

3강. 폼, 드래깅, 텍스트 편집

1. 폼

1.1 form 요소와 input 요소

- 폼은 입력양식이라고도 하며 웹 문서의 사용자 인터페이스(UI : user interface)를 구성 하는데 중요한 역할을 하고 있다.
- 폼은 텍스트 필드나 버튼, 체크박스, 범위 컨트롤, 색상선택기(color picker) 등과 같은 폼 컨트롤이 있는 페이지의 구성요소이다.
- 모든 폼은 form 요소를 사용한다. 이 요소 안에 컨트롤들이 들어가지만, 대부분의 컨트롤은 input 요소를 사용하기 때문에 폼에 많은 기능들이 추가되었다.

예제

```

<h2>피자 주문하기</h2><hr />
<form action="order.html" method="post">
  <p><label>고객 이름: <input type="text" name="custname"> </label></p>
  <p><label>전화 번호: <input type="tel" name="custtel"> </label></p>
  <p><label>메일 주소: <input type="email" name="custemail"> </label></p>
  <fieldset>
    <legend> 피자 크기 </legend>
    <label> <input type="radio" name="size" value="small"> Small </label>
    <label> <input type="radio" name="size" value="medium"> Medium </label>
    <label> <input type="radio" name="size" value="large"> Large </label>
  </fieldset>
  <fieldset>
    <legend> 피자 토핑 고르기 </legend>
    <label> <input type="checkbox" name="topping" value="bacon"> 베이컨
  </label>
    <label> <input type="checkbox" name="topping" value="cheese"> 치즈
  </label>
    <label> <input type="checkbox" name="topping" value="onion"> 양파 </label>
    <label> <input type="checkbox" name="topping" value="mushroom"> 버섯
  </label>
  </fieldset>
  <label>원하는 배달 시간: <input type="time" min="11:00" max="21:00" step="900"
name="delivery"> </label>
  <label>배달 요구 사항: <textarea name="comments"> </textarea> </label>
  <p><button>피자 주문하기</button></p>
</form>


```

fieldset

form요소 안의 요소들을 그룹핑하기 위해서 사용되며 legend는 fieldset의 캡션을 설정합니다.

```

<form>
  <fieldset>
    <legend>학생</legend>
    Name: <input type="text"><br />
    Email: <input type="text"><br />
    Date of birth: <input type="text">
  </fieldset>
</form>
    
```



- input 요소의 type 속성 값의 종류

type 종류	설명
text	사용자가 텍스트를 입력할 수 있는 한 줄짜리 입력 타입
password	비밀번호 입력 (입력 글자가 다른 글자로 대체되어 표시)
radio	라디오 버튼(배타적 선택)
checkbox	체크박스 (0개 이상의 값들의 선택이 가능)
hidden	숨겨진 입력 데이터를 받기 위한 타입
file	파일 업로드를 위한 파일 입력과 선택 필드 정의
image	이미지를 전송 버튼으로 정의
submit	전송 버튼 (폼 데이터를 서버로 전송)
reset	리셋 버튼 (폼의 모든 필드를 초기값으로 초기화)
button	클릭이 가능한 버튼 정의 (주로 자바스크립트를 활성화시키기 위해서 사용)

1.2 HTML5 에서 추가된 폼 요소

- datalist요소
 - ▶ 텍스트박스에 대한 옵션 리스트를 보여주는 것으로 텍스트박스가 포커스를 받으면 하단에 미리 입력한 목록 값을 표시하여 준다.

사용형식

```

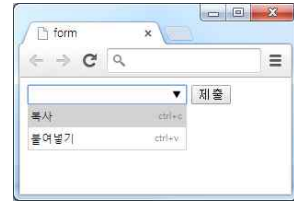
<input type="text" name="이름" list="데이터리스트 아이디">
<datalist id="데이터리스트 아이디">
  <option value="값:서버로 전달되는 항목 값"
    label="값:긴 항목을 짧은 레이블로 표시">
  <option value="값" label="값">
</datalist>
    
```

예제

```

<form action="" method="get">
  <input type="text" list="edit">
    <datalist id="edit">
      <option value="복사" label="ctrl+c" >
      <option value="붙여넣기" label="ctrl+v" >
    </datalist>
  <input type="submit">
</form>

