

JavaScript

드래그 앤 드롭 - 김근형 강사

드래그 앤 드롭

- 드래그 앤 드롭 API
 - 드래그 앤 드롭 기능은 웹 표준으로 규정되어 있지 않았음
 - HTML5 에서 IE가 미리 구현해놨던 드래그 앤 드롭 API를 웹 표준으로 채용하게 됨
 - 처음에는 HTML5 명세에 클립 보드도 포함되었지만 현재는 클립 보드는 제외되고 드래그 앤 드롭 제어만이 웹 표준으로 채용되었다.

드래그 앤 드롭

- draggable 속성
 - img 와 같은 속성은 마우스로 드래그 할 수 있는 요소로 규정되어 있다.
 - 하지만 모든 속성이 드래그 할 수 있는 속성은 아니며 반드시 draggable이란 속성을 통해 해당 요소를 드래그 할 수 있도록 설정해야 한다.
 - draggable 속성에 "true"를 지정하면 해당 요소 드래그가 가능하고 "false"일 경우 드래그를 못하도록 설정할 수 있다.

```
<div draggable="true">드래그 가능</div>  
<div draggable="false">드래그 불가능</div>
```

드래그 앤 드롭

○ 드래그 앤 드롭의 이벤트

- 마우스로 요소를 드래그하거나 드롭을 하면 다양한 이벤트가 발생한다.
- 다음 표는 HTML5 명세에 규정된 드래그 앤 드롭 관련 이벤트이다.

드래그 측에서 발생하는 이벤트	설명
dragstart	드래그 요소를 마우스 버튼을 눌러 드래그하기 시작했을 때 드래그 요소에서 발생한다.
drag	드래그 요소를 마우스 버튼을 누른 상태에서 움직여 드래그하는 중에 드래그 요소에서 발생한다.
dragend	드래그가 종료했을 때 발생한다.

드롭 측에서 발생하는 이벤트	설명
dragenter	드래그 요소가 드롭 요소에 들어왔을 때 드롭 요소에서 발생한다.
dragleave	드래그 요소가 드롭 요소에서 나갔을 때 드롭 요소에서 발생한다.
dragover	드래그 요소가 드롭 요소 위에서 드래그되고 있을 때 드롭 요소에서 연속해서 발생한다.
drop	드래그 요소가 드롭 요소에 드롭되었을 때 드롭 요소에서 발생한다.

드래그 앤 드롭

○ 드래그 앤 드롭의 이벤트

- 기본 액션이란 기본적으로 태그가 갖고있는 액션을 의미한다 ex) a 링크를 클릭 시 다른 화면으로 이동
- 보통 드래그 앤 드롭의 이벤트는 기본 액션을 취소해야 할 경우가 많은데 기본 액션은 아래와 같다.

드래그 측에서 발생하는 이벤트	기본 액션
dragstart	드래그 앤 드롭 조작을 초기화한다
drag	드래그를 지속한다.
dragend	무엇이 어디에 드롭됐는지에 따라 변한다

드롭 측에서 발생하는 이벤트	설명
dragenter	드래그되는 요소에 드롭할 수 없게 된다.
dragleave	없음
dragover	드래그 조작을 무효화 한다.
drop	무엇이 어디에 드롭됐는지에 따라 변한다.

드래그 앤 드롭

○ DataTransfer

- drag 관련 event 에서 얻을 수 있는 객체
- 드래그 앤 드롭 동작 중에 끌고있는 데이터를 보유하기 위해 사용한다.
- DataTransfer 에서 제공하는 기본적인 메서드와 속성은 다음과 같다.

