* **Drag**드래그하고 싶은 엘리먼트에 드래그 가능여부 속석을 추가하고 드래그시 발생할 이벤트를 등록한다.  
  + 먼저 드래그 하고 싶은 엘리먼트에 **draggable 속성을 true**로 작성한다.   
    <div id="draggable" draggable="true"></div>  
    아직 draggable는 드래그 되는것 외에는 어떠한 기능도 없다.
  + **ondragstart** 이벤트를 등록한다.  
    <div id="draggable" draggable="true" ondragstart="drag()"></div>  
    ondragstart이벤트를 통해 앨리먼트가 드래그 되면 drag() 함수를 호출한다.
* **Drop**  
  드래그한 엘리먼트를 드롭받을 엘리먼트를 정할 수 있다.  
  드롭받는 엘리먼트에는 **ondrop, ondragover** 두가지 이벤트를 등록할 수 있다.  
  + ondragover는 드래그된 엘리먼트가 드랍될 수 있을때 발생한다.  
    <div id="droppable" ondragover="ondrag(event)"></div>  
    기본적으로 엘리먼트는 다른 엘리먼트로 드롭될 수 없기때문에 기본 이벤트를 제한해야한다.  
    ondrag함수내에 event.preventDefault()를 작성하면 ondragover의 디폴트 이벤트가 호출되지 않는다.
  + 드래그한 엘리먼트를 drop할때 발생한 이벤트를 등록한다.  
    <div id="droppable" ondrop="drop(event)"></div>  
    드래그된 엘리먼트가 droppable에 drop되면 drop()함수가 호출된다.
* javascript 만으로 drag&drop자체의 구현은 어렵지 않지만 다양한 기능을 사용하기위해서는 브라우져별로 구현해줘야 할 것들이 많다. 브라우저에 영향없이 다양하고 강력한 기능이 있는 jQueryui의 dragNdrop에 대해서 알아보겠다.

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()

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));  
}

Execute a JavaScript when the user has finished dragging a <p> element:

**Ex01)**

<!DOCTYPE HTML>

<html>

<head>

<style>

#div1 {

width: 350px;

height: 70px;

padding: 10px;

border: 1px solid #aaaaaa;

}

</style>

<script>

function allowDrop(ev) {

ev.preventDefault();

}

function drag(ev) {

ev.dataTransfer.setData("Text", ev.target.id);

}

function drop(ev) {

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

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

ev.preventDefault();

}

</script>

</head>

<body>

<div id="div1" ondrop="drop(event)" ondragover="allowDrop(event)"></div>

<br>

<p id="drag1" draggable="true" ondragstart="drag(event)">This is a draggable paragraph. Drag this element into the rectangle.</p>

</body>

</html>

Ex02)

<!DOCTYPE HTML>

<html>

<head>

<style>

#div1, #div2

{float:left; width:100px; height:35px; margin:10px;padding:10px;border:1px solid #aaaaaa;}

</style>

<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)">

<img src="img\_w3slogo.gif" draggable="true" ondragstart="drag(event)" id="drag1" width="88" height="31">

</div>

<div id="div2" ondrop="drop(event)" ondragover="allowDrop(event)"></div>

</body>

</html>

**Ex03)**

<!DOCTYPE HTML>

<html>

<head>

<style>

.droptarget {

float: left;

width: 100px;

height: 35px;

margin: 15px;

padding: 10px;

border: 1px solid #aaaaaa;

}

</style>

</head>

<body>

<p>Drag the p element back and forth between the two rectangles:</p>

<div class="droptarget" ondrop="drop(event)" ondragover="allowDrop(event)">

<p ondragstart="dragStart(event)" draggable="true" id="dragtarget">Drag me!</p>

</div>

<div class="droptarget" ondrop="drop(event)" ondragover="allowDrop(event)"></div>

<p style="clear:both;"><strong>Note:</strong> drag events are not supported in Internet Explorer 8 and earlier versions or Safari 5.1 and earlier versions.</p>

<p id="demo"></p>

<script>

function dragStart(event) {

event.dataTransfer.setData("Text", event.target.id);

document.getElementById("demo").innerHTML = "Started to drag the p element";

}

function allowDrop(event) {

event.preventDefault();

}

function drop(event) {

event.preventDefault();

var data = event.dataTransfer.getData("Text");

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

}

</script>

</body>

</html>

A node list is a collection of nodes

The getElementsByTagName() method returns a **node list**. A node list is an array-like collection of nodes.

The following code selects all <p> nodes in a document:

var x = document.getElementsByTagName("p");

The nodes can be accessed by an index number. To access the second <p> node you can write:

y = x[1];

<!DOCTYPE html>

<html>

<body>

<p>Hello World!</p>

<p>The DOM is very useful!</p>

<p id="demo"></p>

<script>

var myNodelist = document.getElementsByTagName("p");

document.getElementById("demo").innerHTML =

"The innerHTML of the second paragraph is: " +

myNodelist[1].innerHTML;

</script>

</body>

</html>

## HTML DOM Node List Length

The length property defines the number of nodes in a node list

Example explained:

1. Get all <p> elements in a node list
2. Display the length of the node list

<!DOCTYPE html>

<html>

<body>

<p>Hello World!</p>

<p>How many paragraphs in this document?</p>

<p>This example demonstrates the length property of a nodelist.</p>

<p id="demo"></p>

<script>

var myNodelist = document.getElementsByTagName("p");

document.getElementById("demo").innerHTML = myNodelist.length;

</script>

</body>

</html>

Ex02)

Change the background color of all <p> elements in a node list:

<!DOCTYPE html>

<html>

<body>

<p>This is a p element</p>

<p>This is also a p element.</p>

<p>This is also a p element - Click the button to change the background color of all p elements in this document.</p>

<button onclick="myFunction()">Try it</button>

<script>

function myFunction() {

var myNodelist = document.getElementsByTagName("p");

var i;

for (i = 0; i < myNodelist.length; i++) {

myNodelist[i].style.backgroundColor = "red";

}

}

</script>

</body>

</html>