código

function dragstart\_handler(ev) {

console.log("dragStart");

// Add the target element's id to the data transfer object

ev.dataTransfer.setData("text/plain", ev.target.id);

}

<body>

<p id="p1" draggable="true" ondragstart="dragstart\_handler(event);">This element is draggable.</p>

</body>

Ver la [referencia de atributos que pueden arrastrarse](https://developer.mozilla.org/Web/HTML/Global_attributes/draggable) y la [guía de operaciones de arrastrar](https://developer.mozilla.org/Web/Guide/HTML/Drag_operations#draggableattribute) para más información.

Definir los datos del arrastre

La aplicación es gratuita para incluir cualquier número de elementos de datos en una operación de arrastre. Cada elemento de datos es una [string](https://developer.mozilla.org/en-US/docs/Web/API/DOMString" \o "DOMString es una cadena UTF-16.  Como ya se utiliza JavaScript dichas cadenas, DOMString se asigna directamente a una cadena.)de un determinado typetípicamente un tipo, MIME tales como text/html.

Cada uno [drag event](https://developer.mozilla.org/en-US/docs/Web/API/DragEvent" \o "La interfaz DragEvent es un evento DOM que representa un arrastrar y soltar interacción.  El usuario inicia una resistencia, colocando un dispositivo de señalización (por ejemplo, un ratón) sobre la superficie táctil y luego arrastrando el puntero a una nueva ubicación (tal como otro elemento DOM).  Las aplicaciones son libres de interpretar un arrastrar y soltar interacción de una forma específica de la aplicación.)tiene una [dataTransfer](https://developer.mozilla.org/en-US/docs/Web/API/DragEvent/dataTransfer" \o "revisión editorial completó.)propiedad que *contiene* los datos del evento. Esta propiedad (que es un [DataTransfer](https://developer.mozilla.org/en-US/docs/Web/API/DataTransfer" \o "El objeto dataTransfer se utiliza para contener los datos que se está arrastrando durante una operación de arrastrar y soltar.  Puede contener uno o más elementos de datos, cada uno de uno o más tipos de datos.  Para obtener más información acerca de arrastrar y soltar, consulte Arrastre HTML y API gota.)objeto) también tiene métodos para *gestionar los* datos de arrastre. El [setData()](https://developer.mozilla.org/en-US/docs/Web/API/DataTransfer/setData" \o "El método dataTransfer.setData () establece datos de arrastre de la operación de arrastrar a los datos y tipo especificados.  Si no se dispone de datos para el tipo determinado, se añade al final del almacén de datos de la fricción, de tal manera que el último elemento de la lista de tipos será el nuevo tipo.  Si los datos para el tipo dado ya existe, los datos existentes serán reemplazados en la misma posición.  Es decir, el orden de la lista de tipos no se cambia la hora de sustituir los datos del mismo tipo.)método se utiliza para agregar un elemento a los datos de arrastre, como se muestra en el siguiente ejemplo.

function dragstart\_handler(ev) {

// Add the drag data

ev.dataTransfer.setData("text/plain", ev.target.id);

ev.dataTransfer.setData("text/html", "<p>Example paragraph</p>");

ev.dataTransfer.setData("text/uri-list", "http://developer.mozilla.org");

}

Para obtener una lista de tipos de datos comunes utilizados para arrastrar y soltar (como texto, HTML, enlaces y archivos), véase [Tipos Arrastre](https://developer.mozilla.org/en-US/docs/DragDrop/Recommended_Drag_Types) y para obtener más información sobre datos de arrastre, ver [datos de arrastre](https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Drag_operations#dragdata) .

Definir la imagen de arrastre

Por defecto, el navegador proporciona una imagen que aparece junto al puntero del ratón durante una operación de arrastre. Sin embargo, una aplicación puede definir una imagen personalizada mediante el [setDragImage()](https://developer.mozilla.org/en-US/docs/Web/API/DataTransfer/setDragImage" \o "Cuando se produce un lastre, una imagen translúcida se genera a partir del destino de arrastre (el elemento del evento dragstart se cuece a), y sigue el puntero del ratón durante el arrastre.  Esta imagen se crea de forma automática, por lo que no es necesario para crear usted mismo.  Sin embargo, si se desea una imagen personalizada, el método DataTransfer.setDragImage () puede ser usado para fijar la imagen personalizada para ser utilizado.)método como se muestra en el siguiente ejemplo.

function dragstart\_handler(ev) {

// Create an image and then use it for the drag image.

// NOTE: change "example.gif" to an existing image or the image

// will not be created and the default drag image will be used.

var img = new Image();

img.src = 'example.gif';

ev.dataTransfer.setDragImage(img, 10, 10);

}

Para obtener más información acerca de las imágenes de retroalimentación de arrastre, consulte [Ajuste de la fricción de votos imagen](https://developer.mozilla.org/en-US/docs/DragDrop/Drag_Operations#dragfeedback) .

