

## 課題 1 2 : ドラッグ&ドロップ処理を組み込もう

HTML5 ではドラッグ用の専用プロパティが用意されているので、ドラッグ処理をやってみよう。

### 【index.html】

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>ドラッグドロップする</title>
    <link rel="stylesheet" href="css/main.css" type="text/css" media="all">
    <script src="js/dragdrop.js"></script>
  </head>
  <body>
    <h1>ドラッグドロップ</h1>
    <div id="dragItems">
      <div>項目 1</div>
      <div>項目 2</div>
      <div>項目 3<br></div>
      <div></div>
    </div>
    <div id="dropArea">
      ドロップ領域/
    </div>
  </body>
</html>
```

### 【dragdrop.js】

```
// ドラッグ&ドロップ処理をする
window.addEventListener("load", function(){
  // ドラッグ側の処理
  var itemList = document.querySelectorAll("#dragItems div");
  for(var i=0; i<itemList.length; i++){
    itemList[i].draggable = true;      // Safari, Chrome
    itemList[i].id = "dragNo"+i;      // ID を割り当てる
    itemList[i].addEventListener("dragstart", function(evt){
      evt.dataTransfer.setData("text", evt.currentTarget.id);
    }, true);
  }
});
```

```
}
// ドロップ側の処理
var drop = document.getElementById("dropArea");
drop.addEventListener("drop", function(evt){
    evt.preventDefault();
    var ID = evt.dataTransfer.getData("text");
    try{
        drop.innerHTML += document.getElementById(ID).innerHTML+"/";
    }catch(e){
        drop.innerHTML += "デスクトップからドロップされました/";
    }
    drop.style.backgroundColor = "#eee";
}, true);
drop.addEventListener("dragenter", function(evt){
    evt.preventDefault();
    drop.style.backgroundColor = "#ecc";
}, true);
drop.addEventListener("dragleave", function(evt){
    evt.preventDefault();
    drop.style.backgroundColor = "#eee";
}, true);
// このイベントは必須
drop.addEventListener("dragover", function(evt){
    evt.preventDefault();
}, true);
}, true);
```

#### 【main.css】

```
h1 {
    font-size:14pt;
    width: 100%;
}
#dragItems div {
    position: absolute;
    top: 80px;
    background-color: orange;
    border: 1px solid red;
    width: 120px;
```

```
        height: 120px;
    }
    #dragItems div:nth-child(1) {
        left: 10px;
    }
    #dragItems div:nth-child(2) {
        left: 110px;
    }
    #dragItems div:nth-child(3) {
        left: 210px;
    }
    #dragItems div:nth-child(4) {
        left: 310px;
    }
    #dropArea {
        position: absolute;
        top: 220px;
        background-color: #ddd;
        border: 1px solid black;
        width: 500px;
        height: 300px;
    }
```

ドラッグ側の処理。まず `querySelectorAll("#dragItems div");` で `dragItems` の中の `div` の一覧を取得します。それを `for` で回しながら、その全てにドラッグ可能のプロパティをセットして、`id` を付けます。そして、ドラッグが始まった時のイベントに、ドロップ側のパラメータ `evt` に対して `setData` で、文字としてデータを渡すことにします。

ドロップ側の処理。ドロップエリアにドロップされると `drop` イベントが発生するので、その `id` の `innerHTML` つまり `div` の内容をドロップ領域の `innerHTML` に追加します。