```



포커스를 받았을 때 화면

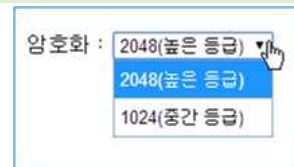
- keygen 요소
 - ▶ 사용자 간의 통신을 할 때 안전하게 데이터 주고 받기 위한 암호 키를 생성하는 역할을 하는 키 쌍 생성 컨트롤(key pair generator control)이다.
 - ▶ 키 쌍을 생성해서 폼이 전송될 때 개인키(private key, 비밀키)는 클라이언트의 저장소에 저장되고, 공개키(public key)는 서버의 응용 프로그램으로 전송된다.
 - ▶ 보안과 관련된 요소로는 다음과 같은 고유한 속성들이 정의되어 있다.

예제

```

<form action="" method="post">
  <p>암호화 : <keygen name="security" /> </p>
</form>

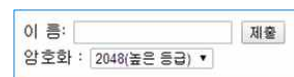
```

**예제**

```

<form action="" method="post" id="security"> ①
  이름: <input type="text" name="u_name">
  <input type="submit">
</form>
암호화 : <keygen name="security" form="security"/> ①

```



- ① keygen을 외부에 지정해도 form 속성을 이용하면 form과 연결되어 사용할 수 있다.

- output 요소
 - ▶ 폼 내의 계산 결과를 출력할 수 있도록 지정하는 요소로써, 폼의 양식 데이터를 입력하고 form의 처리한 결과를 표시할 때 유용하게 사용될 수 있다.
 - ▶ input 요소에 입력된 값이 변하게 되면 그것을 곧바로 반영하여 화면으로 출력한다. form 요소의 oninput 이벤트, output 요소의 onforminput 이벤트(오페라만 지원) 사용할 수 있다.

예제

```
<form action="" method="post"
  oninput="out.value = parseInt(in_a.valueAsNumber) +
  in_b.valueAsNumber">
  A <input name=in_a type=range step=1 value=5> +
  B <input name=in_b type=number step=1 value=7> =
  <output name=out for="in_a in_b"></output>
</form>
```

A + B = 27

range를 이용한
계산 화면

A + B = 44

number를 이용한
계산 화면

- progress 요소
 - ▶ 어떠한 작업의 완료 진행 상황을 나타내는 것으로 파일의 복사나 다운로드 할 때의 진척도를 알려줄 때 많이 사용될 수 있는 유용한 요소이다.
 - ▶ 진척도를 업데이트하지 않으면 정적인 상태만이 표시되기 때문에 자바스크립트와 연동하여 현재 진행 중인 상태 값을 업데이트 시켜 주어야 한다.

속성	설명
value	현재 진행중인 상태의 값으로, 0.0~1.0 사이의 값이나 max 값이 지정되어 있을 경우 0~max 사이의 값이 된다.
max	작업이 전체에서 얼마나 필요한지 나타내는 최대값
min	작업이 전체에서 얼마나 필요한지 나타내는 최소값
position	value값의 현재 위치를 비율로 나타낸다.
labels	요소의 레이블 목록을 제공한다.

예제

현재 다운로드 상태 :

```
<progress min="0" max="100" value="72">72%</progress>
```

현재 다운로드 상태 :

- meter 요소
 - ▶ 일정 범위 안의 측정 값이나 분포 비율 등을 나타낼 때 사용되며 알려진 범위 내에서의 스킨 크기나 분수 값을 가진다.

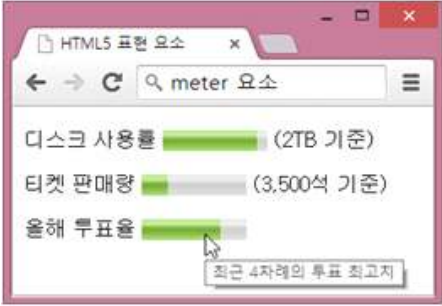
속성	설명
value	실제로 측정된 데이터 값, 반드시 지정해야 한다.
min	값이 지정되었다면 가장 낮은 경계 값이 되고, 지정되지 않으면 0값이 된다.
max	값이 지정되었다면 가장 높은 경계 값이 되고, 값을 지정하지 않으면 1.0값이 된다.
low	허용되는 범위의 최소값을 나타낸다.
high	허용되는 범위의 최대값을 나타낸다.
optimum	최적의 기대치 값을 나타낸다.
labels	요소의 레이블 목록을 제공한다.
title	마우스 오버스 툴 팁을 제공한다.