Definir el arrastre *efecto*

La [dropEffect](https://developer.mozilla.org/en-US/docs/Web/API/DataTransfer/dropEffect" \o "La propiedad dataTransfer.dropEffect controla la retroalimentación (típicamente visual) se le da al usuario durante una operación de arrastrar y soltar.  Que afectará a la que el cursor se visualiza mientras se arrastra.  Por ejemplo, cuando el usuario se desplaza sobre un elemento de caída de destino, el cursor del navegador puede indicar que se producirá tipo de operación.)propiedad se utiliza para controlar la retroalimentación (típicamente visual) se le da al usuario durante una operación de arrastrar y soltar. Afecta el cursor que las pantallas del navegador mientras se arrastra. Por ejemplo, cuando el usuario se desplaza sobre un elemento de caída de destino, el cursor del navegador puede indicar el tipo de operación que va a producir.

Tres efectos pueden ser definidos:

copy indica que los datos que se arrastran se copiarán desde su ubicación actual a la ubicación de destino.

move indica que los datos que se arrastran se moverán

link indica que se creará algún tipo de relación o conexión entre los lugares de origen y de la gota.

Durante la operación de arrastre, los efectos de arrastre pueden ser modificados para indicar que ciertos efectos se les permite en ciertos lugares. Si se permite, una gota se puede producir en ese lugar.

El siguiente ejemplo muestra cómo se utiliza esta propiedad.

function dragstart\_handler(ev) {

// Set the drag effect to copy

ev.dataTransfer.dropEffect = "copy";

}

Ver [Efectos Arrastre](https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Drag_operations#drageffects) para más detalles.

Definir una *zona de arrastre*

Por defecto, el navegador evita que pase algo cuando dejar caer algo sobre el elemento HTML.Para cambiar esto para que un elemento se convierte en una *zona de arrastre* o es *lanzables* , el elemento debe tener tanto [ondragover](https://developer.mozilla.org/en-US/docs/Web/API/GlobalEventHandlers/ondragover" \o "Un controlador de eventos para el evento mundial dragover.)y  [ondrop](https://developer.mozilla.org/en-US/docs/Web/API/GlobalEventHandlers/ondrop" \o "revisión editorial completó.)atributos de controlador de eventos. El siguiente ejemplo muestra cómo utilizar esos atributos e incluye controladores de eventos básicos para cada atributo.

function dragover\_handler(ev) {

ev.preventDefault();

// Set the dropEffect to move

ev.dataTransfer.dropEffect = "move"

}

function drop\_handler(ev) {

ev.preventDefault();

// Get the id of the target and add the moved element to the target's DOM

var data = ev.dataTransfer.getData("text");

ev.target.appendChild(document.getElementById(data));

}

<body>

<div id="target" ondrop="drop\_handler(event);" ondragover="dragover\_handler(event);">Drop Zone</div>

</body>

Nota cada controlador de llama [preventDefault()](https://developer.mozilla.org/en-US/docs/Web/API/Event/preventDefault" \o "Cancela el evento si es cancelable, sin detener el resto del evento.)para evitar el procesamiento de eventos adicionales para este prevenir (como eventos de toque o eventos de puntero).

Para obtener más información, consulte [Objetivos Especificación de la gota](https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Drag_operations#droptargets) .

Manejar la caída de *efecto*

El controlador para el [drop](https://developer.mozilla.org/en-US/docs/Web/Events/drop" \o "/en-US/docs/Web/Events/drop)evento es libre para procesar los datos de arrastre en una forma específica de la aplicación. Por lo general, una aplicación utilizará el [getData()](https://developer.mozilla.org/en-US/docs/Web/API/DataTransfer/getData" \o "El método dataTransfer.getData () recupera datos de arrastre (como DOMString) para el tipo especificado.  Si la operación de arrastre no incluye datos, este método devuelve una cadena vacía.)método para recuperar los elementos de arrastre y procesarlos en consecuencia. Además, la semántica de aplicación pueden variar en función del valor de la [dropEffect](https://developer.mozilla.org/en-US/docs/Web/API/DataTransfer/dropEffect" \o "La propiedad dataTransfer.dropEffect controla la retroalimentación (típicamente visual) se le da al usuario durante una operación de arrastrar y soltar.  Que afectará a la que el cursor se visualiza mientras se arrastra.  Por ejemplo, cuando el usuario se desplaza sobre un elemento de caída de destino, el cursor del navegador puede indicar que se producirá tipo de operación.)y / o el estado de las teclas modificadoras.