메서드/속성	설명
<code>event.dataTransfer.setData(format, data)</code>	첫 번째 매개변수로 포맷 문자열을 지정한다. 이곳에 지정한 포맷과 일치하는 값을 두 번째 매개변수로 지정한다. 두 번째 매개변수로는 문자열만 지정할 수 있다.
<code>event.dataTransfer.getData(format)</code>	첫 번째 매개변수에 지정한 포맷의 전송 데이터를 반환한다. 지정된 포맷의 데이터가 지정되어 있지 않으면 공백 문자열을 반환한다.
<code>event.dataTransfer.clearData()</code> <code>event.dataTransfer.clearData(format)</code>	데이터 전송용으로 지정된 데이터를 모두 제거한다 첫 번째 매개변수로 포맷 문자열을 지정하면 해당 형식과 일치하는 데이터만을 제거한다.
<code>event.dataTransfer.types</code>	dragstart 이벤트 리스너로 <code>setData()</code> 함수를 사용하여 지정한 데이터의 포맷 리스트를 <code>DOMStringList</code> 객체로 반환한다.

드래그 앤 드롭

○ DataTransfer format

format 명	설명
text	텍스트 데이터를 나타낸다. 브라우저에서는 "text/plain" 으로 처리한다.
url	URL을 나타낸다. 브라우저에서는 "text/url-list" 로 처리한다. 하지만 복수의 URL 데이터를 전송하는 것은 아니다. <code>getData("url")</code> 함수로 얻을 수 있는 값은 하나다.

드래그 앤 드롭

○ DataTransfer 예제 - 1

```
<h2>상품 목록</h2>
<ul id="items">
  <li></li>
  <li></li>
  <li></li>
</ul>
<h2>장바구니</h2>
<ul id="cart">
</ul>
<script type="text/javascript">
  /* -----
  * ■드래그에 대한 처리
  * ----- */
  // 상품목록 이미지 리스트
  var imgs = document.querySelectorAll('#items>li>img');
  for( var i=0; i<imgs.length; i++ ) {
    // img
    var img = imgs.item(i);
    // img요소에 dragstart이벤트 리스너를 지정
    img.addEventListener("dragstart", function(event) {
      // 데이터 전송을 위한 데이터 지정
      event.dataTransfer.setData("text", event.target.id);
    }, false);
  }
}
```


드래그 앤 드롭

○ DataTransfer 예제 - 1

```
/* -----  
* ■드롭에 대한 처리  
* ----- */  
// 장바구니의 ul요소에 dragenter이벤트 리스너 지정  
var cart = document.querySelector('#cart');  
cart.addEventListener("dragenter", function(event) {event.preventDefault();}, false);  
//장바구니의 ul요소에 dragover이벤트 리스너 지정  
cart.addEventListener("dragover", function(event) {event.preventDefault();}, false);  
// 장바구니의 ul 요소에 drop이벤트 리스너 지정  
cart.addEventListener("drop", function(event) {  
    // 기본 액션을 취소  
    event.preventDefault();  
    // 데이터 전송으로 전송된 정보  
    var id = event.dataTransfer.getData("text");  
    // 드롭된 img요소  
    var img = document.getElementById(id);  
    // li요소를 생성하여 장바구니에 추가  
    var li = document.createElement("li");  
    li.appendChild(img);  
    cart.appendChild(li);  
}, false);
```