- ▶ meter요소의 범위를 "상", "중", "하" 세부분으로 나눌 때 high, low, optimum 속성은 각각 "상", "하", "최적"의 값을 의미한다. 만일, optimum값이 high보다 크다면 value값이 클수록 더 좋다는 뜻이고 low값보다 작다면 value값이 작을수록 더 좋다는 것을 의미한다.

예제

```

<p>디스크 사용률 <meter max=100 value=90></meter> (2TB 기준)</p>
<p>티켓 판매량 <meter min=0 max=100 value=25></meter> (3,500석 기준)</p>
<p>올해 투표율 <meter value=0.75 title="최근 4차례의 투표 최고치"></meter></p>
    
```



1.3 HTML5 에서 추가된 폼 속성

- autocomplete 속성
 - ▶ input 요소의 자동 완성 상태를 지정하기 위한 속성으로 2개의 상태를 지정할 수 있다.
 - ▶ on 속성 값은 폼 내의 input요소가 자신의 결과로 나타내는 자동완성 상태를 on으로 설정할 것임을 기본적으로 나타내기 때문에 input요소별로 autocomplete속성을 별도로 지정하지 않아도 된다. off속성 값은 자동 완성 상태를 off로 설정한다.

예제

```
<form action="" autocomplete="on">
  이름: <input type="text" name="name"><br>
  닉네임: <input type="text" name="nickname"><br>
  이메일: <input type="email" name="email" autocomplete="off"><br>
  <input type="submit">
</form>
```

- novalidate 속성
 - ▶ 폼 제출시 양식 입력 데이터의 유효성을 검사하지 않도록 지정하는 속성이다.

1.4 HTML5 에서 추가된 입력 타입

- tel 타입, search 타입
 - ▶ tel 타입은 사용자로부터 전화번호 데이터를 입력받기 위한 타입으로, 텍스트 필드와 동일하지만 줄바꿈 문자를 포함하지 않아야 한다.
 - ▶ search 타입은 사용자로부터 검색어를 입력받기 위한 타입으로 텍스트 필드와 동일하지만 검색이 동반된다.

예제

```
전화번호: <input type="tel" name="tel">①
검색 문자열:
<input type="search" name="search"
value="HTML5">②
```

① tel은 검색하지 않는다.
 ② 검색 후 value속성을 통해 초기값을 지정한다.

- url 타입, email 타입
 - ▶ url 타입은 절대경로의 URL 주소 입력을 위한 타입으로 유효성을 점검해주는 역할을 한다. URL 주소가 형식에 맞게 입력되지 않으면 전송 버튼을 클릭했을 때 경고 메시지가 표시되고 값은 전송되지 않는다.
 - ▶ email 타입은 사용자가 입력한 데이터가 이메일 형식에 맞게 입력했는지를 자동으로 유효성 검사를 해준다.

예제

홈페이지(URL):<input type="url" name="url">
 메일 주소(email):<input type="email" name="email">

type을 url로
지정한 화면

type을 email로
지정한 화면

- number 타입, range 타입
 - ▶ number타입은 사용자로부터 한 줄의 숫자 데이터를 입력 받기 위한 타입으로, 지정된 특성 범위의 숫자만 입력 할 수 있는 것이 특징이다.
 - ▶ range 타입은 사용자로부터 특정 범위의 숫자 데이터를 슬라이더 형태로 입력받기 위한 타입으로 슬라이더를 움직여서 값을 선택한다.

예제

<input type="number" min="1" max="11" step="2" value="5">
 <input type="range" min="1" max="11" step="2" value="5">

number 타입: 5 직접 입력 또는 스펠 버튼 이용
 range 타입: 6단계 동작(1,3,5,7,9,11)

number 타입: 15 max값이 초과된
화면
 range 타입: 값은 11 이하여야 합니다.

number 타입: 8 step(간격)이 잘못
입력된 화면
 range 타입: 유효한 값을 입력해 주세요. 가장 근접한 유효 값 2개는 7 및 9입니다.

- color 타입
 - ▶ 사용자로부터 색상을 입력 받기 위한 타입으로, 다양한 색상표가 생성되고 간단히 색상을 선택함으로써 색상을 입력할 수 있다.

