[illegible]

[illegible]

- [illegible]
- DragEvent [illegible]
- DataTransfer [illegible]
- DataTransfer [illegible]
  - DataTransfer.dropEffect
  - DataTransfer.effectAllowed
  - DataTransfer.files
  - DataTransfer.types
  - DataTransfer.items
- DataTransfer [illegible]
  - DataTransfer.setData()
  - DataTransfer.getData()
  - DataTransfer.clearData()
  - DataTransfer.setDragImage()

[illegible]

[illegible] drag [illegible]

[illegible] `draggable` [illegible] `true` [illegible]

```
<div draggable="true">
  [illegible]
</div>
```

[illegible] `div`
[illegible]

`draggable` [illegible] `<img>` [illegible] `<a>` [illegible] `false` [illegible]

[illegible] `draggable` [illegible] `true` [illegible]

[illegible]

- `drag` [illegible]
- `dragstart` [illegible] `target` [illegible]
- `dragend` [illegible] ESC [illegible] `target` [illegible] `dragstart` [illegible] `dragend` [illegible]
- `dragenter` [illegible] `target` [illegible] drop [illegible]
- `dragover` [illegible] `target` [illegible] `dragenter` [illegible] `dragenter` [illegible] `dragover` [illegible]
- `dragleave` [illegible] `target` [illegible]
- `drop` [illegible] `drop` [illegible] ESC [illegible]

[illegible]

```
div.addEventListener('dragstart', function (e) {
  this.style.backgroundColor = 'red';
}, false);

div.addEventListener('dragend', function (e) {
  this.style.backgroundColor = 'green';
}, false);
```

[illegible] `div` [illegible]

[illegible]

```
/* HTML □□□□
 <div class="dropzone">
   <div id="draggable" draggable="true">
     □□□□□□□
   </div>
 </div>
 <div class="dropzone"></div>
 <div class="dropzone"></div>
 <div class="dropzone"></div>
*/

// □□□□□
var dragged;

document.addEventListener('dragstart', function (event) {
  // □□□□□□□□
  dragged = event.target;
  // □□□□□□□□□□□□□□
  event.target.style.opacity = 0.5;
}, false);

document.addEventListener('dragend', function (event) {
  // □□□□□□□□□□□□□□□
  event.target.style.opacity = '';
}, false);

document.addEventListener('dragover', function (event) {
  // □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
  event.preventDefault();
}, false);

document.addEventListener('dragenter', function (event) {
  // □□□□□□□□□□□□□
  // □□□□□□□□□□□□□□□□□□□
  if (event.target.className === 'dropzone') {
    event.target.style.background = 'purple';
  }
}, false);

document.addEventListener('dragleave', function( event ) {
```

```
  // □□□□□□□□□□□□□□□
  if (event.target.className === 'dropzone') {
    event.target.style.background = '';
  }
}, false);

document.addEventListener('drop', function( event ) {
  // □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
  event.preventDefault();
  if (event.target.className === 'dropzone') {
    // □□□□□□□□□□
    event.target.style.background = '';
    // □□□□□□□□□□□□□□
    dragged.parentNode.removeChild(dragged);
    event.target.appendChild( dragged );
  }
}, false);
```

□□□□□□□□□□□□□□□□

- □□
- □□□□□□□□□□□□□□□□□□□□□□ `dragstart` □ `dragend` □□□
- `dragenter` □ `dragover`
  □□
  □□
  □□

```
<div ondragover="return false">
<div ondragover="event.preventDefault()">
```

□□ `div` □□□□□□□□□□□□□□□□

DragEvent □□

□□□□□□□□ `DragEvent` □□□□□□□□□□□□□□ `MouseEvent` □□□ `Event` □□□

□□□□□□□□□ `DragEvent()` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
new DragEvent(type, options)
```

`DragEvent()`

`MouseEvent` `Event`

`dataTransfer` `null` `DataTransfer`

`DataTransfer` DataTransfer

DataTransfer

`DragEvent.dataTransfer`
`DataTransfer`

`DataTransfer()` `DataTransfer`