상품 목록



장바구니



드래그 앤 드롭

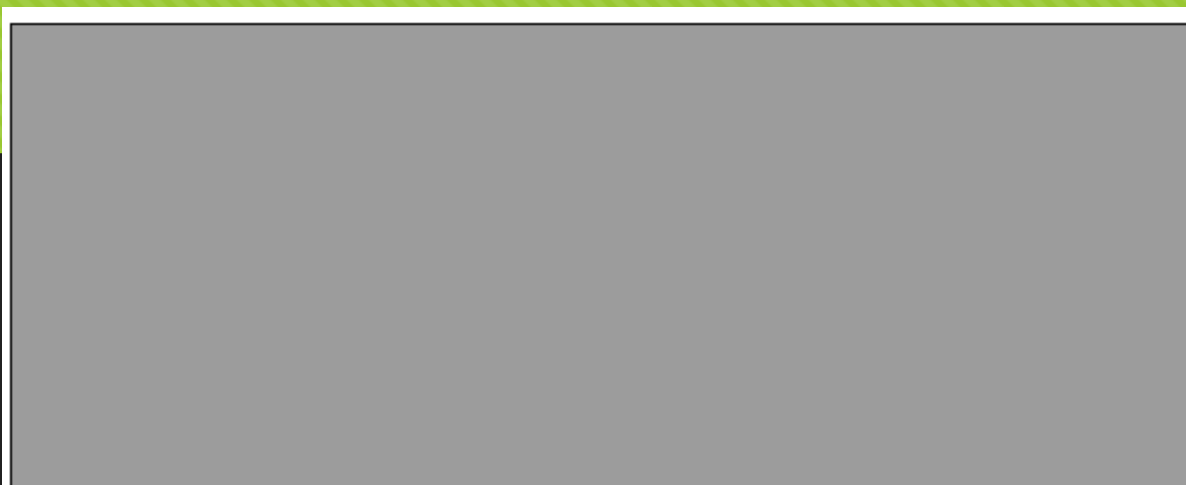
○ DataTransfer 예제 - 2

```
<ul id="items">
  <li id="b1" draggable="true">HTML5+자바스크립트</li>
  <li id="b2" draggable="true">웹 표준 가이드</li>
  <li id="b3" draggable="true">jQuery 입문</li>
</ul>
<h2>장바구니 </h2>
<ul id="cart"></ul>
<script type="text/javascript">
  /* -----
  * ■드래그 측 처리
  * ----- */
  // 상품목록의 이미지 리스트
  var list = document.querySelectorAll('#items>li');
  for( var i=0; i<list.length; i++ ) {
    // li요소
    var li = list.item(i);
    // li요소에 dragstart이벤트 리스너를 지정
    li.addEventListener("dragstart", function(event) {
      // 전송하기 위한 데이터 지정
      event.dataTransfer.setData("text", event.currentTarget.id);
    }, false);
  }
}
```

```
/* -----
* ■드롭 측 처리
* ----- */
// 장바구니의 ul요소에 dragenter 이벤트 리스너를 지정
var cart = document.querySelector('#cart');
cart.addEventListener("dragenter", function(event) {event.preventDefault();});
// 장바구니의 ul 요소에 dragover 이벤트 리스너 지정
cart.addEventListener("dragover", function(event) {event.preventDefault();});
// 장바구니의 ul 요소에 drop 이벤트 리스너를 지정
cart.addEventListener("drop", function(event) {
  // 기본 액션을 무효화
  event.preventDefault();
  // 전송된 데이터
  var id = event.dataTransfer.getData("text");
  // 드롭된 li 요소
  var li = document.getElementById(id);
  // li요소를 ul 요소에 추가
  cart.appendChild(li);
}, false);
```