예제

`<input type="color" name="color" value="#000000">`

참조사이트

브라우저에 따라 요소를 지원하지 않는 경우도 있으므로 <http://www.w3schools.com/> 에서 지원여부를 확인 할 수 있다.

Example

```

<form>
  Select your favorite color:
  <input type="color" name="favcolor">
</form>

```

Try it Yourself >>

브라우저별 color 요소 지원 여부

- 날짜 / 시간 입력 타입
 - ▶ 사용자로부터 다양한 형태의 달력을 생성하기 위한 타입으로, 기존 자바스크립트를 사용해서 개발했던 달력을 손쉽게 만들 수 있게 제공하는 입력 타입이다.

속성	설 명	
date	년, 월, 일 단위로 선택할 수 있다.	<code><input type="date" ...></code>
month	년, 월 단위로 선택할 수 있다.	<code><input type="month" ...></code>
week	년, 월, 주 단위로 선택할 수 있고 달력의 왼쪽에 주간이 표시되는 것을 확인 할 수 있다.	<code><input type="week" ...></code>
time	시간(시,분,초) 입력할 수 있는 타입으로 달력은 표시되지 않고 시간을 입력하거나 스피너 버튼을 클릭해서 입력할 수 있다.	<code><input type="time" ...></code>
datetime-local	지역 시간대를 기준으로 년, 월, 일, 시간을 선택할 수 있다.	<code><input type="datetime-local" ...></code>
datetime	UTC 시간대를 기준으로 년, 월, 일, 시간을 선택할 수 있고, 한국과는 9시간 차이가 나기 때문에 한국 시간을 사용하려면 datetime-local 타입을 사용해야 한다.	<code><input type="datetime" ...></code>

1.5 HTML5에서 추가된 input 요소의 속성

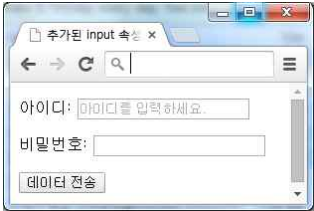
- required 속성
 - ▶ 필수 입력항목을 지정하는 것으로, 회원 가입 같은 양식에서 이름이나 주소 항목에는 반드시 입력해야 한다면 이러한 항목들에 required 속성을 지정할 수 있다.
- minlength 속성 (주요 브라우저 미지원)
 - ▶ 지정한 입력 항목의 최소로 입력할 수 있는 길이를 지정하는 속성이다. 지정한 입력 항목의 길이가 최소로 입력할 수 있도록 제어한다.
- placeholder 속성
 - ▶ 입력 타입에 기본 텍스트(미리 참조할 문자열)를 지정하도록 하는 것으로 사용자의 데이터 입력을 도와주는 짧은 힌트를 지정할 때 사용된다.

예제

```
<form>
  <p>아이디: <input type=text placeholder="아이디를 입력하세요."> </p> ①
  <p>비밀번호: <input type=password minlength=12 required></p> ②
  <p><input type="submit" value="데이터 전송"></p>
</form>
```

① placeholder 속성으로 기본텍스트를 지정하였다.

② type속성은 password 최소 입력길이는 12글자로 반드시 값을 입력해야한다. 입력하지 않을 경우 입력하라는 메시지가 표시된다.



첫화면

- multiple 속성
 - ▶ 사용자가 하나 이상의 값을 지정할 수 있는지를 지정하는 것으로 이메일 주소록에 하나 이상의 주소를 입력할 때 유용하게 사용할 수 있는 속성이다.
- min과 max속성
 - ▶ 요소에 허용된 값의 범위 지정해 주는 속성으로 max(최대치)는 min(최소치)보다 작아서는 안된다.
- step 속성
 - ▶ 요소에 허용된 값을 제한하여 값에 대한 기대되는(필수적인) 단계를 지정해 주는 속성이다. 요소에 정해진 최소치 또는 최대치가 있다면, 이 요소의 범위 안에서 값이 정해진다. step값은 input요소의 type값에 따라서 단위가 달라진다.

