

Drag and Drop

ドラッグ要素：要素をドラッグ可能にする

ドラッグしたい要素のプロパティに `draggable = "true"` を追加

`img` 要素と `a` 要素は default が `draggable = "true"`

DragEvent

eventの種類	イベントが発生するタイミング	デフォルト動作
dragstart	ドラッグ開始時	ドラッグを開始する
drag	ドラッグが継続している間	ドラッグを継続する
dragenter	ドラッグ要素がドロップ要素に入った時	ユーザーによるドロップ操作を拒否
dragleave	ドラッグ要素がドロップ要素から出た時	何もしない
dragover	ドラッグ要素がドロップ要素に重なっている間	現在のドラッグ操作をリセットする <code>event.preventDefault()</code> 必須
drop	ドラッグ要素が有効なドロップ対象にドロップされた時	(場合による)
dragend	ドラッグ終了時	(場合による)

event.dataTransfer

ドラッグ項目のデータを保持するプロパティで値は `DataTransfer` オブジェクト

`event.dataTransfer.setData(format, data)` // データをセットする

例：`event.dataTransfer.setData("text/plain", event.target.id)`

`event.dataTransfer.getData(format)` // データを取得する

例：`const targetId = event.dataTransfer.getData("text/palin");`

```

<ul id="dragUL">
  <li id="item1" draggable="true">item1</li>
  <li id="item2" draggable="true">item2</li>
  <li> </li>
</ul>

```

```

<ul id="dropUL">
  <li class="dropLi"> </li>
</ul>

```

① drag要素 (item1) がdragされると dragstart イベントが発火
dropの際にdrag要素を取得する目的で、
dataTransferにid名を記録する。

```

elm.ondragstart = function(event){
  event.dataTransfer.setData('text/plain',
    event.target.id);
};

```

② drag要素 (item1) がdrop要素 (dropLi) に重
なるとdrop要素でdragoverイベントが発火
drop要素のstyleをborderTop="2px solid blue"
としてdrop位置を明示する。

```

elm.ondragover = function(event){
  event.preventDefault(); // 必須
  this.style.borderTop = "2px solid blue";
};

```

③ drag要素 (item1) がdrop要素 (dropLi) に重なった状態で
マウスボタンが離されるとdrop要素でdropイベントが発火
dataTransferからdrag要素のid名を取得
id名からdrag要素を取得、drag要素をdrop要素の前に挿入
drop要素のstyleを削除

```

elm.ondrop = function(event){
  event.preventDefault();
  const id_dragElm = event.dataTransfer.getData('text/plain');
  const dragElm = document.getElementById(id_dragElm);
  this.parentNode.insertBefore(dragElm, this);
  this.style.borderTop = "";
};

```

Draggable Lists

```
<!DOCTYPE html>
<html lang="ja">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>drag and drop Lists</title>
  <style>
    ul{ list-style-type: none; border: solid 1px gray; }
    ul#dropUL{ height: 7rem;}
  </style>
</head>
<body>
<p>drop ゾーンへドラッグして下さい。</p>
<ul id="dragUL">
  <li id="item1" draggable="true">item1</li>
  <li id="item2" draggable="true">item2</li>
  <li id="item3" draggable="true">item3</li>
  <li> </li>
</ul>

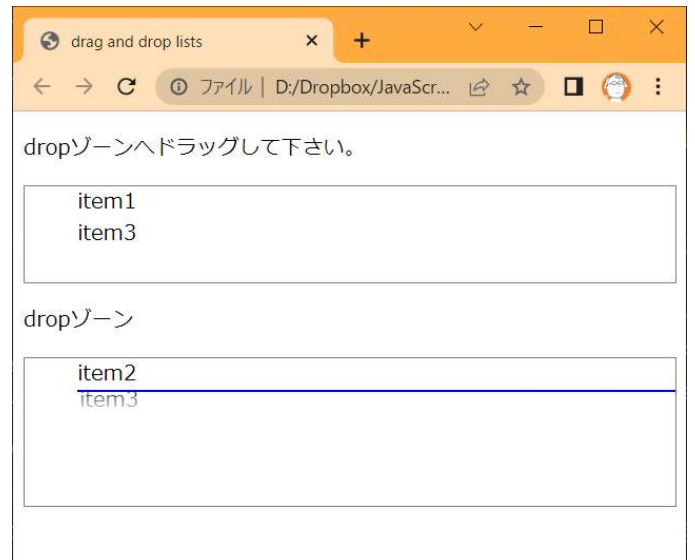
<p>drop ゾーン</p>
<ul id="dropUL">
  <li> </li>
</ul>

<script>
document.querySelectorAll('#dragUL li').forEach(elm => {
  // li 要素を draggable にする
  // elm.setAttribute('draggable', 'true');

  // li 要素がドラッグ開始したら DataTransfer オブジェクトに li 要素の id 名を登録する
  elm.ondragstart = function(event){
    event.dataTransfer.setData('text/plain', event.target.id);
  };

  // drag 要素が elm に重なっている間, elm 要素の borderTop を青にして挿入位置を明示する
  elm.ondragover = function(event){
    event.preventDefault(); // 必須
    this.style.borderTop = "2px solid blue"; // this = elm
  };

  // drag 要素 (li 要素) が他の要素 (ドロップ要素) から出たら li 要素の枠線を消す
  elm.ondragleave = function(event){
    this.style.borderTop = "";
  };
};
```



```

};

// drag 要素が drop されたら dataTransfer に登録された文字を取得して、
// それを元にして drag 要素を取得して、drop 要素の前に挿入する
elm.ondrop = function(event){
    event.preventDefault(); // 書くべき
    const id_dragElm = event.dataTransfer.getData('text/plain');
    const dragElm = document.getElementById(id_dragElm);
    // insertBefore(newNode, referenceNode);
    this.parentNode.insertBefore(dragElm, this); // this = elm
    this.style.borderTop = "";
};

});

document.querySelectorAll('#dropUL li').forEach(elm => {
    // drag 要素が elm に重なっている間、elm 要素の borderTop を青にして挿入位置を明示する
    elm.ondragover = function(event){
        event.preventDefault(); // 必須
        this.style.borderTop = "2px solid blue";
    };

    // drag 要素が elm から離れたら枠線を消す
    elm.ondragleave = function(event){
        this.style.borderTop = "";
    };

    // drag 要素が drop されたら dataTransfer に登録された文字を取得して、
    // それを元にして drag 要素を取得して drop 要素前に挿入する
    elm.ondrop = function(event){
        event.preventDefault();
        const id_dragElm = event.dataTransfer.getData('text/plain');
        const dragElm = document.getElementById(id_dragElm);
        this.parentNode.insertBefore(dragElm, this);
        this.style.borderTop = "";
    };

});

</script>
</body>
</html>

```

参考：<https://blog.ver001.com/javascript-dragdrop-sort/>
https://developer.mozilla.org/ja/docs/Web/API/HTML_Drag_and_Drop_API