드래그 앤 드롭

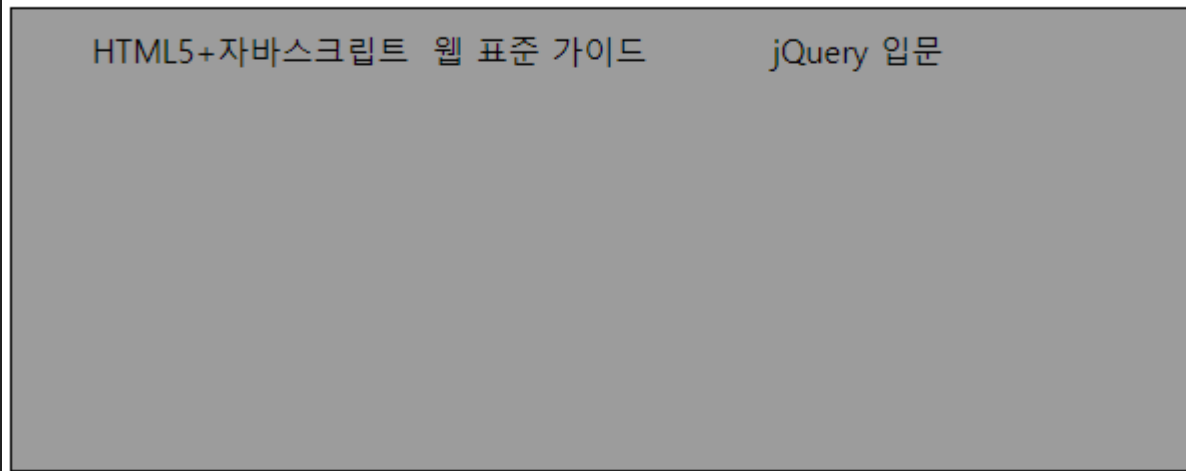
○ DataTransfer 예제 - 2



장바구니

HTML5+자바스크립트 웹 표준 가이드

jQuery 입문



드래그 앤 드롭

○ DataTransfer 예제 - 3

```
<p id="text">텍스트 영역을 선택해서 아래 영역에 드롭해 주세요</p>
<div id="drop"></div>
<script type="text/javascript">
  /* -----
  * ■드롭 측의 처리
  * ----- */
  // div요소에 dragenter이벤트 리스너 지정
  var div = document.querySelector('#drop');
  div.addEventListener("dragenter", function(event) {event.preventDefault();});
  // div요소에 dragover이벤트 리스너 지정
  div.addEventListener("dragover", function(event) {event.preventDefault();});
  // div요소에 drop이벤트 리스너 지정
  div.addEventListener("drop", function(event) {
    // 기본 액션 해제
    event.preventDefault();
    // 선택한 텍스트 얻기
    var text = event.dataTransfer.getData("text");
    // 선택한 텍스트를 div요소에 추가
    div.appendChild( document.createTextNode(text) );
  });
</script>
```

텍스트 영역을 선택해서 아래 영역에 드롭해 주세요

선택해서 아래 영

드래그 앤 드롭

○ DataTransfer 예제 - 4

```
<h2>상품 목록 </h2>
<ul id="items">
  <li></li>
  <li></li>
  <li></li>
</ul>
<h2>장바구니 </h2>
<ul id="cart"></ul>
<script type="text/javascript">
  /* -----
  * ■드래그 측의 처리
  * ----- */
  // 상품 목록의 이미지 리스트
  var imgs = document.querySelectorAll('#items>li>img');
  for( var i=0; i<imgs.length; i++ ) {
    // img 요소
    var img = imgs.item(i);
    // img요소에 dragstart이벤트 리스너 지정
    img.addEventListener("dragstart", function(event) {
      // 데이터 전송을 위한 데이터 지정
      event.dataTransfer.setData("src", event.target.src);
      event.dataTransfer.setData("title", event.target.title);
    });
  }
}
```

드래그 앤 드롭

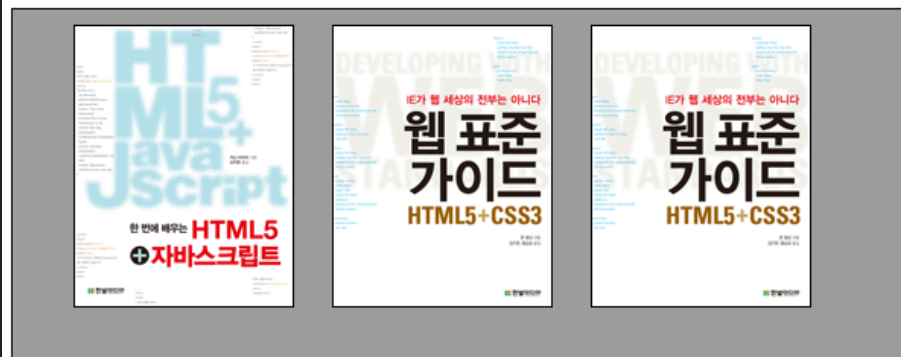
○ DataTransfer 예제 - 4

```
/* -----  
* ■ 드롭 측 처리  
* ----- */  
// 장바구니의 ul 요소에 dragenter 이벤트 리스너 지정  
var cart = document.querySelector('#cart');  
cart.addEventListener("dragenter", function(event) {event.preventDefault();});  
// 장바구니의 ul 요소에 dragover 이벤트 리스너 지정  
cart.addEventListener("dragover", function(event) {event.preventDefault();});  
// 장바구니의 ul 요소에 drop 이벤트 리스너 지정  
cart.addEventListener("drop", function(event) {  
    // 기본 액션 해제  
    event.preventDefault();  
    // img 요소를 생성  
    var img = document.createElement("img");  
    img.src = event.dataTransfer.getData("src");  
    img.title = event.dataTransfer.getData("title");  
    // li 요소를 생성하여 장바구니에 추가  
    var li = document.createElement("li");  
    li.appendChild(img);  
    cart.appendChild(li);  
});
```