예제

```
파일첨부: <input type=file name=att multiple accept="image/jpeg, image/png">
(90년대생)생년월일: <input type=date name=bday
                    min="1990-01-01" max="1999-12-31">
0과 1사이(간격: 0.05): <input type=number name=opacity
                    min=0 max=1 step=0.05>
임의 시간 선택: <input type=time name=favtime step=any>
```

<p>파일첨부: <input type="button" value="파일 선택"/> 파일 5개</p> <p>(90년대생)생년월일: 연도-월 <input type="text" value="연도-월"/></p> <p>0과 1사이(간격: 0.05): <input type="text" value="0.05"/></p> <p>임의 시간 선택: --:--:-- <input type="button" value="데이터 전송"/></p>	<p>파일선택 버튼을 클릭하여 파일을 선택하고 나면 옆에 파일이름이 표시/ 2개 이상의 파일을 선택하면 몇 개의 파일이 선택되었는지 표시한다.</p>
<p>파일첨부: <input type="button" value="파일 선택"/> 파일 5개</p> <p>(90년대생)생년월일: 1999-08-18 x</p> <p>0과 1사이(간격: 0.05): <input type="text" value="0.05"/> ! 값은 1990-01-01 이후여야 합니다.</p> <p>임의 시간 선택: --:--:-- <input type="button" value="데이터 전송"/></p>	<p>min속성 값보다 크고 max속성 값보다 작은 값을 입력해야 한다.</p>
<p>파일첨부: <input type="button" value="파일 선택"/> 파일 5개</p> <p>(90년대생)생년월일: 1999-08-18</p> <p>0과 1사이(간격: 0.05): 0.2</p> <p>임의 시간 선택: --:--:-- ! 유효한 값을 입력해 주세요. 가장 근접한 유효 값 2개는 0.2 및 0.25 입니다.</p>	<p>step값을 잘못 입력한 화면</p>
<p>파일첨부: <input type="button" value="파일 선택"/> 파일 5개</p> <p>(90년대생)생년월일: 1999-08-18</p> <p>0과 1사이(간격: 0.05): 0.25</p> <p>임의 시간 선택: 오전 08:55 x <input type="button" value="데이터 전송"/></p>	<p>임의 시간 선택의 step 값을 any로 지정</p>

- list 속성
 - ▶ 사용자에게 제시된 미리 정의된 선택항목을 갖고 있는 요소를 구별한다. 그 값이 정의되어 있는 경우, datalist요소의 아이디(id)여야 한다.
- pattern속성
 - ▶ 유효성 검사를 위한 정규 표현식을 지정할 수 있도록 하는 것으로, 자동으로 사용자가 입력한 값에 대하여 유효성 검사가 진행되고 값에 따라 오류 메시지가 출력된다.

정규 표현식(regular expression)
<p>정규 표현식은 특정한 규칙을 가진 문자열들을 표현하는 데 사용하는 언어로, 주로 문자열의 검색이나 유효성 검사에 사용이 된다.</p> <p>^ : 문자열의 처음을 의미</p> <p>? : 0 또는 1개를 의미</p> <p>[a-z] : 문자열의 범위 a에서 z까지 의미</p> <p>\$: 문자열의 끝부분을 의미</p> <p>[0-9] : 0부터 9까지의 숫자를 의미</p> <p>{n} : n개의 문자를 의미</p> <p>wd : 숫자로 시작(0부터 9까지 숫자를 의미)</p>

이름	정규식 표현의 예
이름	[a-zA-Z] {4,10} : 알파벳문자를 4개~10개만 입력받는 표현
전화번호	0Wd {1,2 }W-Wd {3,4 }W-Wd {4 }
주민등록번호	Wd {6 }W-[1-4]Wd {6 }
이메일	^[a-zA-Z0-9]+@[a-zA-Z0-9]+\$

예제

전화번호:
`<input type="text" name=tel pattern="0 Wd {1,2 } W- Wd {3,4 } W- Wd {3,4 }" title="올바른 형식이 아닙니다.">`

주요 브라우저:
`<input type=text list="browsers">`
`<datalist id="browsers">`
`<option value="Internet Explorer">`
`<option value="Firefox">`
`<option value="Google Chrome">`
`<option value="Opera">`
`<option value="Safari">`
`</datalist>`

전화번호:

주요 브라우저: ! 요청한 형식과 일치시키세요. 올바른 형식이 아닙니다.