```
var dataTrans = new DataTransfer();
```

`DataTransfer()`

MIME
`text/plain` `image/jpeg`

URL

`dragenter` `dragover`
drop
`text/uri-list`

`drop`

DataTransfer

DataTransfer.dropEffect

`DataTransfer.dropEffect` drop

- copy
- move
- link
- none

[illegible]

```
target.addEventListener('dragover', function (e) {
  e.preventDefault();
  e.stopPropagation();
  e.dataTransfer.dropEffect = 'copy';
});
```

[illegible] `drop` [illegible]

`dropEffect` [illegible] `dragenter` [illegible] `dragover` [illegible] `dragstart` [illegible] `drag` [illegible] `dragleave` [illegible]

DataTransfer.effectAllowed

`DataTransfer.effectAllowed` [illegible]

- copy [illegible]
- move [illegible]
- link [illegible]
- copyLink [illegible] `copy` [illegible] `link`
- copyMove [illegible] `copy` [illegible] `move`
- linkMove [illegible] `link` [illegible] `move`
- all [illegible]
- none [illegible]
- uninitialized [illegible] `all`

[illegible]

[illegible] `dropEffect` [illegible]

`dragstart` [illegible]

```
source.addEventListener('dragstart', function (e) {
  e.dataTransfer.effectAllowed = 'move';
});

target.addEventListener('dragover', function (e) {
  e.dataTransfer.dropEffect = 'move';
});
```

[illegible] `dropEffect` [illegible] `effectAllowed` [illegible] `none` [illegible] `drop` [illegible]

DataTransfer.files

`DataTransfer.files` [illegible] FileList [illegible] FileList [illegible]

[illegible]

```
// HTML [illegible]
// <div id="output" style="min-height: 200px;border: 1px solid black;">
//   [illegible]
// </div>

var div = document.getElementById('output');

div.addEventListener("dragenter", function( event ) {
  div.textContent = '';
  event.stopPropagation();
  event.preventDefault();
}, false);

div.addEventListener("dragover", function( event ) {
  event.stopPropagation();
  event.preventDefault();
}, false);

div.addEventListener("drop", function( event ) {
  event.stopPropagation();
  event.preventDefault();
  var files = event.dataTransfer.files;
  for (var i = 0; i < files.length; i++) {
    div.textContent += files[i].name + ' ' + files[i].size + '[illegible]\n';
  }
}, false);
```

[illegible] `dataTransfer.files` [illegible] `FileReader` [illegible]

```
div.addEventListener('drop', function(e) {
  e.preventDefault();
  e.stopPropagation();

  var fileList = e.dataTransfer.files;
  if (fileList.length > 0) {
    var file = fileList[0];
    var reader = new FileReader();
    reader.onloadend = function(e) {
      if (e.target.readyState === FileReader.DONE) {
        var content = reader.result;
        div.innerHTML = 'File: ' + file.name + '\n\n' + content;
      }
    }
    reader.readAsBinaryString(file);
  }
});
```

DataTransfer.types

`DataTransfer.types` MIME
`text/plain`

`dataTransfer` `drop`

```
function contains(list, value){
  for (var i = 0; i < list.length; ++i) {
    if(list[i] === value) return true;
  }
  return false;
}

function doDragOver(event) {
  var isLink = contains(event.dataTransfer.types, 'text/uri-list');
  if (isLink) event.preventDefault();
}
```

DataTransfer.items

`DataTransfer.items` DataTransferItemList
DataTransferItem

DataTransferItemList

- `length`
- `add(data, type)` `text/html` `text/plain`
- `add(file)` `add`
- `remove(index)`
- `clear()`

DataTransferItem

- `kind` `string` `file`
- `type` MIME
- `getAsFile()` `null`
- `getAsString(callback)`