상품 목록



장바구니



드래그 앤 드롭

○ DOMStringList

- DataTransfer 객체의 types 속성은 dragstart 이벤트 리스너에서 setData() 함수를 사용해 지정된 데이터 포맷의 목록을 DOMStringList 객체로 반환한다.
- DOMStringList에서는 다음과 같은 속성과 메서드가 사용된다.

메서드/속성	설명
DOMStringList.item(index)	DOMStringList 객체에 담긴 문자열 중 매개변수(index) 번째에 담긴 문자열을 반환한다. index는 0부터 시작하며 index 번째에 문자열이 없다면 null을 반환한다.
DOMStringList.length	DOMStringList 객체에 담긴 문자열의 수를 반환한다.
DOMStringList.contains(string)	DOMStringList 객체에 매개변수(String)로 지정한 문자열이 있으면 true 아니면 false 를 반환한다.

드래그 앤 드롭

○ DOMStringList 예제 - 1

```
cart.addEventListener("drop", function(event) {  
    // 기본 액션 해제  
    event.preventDefault();  
    // 생성 전 데이터 전송 확인  
    var types = event.dataTransfer.types;  
    if(types.contains("src")==false){return;}  
    // img 요소를 생성  
    var img = document.createElement("img");  
    img.src = event.dataTransfer.getData("src");  
    img.title = event.dataTransfer.getData("title");  
    // li 요소를 생성하여 장바구니에 추가  
    var li = document.createElement("li");  
    li.appendChild(img);  
    cart.appendChild(li);  
});
```


드래그 앤 드롭

○ 애플리케이션 연동

- HTML5의 명세의 드래그 앤 드롭 API는 브라우저가 아닌 PC 애플리케이션에 데이터를 보내거나 받는 상황도 고려한다.
- 이것은 HTML5에서 새로 도입한 확장 기능이다.
- 하지만 브라우저에서 기능을 구현해야 하고 OS가 PC 애플리케이션 간의 데이터 송수신을 지원하고 PC 애플리케이션이 MIME 형식에 따라 데이터 전송을 지원해야 한다.
- 이런 복잡한 구조를 일부 PC 애플리케이션에서 지원하며 자바스크립트의 사용 없이 어느정도 처음부터 애플리케이션 간의 데이터 전송을 구현할 수 있다.

```
<p>해당 이미지를 OS 의 그림파일이나 URL 형태로 다운이 가능하다.</p>  
<p></p>
```

해당 이미지를 OS 의 그림파일이나 URL 형태로 다운이 가능하다.



드래그 앤 드롭

- 드래그 중인 아이콘을 지정
 - 페이지의 콘텐츠를 드래그하면 드래그되는 콘텐츠가 마우스 포인터의 위치에 표시된다
 - HTML5 드래그 앤 드롭 API 에는 이 드래그 중인 아이콘을 변경하는 것이 가능하다.