전화번호가 pattern 형식과 다른 경우

전화번호:

주요 브라우저: ▼

Internet Explorer
 Firefox
 Google Chrome
 Opera
 Safari

list요소 적용화면

- autofocus속성
 - ▶ 웹 문서의 로드가 완료됨과 동시에 지정한 입력 타입에 자동으로 포커스를 두도록 한다. 이러한 기능은 검색 포털 사이트 방문시 검색창으로 포커스가 자동으로 이동되게 하거나, 로그인을 하는 부분에서는 아이디를 입력 받는 부분에 자동으로 포커스를 이동시키는 등에 주로 사용이 된다.
 - ▶ 만일 여러개의 입력 양식에서 autofocus 속성을 지정하면, 마지막으로 지정한 항목에 포커스가 위치하게 된다.
- autocomplete속성
 - ▶ 지정한 입력 항목은 값을 입력한 후에 다른 웹 사이트나 페이지로 이동했다가 다시 돌아오더라도 값이 그대로 채워져 있도록 해준다. 웹 사이트에서 정보들을 입력하고 전송 버튼을 클릭했을 때, 전송 실패 시에도 입력한 정보가 사라지지 않고 같은

내용을 입력할 때 입력한 정보를 다시 그대로 표시하여 입력의 편리함을 제공한다.

- form속성
 - ▶ 해당 input 요소를 특정 form 요소와의 연결을 지정하는 것으로, input 요소가 form 요소 안에 있지 않아도 form 속성으로 연결 할 수 있다. 특정 form 요소를 지정하기 위해서는 form요소의 id 속성 값과 동일해야 한다.

예제

```
<form id="submit">
  과목:<input type="text" name="data" autofocus autocomplete="on">
</form>
<input type="submit" form="submit">
```

- dirname속성
 - ▶ 폼을 제출할 때 이 속성을 사용한 요소의 방향성을 지정하는 것으로 폼 제출시에 이 값을 포함하는 필드의 이름이 함께 제출된다. 만일 이 속성을 지정하면 속성 값은 빈 문자열이 아니어야 한다.


예제

```
<form action="add_comment.php" method="post">
  <p>댓글 : <input type="text" name="comment" dirname="comment.dir"
required> </p>
  <p><button name="mode" type="submit" value="add">댓글 달기</button> </p>
</form>
```

- checkbox 타입의 indeterminate속성
 - ▶ 하위의 체크 박스 전부가 아닌 일부만 선택되었을 때 보여주는 '미확정 선택'상태를 지정하는 속성이다. 이 속성 값을 true로 지정하면 체크박스가 미확정 상태로 변경되면서 '-' 가 표시되게 된다.
- image 타입의 width 및 height속성
 - ▶ input 요소 type="image"의 폭과 높이를 픽셀값으로 지정하는 것으로 항상 폭과 높이를 함께 지정해야 한다.

예제

```
<input type="checkbox" onchange="this.indeterminate=!this.indeterminate;">
미확정 상태
<input type="image" src="HTML5_logo1.jpg" width="50" height="50">
```



- form 요소의 속성 확장
 - ▶ submit 및 image 입력 타입에 사용할 수 있도록 대체 속성의 추가되었다.

속성	Type	설 명
formaction	submit image	폼을 전송해서 처리할 프로그램이 있는 URL을 지정하되, 앞뒤로 공백을 허용하는 URL이어야한다.
formenctype	submit image	제출되는 폼 데이터를 인코딩하기 위한 MIME유형을 지정한다.
formmethod	submit image	form요소의 method 속성을 대체하는 속성으로 action 속성의 URL로 폼을 전송하기 위한 방법을 지정한다.
formtarget	submit image	form요소의 target 속성을 대체하는 속성이다.
formnovalidate		form요소의 novalidate 속성을 대체하는 속성으로 'true'값이 지정되면 폼 제출시 input요소에 대한 값을 검증하지 않는다.

2. 드래깅

2.1 드래그 앤 드롭

- drag & drop은 마우스를 이용하여 어떠한 콘텐츠를 끌어다 놓는 것이라 한다.
- 처리 과정
 1. 드래그 할 요소에 드래그 속성 지정한다. 요소가 드래깅 되기 위해서는 전역 draggable 속성을 이용하여 해당 요소에 (draggable=true)와 같이 지정하면 이 요소는 다른 위치로 이동될 수 있다.
 2. 요소와 함께 이동할 자료를 결정한다. 어떤 요소를 전달할 것인지는 드래그의 시작 단계에서 결정해야 하기 때문에 dragstart에 대한 이벤트 리스너를 설정해야 한다.
 3. 드롭할 수 있는 대상 요소 지정한다. 이동한 데이터를 드롭하기 위해서는 드롭 대상이 dropzone 속성을 갖고 이벤트를 기다려야 한다.