El siguiente ejemplo muestra un controlador gota conseguir id del elemento de origen de los datos de arrastre y luego utilizando el ID para mover el elemento de origen al elemento gota.

function dragstart\_handler(ev) {

ev.preventDefault();

// Add the target element's id to the data transfer object

ev.dataTransfer.setData("text/plain", ev.target.id);

ev.dropEffect = "move";

}

function drop\_handler(ev) {

ev.preventDefault();

// Get the id of the target and add the moved element to the target's DOM

var data = ev.dataTransfer.getData("text");

ev.target.appendChild(document.getElementById(data));

}

<body>

<p id="p1" draggable="true" ondragstart="dragstart\_handler(event);">This element is draggable.</p>

<div id="target" ondrop="drop\_handler(event);" ondragover="dragover\_handler(event);">Drop Zone</div>

</body>

Para obtener más información, consulte [Realización de una gota](https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Drag_operations#drop) .

fin de arrastrar

Al final de una operación de arrastre, los [dragend](https://developer.mozilla.org/en-US/docs/Web/Events/dragend" \o "/en-US/docs/Web/Events/dragend)incendios de evento en la *fuente de* elemento - el elemento que era el objetivo del inicio de arrastre. Este evento se dispara si el arrastre completó o se cancela. El [dragend](https://developer.mozilla.org/en-US/docs/Web/Events/dragend" \o "/en-US/docs/Web/Events/dragend)controlador de eventos puede comprobar el valor de la[dropEffect](https://developer.mozilla.org/en-US/docs/Web/API/DataTransfer/dropEffect)propiedad para determinar si la operación de arrastre tuvo éxito o no.

Para obtener más información acerca de la manipulación al final de una operación de arrastre, ver [acabado de un Drag](https://developer.mozilla.org/en-US/docs/DragDrop/Drag_Operations#dragend) .

**interoperabilidad**[EDITAR](https://developer.mozilla.org/en-US/docs/Web/API/HTML_Drag_and_Drop_API$edit#Interoperability)

Como puede verse en la [tabla de compatibilidad del navegador de interfaz de DataTransferItem](https://developer.mozilla.org/docs/Web/API/DataTransferItem#Browser_compatibility) , arrastrar y soltar la interoperabilidad es relativamente amplio entre los navegadores de escritorio (excepto el [DataTransferItem](https://developer.mozilla.org/en-US/docs/Web/API/DataTransferItem" \o "El objeto DataTransferItem representa un elemento de datos de arrastre.  Durante una operación de arrastre, arrastre cada evento tiene una propiedad dataTransfer que contiene una lista de elementos de datos de arrastre.  Cada elemento de la lista es un objeto DataTransferItem.)e [DataTransferItemList](https://developer.mozilla.org/en-US/docs/Web/API/DataTransferItemList" \o "El objeto DataTransferItemList es una lista de objetos que representan elementos DataTransferItem ser arrastrado.  Durante una operación de arrastre, cada uno tiene una propiedad DragEvent dataTransfer y que la propiedad es un DataTransferItemList.)interfaces tienen menos apoyo). Estos datos también indican arrastrar y soltar apoyo entre los navegadores móviles es muy baja.

**Los ejemplos y demostraciones**[EDITAR](https://developer.mozilla.org/en-US/docs/Web/API/HTML_Drag_and_Drop_API$edit#Examples_and_demos)

* [Copia y elementos móviles con la DataTransferinterfaz](http://mdn.github.io/drag-and-drop/copy-move-DataTransfer.html)
* [Copia y elementos móviles con la DataTransferListIteminterfaz](http://mdn.github.io/drag-and-drop/copy-move-DataTransferItemList.html)
* Arrastrar y soltar archivos; Sólo Firefox: <http://jsfiddle.net/9C2EF/>
* Arrastrar y soltar archivos; Todos los navegadores: [https://jsbin.com/hiqasek/](https://jsbin.com/hiqasek/edit?html,js,output)

**ver también**[EDITAR](https://developer.mozilla.org/en-US/docs/Web/API/HTML_Drag_and_Drop_API$edit#See_also)

* [Las operaciones de arrastrar](https://developer.mozilla.org/Web/Guide/HTML/Drag_operations)
* [Arrastrar y soltar los diversos elementos](https://developer.mozilla.org/Web/Guide/HTML/Dragging_and_Dropping_Multiple_Items)
* [Tipos Arrastre recomendados](https://developer.mozilla.org/Web/Guide/HTML/Recommended_Drag_Types)
* [HTML5 Nivel de Vida: arrastrar y soltar](https://html.spec.whatwg.org/multipage/interaction.html#dnd)
* [Arrastre y la gota de interoperabilidad de datos de CanIUse](http://caniuse.com/#search=draganddrop)

**Documento Etiquetas y contribuyentes**

**Etiquetas:**

* [Avanzado](https://developer.mozilla.org/en-US/docs/tag/Advanced)

* [arrastrar y soltar](https://developer.mozilla.org/en-US/docs/tag/drag%20and%20drop)

* [Eventos](https://developer.mozilla.org/en-US/docs/tag/Events)

* [eventos](https://developer.mozilla.org/en-US/docs/tag/events)

* [Guía](https://developer.mozilla.org/en-US/docs/tag/Guide)

* [HTML5](https://developer.mozilla.org/en-US/docs/tag/HTML5)

* [visión de conj](https://developer.mozilla.org/en-US/docs/tag/Overview)