메서드/속성	설명
<code>event.dataTransfer.setDragImage(element, x, y)</code>	드래그 중인 아이콘을 첫 번째 매개변수(element)에 지정한 요소가 나타내는 이미지로 바꾼다. 두 번째와 세 번째 매개변수는 마우스 포인터의 위치를 기준으로 상대적인 아이콘의 표시 위치를 나타낸다. 왼쪽으로 x 위로 y 만큼 아이콘을 이동시킨다.
<code>event.dataTransfer.addElement(element)</code>	지정한 요소의 렌더링 결과를 드래그 아이콘의 이미지로 추가한다. 첫 번째 매개변수로는 페이지에 표시된 요소를 지정한다.

드래그 앤 드롭

○ 드래그 중인 아이콘을 지정 예제 - 1

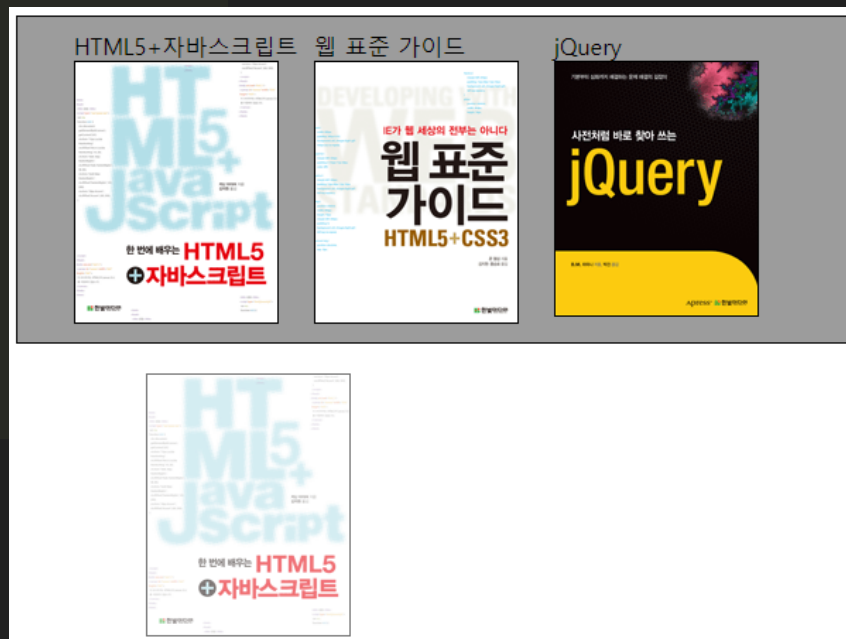
```
<p></p>
<script type="text/javascript">
    /* -----
    * ■드래그측의 처리
    * ----- */
    // 드래그 아이콘을 위한 이미지 생성
    var fb = document.createElement("img");
    fb.src = "fb.png";
    // 드래그할 img 요소에 dragstart 이벤트 리스너 지정
    var img = document.querySelector("#book");
    img.addEventListener("dragstart", function(event) {
        // 드래그 아이콘을 지정
        event.dataTransfer.setDragImage(fb, 30, 30);
    }, false);
</script>
```



드래그 앤 드롭

○ 드래그 중인 아이콘을 지정 예제 - 2

```
<ul id="items">
  <li id="b1" draggable="true">HTML5+자바스크립트 </li>
  <li id="b2" draggable="true">웹 표준 가이드</li>
  <li id="b3" draggable="true">jQuery </li>
</ul>
<script type="text/javascript">
  // li 요소에 dragstart 이벤트 리스너를 지정
  var lis = document.querySelectorAll('#items>li');
  for( var i=0; i<lis.length; i++ ) {
    lis.item(i).addEventListener("dragstart", function(event) {
      // 아이콘 표시를 위한 요소를 추가
      event.dataTransfer.addElement(event.currentTarget);
    }, false);
  }
</script>
```



선택 텍스트의 드래그 포인터

- 드래그 중인 마우스 포인터를 변경하기
 - 드래그 앤 드롭 API를 사용하여 페이지의 텍스트를 선택하고 드롭 영역으로 드래그 하면 마우스 포인터에는 그것이 복사인지 이동인지를 나타내는 포인터가 표시된다.
 - 이 포인터는 드롭이 일어날 때 무슨 일이 일어나는지를 사용자가 알 수 있게 해주는 중요한 정보다.
 - 선택한 텍스트를 드래그하여 드롭하는 목적은 복사, 링크, 이동의 3가지 패턴이 규정되어 있다. 또한 각 상황마다 마우스 포인터가 달라진다.
 - 드래그 앤 드롭 API에서는 드롭 중인 마우스 포인터를 용도에 맞춰 변경할 수 있다.