```
div.addEventListener('drop', function (e) {
  e.preventDefault();
  if (e.dataTransfer.items != null) {
    for (var i = 0; i < e.dataTransfer.items.length; i++) {
      console.log(e.dataTransfer.items[i].kind + ': ' + e.dataTransfer.items[i].type);
    }
  }
});
```

DataTransfer

DataTransfer.setData()

`DataTransfer.setData()`

```
event.dataTransfer.setData('text/plain', 'Text to drag');
```

[illegible]

[illegible] `text/plain` [illegible] `dataTransfer` [illegible]

[illegible] `dataTransfer` [illegible]

```
<div draggable="true">
  aaa
</div>
```

[illegible] `<div>` [illegible] `aaa` [illegible]

[illegible] `setData` [illegible]

```
<div
  draggable="true"
  ondragstart="event.dataTransfer.setData('text/plain', 'bbb')"
>
  aaa
</div>
```

[illegible] `bbb` [illegible] `aaa` [illegible]

[illegible] `text/plain` [illegible]

```
var dt = event.dataTransfer;

// [illegible]
dt.setData('text/uri-list', 'http://www.example.com');
dt.setData('text/plain', 'http://www.example.com');

// [illegible] HTML [illegible]
dt.setData('text/html', 'Hello there, <strong>stranger</strong>');
dt.setData('text/plain', 'Hello there, <strong>stranger</strong>');

// [illegible] URL
```

```
dt.setData('text/uri-list', imageurl);
dt.setData('text/plain', imageurl);
```

[illegible]

```
var dt = event.dataTransfer;
dt.setData('application/x-bookmark', bookmarkString);
dt.setData('text/uri-list', 'http://www.example.com');
dt.setData('text/plain', 'http://www.example.com');
```

[illegible] `drop` [illegible] `drop` [illegible]

DataTransfer.getData()

`DataTransfer.getData()` [illegible] `setData` [illegible] `drop` [illegible]

[illegible] `drop` [illegible]

```
function onDrop(event) {
  var data = event.dataTransfer.getData('text/plain');
  event.target.textContent = data;
  event.preventDefault();
}
```

[illegible]

`getData` [illegible]

```
function doDrop(event) {
  var lines = event.dataTransfer.getData('text/uri-list').split('\n');
  for (let line of lines) {
    let link = document.createElement('a');
    link.href = line;
    link.textContent = line;
    event.target.appendChild(link);
  }
```

```
  event.preventDefault();
}
```

[illegible] `getData` [illegible]

[illegible] `URL` [illegible]

```
var link = event.dataTransfer.getData('URL');
```

[illegible]

```
function doDrop(event) {
  var types = event.dataTransfer.types;
  var supportedTypes = ['text/uri-list', 'text/plain'];
  types = supportedTypes.filter(function (value) { types.includes(value) });
  if (types.length) {
    var data = event.dataTransfer.getData(types[0]);
  }
  event.preventDefault();
}
```

DataTransfer.clearData()

`DataTransfer.clearData()`
[illegible]
[illegible]

```
event.dataTransfer.clearData('text/uri-list');
```

[illegible] `text/uri-list` [illegible]

[illegible] `DataTransfer.types` [illegible] `Files`
[illegible]

[illegible] `dragstart` [illegible]

DataTransfer.setDragImage()

[illegible] `dragstart`
[illegible]
[illegible]

DataTransfer.setDragImage()[illegible] <img>[illegible]
<canvas>[illegible] null
[illegible]

[illegible]

```
/* HTML [illegible]
 <div id="drag-with-image" class="dragdemo" draggable="true">
   drag me
 </div>
*/

var div = document.getElementById('drag-with-image');
div.addEventListener('dragstart', function (e) {
  var img = document.createElement('img');
  img.src = 'http://path/to/img';
  e.dataTransfer.setDragImage(img, 0, 0);
}, false);
```

[illegible] #2
[illegible] 2018-05-08 12:18:12 UTC
[illegible] 2021-02-12 17:37:57 UTC