메서드/속성	설명
event.dataTransfer.effectAllowed	드래그 앤 드롭으로 할 수 있는 조작을 나타내는 문자열을 반환한다. 값을 지정해서 작업 변경이 가능하다. 지정 가능한 값은 none, copy, copyLink, copyMove, link, linkMove, move, all, uninitialized 중 하나이다.
event.dataTransfer.dropEffect	드랍될 때 처리 가능한 조작을 나타내는 문자열을 반환한다. 값을 지정하여 가능한 조작 변경이 가능하다. 가능한 값은 none, copy, link, move 이다. 값을 지정하면 해당 조작에 맞는 드래그 포인터가 표시된다.

선택 텍스트의 드래그 포인터

- 드래그 중인 마우스 포인터를 변경하기
 - 일반적으로 `effectAllowed` 속성은 드래그되는 요소에서 발생하는 `dragstart` 이벤트 리스너에서 지정한다.
 - `dropEffect` 속성은 드롭되는 요소에서 발생하는 `dragenter` 이벤트와 `dragover` 이벤트에서 사용된다.
 - `effectAllowed` 속성에서 허용하는 동작만 드랍 측에서 사용가능하다. 따라서 `dropEffect` 속성에는 `effectAllowed` 속성에서 허용하는 작업을 나타내는 값을 지정해야 한다.
 - 이 값이 일치하지 않으면 `none`에 해당하는 드래그 포인터가 표시된다.

선택 텍스트의 드래그 포인터

- 드래그 중인 마우스 포인터를 변경하기
 - effectAllowed 속성 값

값	설명
copy	선택 영역이 복사됨을 의미한다
link	선택 영역이 드롭 영역에 링크됨을 의미한다
move	선택 영역이 드롭 영역에 이동함을 의미한다
copyLink	선택 영역이 복사 또는 링크됨을 의미한다.
copyMove	선택 영역이 복사 또는 이동함을 의미한다.
linkMove	선택 영역이 링크 또는 이동함을 의미한다.
All	모든 드롭 조작이 유효해진다.
none	모든 드롭 조작이 무효화 된다. 드롭 중에는 드롭 불가를 나타내는 커서가 나타난다.
uninitialized	effectAllowed 에 아무것도 정의되지 않은 상태를 나타낸다. (기본값)

선택 텍스트의 드래그 포인터

- 드래그 중인 마우스 포인터를 변경하기
 - `dropEffect` 속성 값

값	설명
copy	복사를 나타내는 포인터가 표시된다
link	링크를 나타내는 포인터가 표시된다
move	이동을 나타내는 포인터가 표시된다.
none	드롭 불가를 나타내는 포인터가 표시된다(기본값)

선택 텍스트의 드래그 포인터

○ 드래그 중인 마우스 포인터를 변경하기 예제 - 1

```
<meta charset="UTF-8">
<title>Document</title>
<style type="text/css">
    #drop {
        border-color: black;
        border-width: 1px;
        border-style: solid;
        height: 100px;
        width: 400px;
    }
</style>
</head>
<body>
    <p id="text">텍스트를 선택해서 아래 영역에 드롭해 주세요. </p>
    <div id="drop"></div>
    <script type="text/javascript">
```

```
/* -----
 * ■드래그 측의 처리
 * ----- */
// p요소에 dragstart 이벤트 리스너 지정
var p = document.querySelector('#text');
p.addEventListener("dragstart", function(event) {
    // 선택한 텍스트 얻기
    var selection = window.getSelection();
    var text = selection.toString();
    // 허용하는 기능
    event.dataTransfer.effectAllowed = "copy";
    // 전송하기 위한 데이터 지정
    event.dataTransfer.setData("text", text);
}, false);
```

선택 텍스트의 드래그 포인터

○ 드래그 중인 마우스 포인터를 변경하기 예제 - 1

텍스트를 선택해서 아래 영역에 드롭해 주세요.

택해서 아래 영역에

```
/* -----  
*   ■ 드롭 처리  
* ----- */  
  
// div요소에 dragenter 이벤트 리스너 지정  
var div = document.querySelector('#drop');  
div.addEventListener("dragenter", function(event) {  
    // 기본 액션 해제  
    event.preventDefault();  
    // 드래그 포인터 지정  
    event.dataTransfer.dropEffect = "copy";  
}, false);  
  
// div요소에 dragover 이벤트 리스너 지정  
div.addEventListener("dragover", function(event) {  
    // 기본 액션 해제  
    event.preventDefault();  
    // 드래그 포인터 지정  
    event.dataTransfer.dropEffect = "copy";  
}, false);  
  
// div 요소에 drop 이벤트 리스너 지정  
div.addEventListener("drop", function(event) {  
    // 기본 액션 해제  
    event.preventDefault();  
    // 선택된 텍스트 얻기  
    var text = event.dataTransfer.getData("text");  
    // 선택된 텍스트를 div 요소에 추가  
    div.appendChild(document.createTextNode(text));  
}, false);
```

파일 드랍

- pc에 있는 파일을 드랍
 - HTML5의 드래그 앤 드롭 API는 PC 파일의 드롭이 가능하도록 해당 기능을 지원한다.

속성	설명
event.dataTransfer.files	바탕화면의 파일이 드롭되면 파일의 목록을 포함하는 FileList 객체를 반환한다. 다만 드래그 중에 파일 데이터를 가질 수 없다.

- DataTransfer 객체의 files 속성은 드랍된 파일 목록을 포함하는 FileList 객체를 반환한다.
- 단 FileList 객체 자체는 HTML5의 명세에 없으며 File API는 명세에 규정된 객체이다.

파일 드랍

○ pc에 있는 파일을 드랍 예제

PC의 파일을 드롭해 주세요



```
<style type="text/css">
  #drop {
    border-color: black;
    border-width: 1px;
    border-style: solid;
    height: 250px;
    width: 450px;
  }
</style>
</head>
<body>
  <p>PC의 파일을 드롭해 주세요</p>
  <div id="drop"></div>
```

```
/* ----- */
*   드롭 측의 처리
* ----- */

// div요소에 dragenter이벤트 리스너 지정
var div = document.querySelector('#drop');
div.addEventListener("dragenter", function(event) {
  // 기본 액션 해제
  event.preventDefault();
}, false);
// div 요소에 dragover 이벤트 리스너 지정
div.addEventListener("dragover", function(event) {
  // 기본 액션 해제
  event.preventDefault();
}, false);
// div 요소에 drop 이벤트 리스너 지정
div.addEventListener("drop", function(event) {
  // 기본 액션 해제
  event.preventDefault();
  // FileList 오브젝트
  var files = event.dataTransfer.files;
  // 드롭된 파일의 File 오브젝트
  var file = files[0];
  if( ! file ) { return; }
  // 파일의 MIME 타입 확인
  if( ! file.type.match(/^image\/) ) { return; }
  // FileReader 오브젝트
  var reader = new FileReader();
  // Data URL 형식으로 파일 데이터 얻기
  reader.readAsDataURL(file);
  // 파일 데이터 읽기를 성공할 때의 처리
  reader.onload = function() {
    // img 요소를 생성하여 div 요소에 추가
    var img = document.createElement("img");
    img.src = reader.result;
    div.appendChild(img);
  };
}, false);
```