4. 드래그 & 드롭 동작이 끝나면 해당 자료를 추출한다. 전달된 데이터를 추출하기 위해서는 drop 이벤트에서 해당 자료를 추출하게 되는데, 전달된 datatransfer 객체의 getData()메서드를 이용하여 원하는 정보를 추출하면 된다.

2.2 draggable 속성

- 특정요소를 다른 곳으로 이동시키기 위해 사용할 수 있는 속성이다. 단순히 드래그를 할 수 있도록 하는 true값과 드래그를 할 수 없도록 하는 false값을 가질 수 있다.
- img요소와 href 속성이 있는 a요소는 draggable 속성을 지정하지 않아도 기본적으로 드래깅이 되기 때문에 드래깅이 되지 않게 하기 위해서는 이 속성 값을 false로 지정해야 한다.

예제

```

<a href="http://cs.knou.ac.kr/" draggable="true">방송대 컴퓨터과학과</a>
```



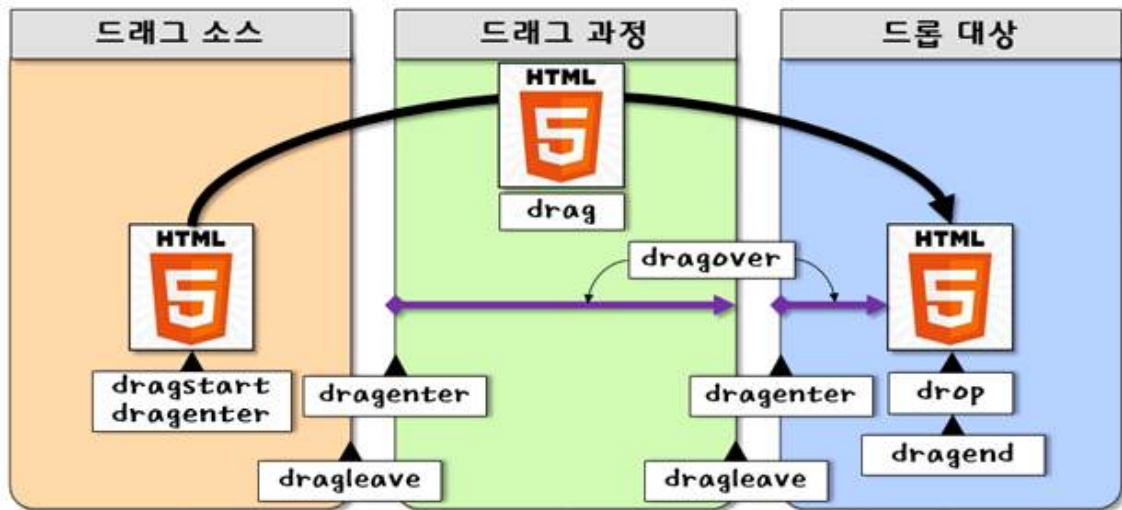
HTML5 이미지를 클릭한 상태에서 마우스를 오른쪽으로 나타낸 화면

2.3 dataTransfer 객체

- 드래그 앤 드롭과 관련된 이벤트들을 제어하기 위해서는 드래그 되는 대상의 요소 안에 속성을 지정하여 이벤트에서 처리할 메서드를 지정하고 이 메서드 안에서는 드래그 앤 드롭으로 전달되는 데이터에 대한 처리를 할 때 사용된다.
- 드래그 앤 드롭 이벤트가 발생하는 동안 사용되며, 그 이벤트가 발생하는 동안만 유효하다.
- 드래그 하는 동안의 시각적 효과를 내기 위한 것으로 드래그 하는 동안의 마우스 포인터 모양을 변경할 수 있다.

2.4 드래그 이벤트 처리

- 드래그 앤 드롭 기능을 위해서는 이벤트 지정과 컨트롤이 필요하다.



이벤트의 종류와 발생 시점을 표현

이벤트	설명
dragstart	드래그를 시작하였을 때 발생하며, 전달할 데이터 저장한다.
drag	드래그 중에 계속 발생하지만 반드시 수행해야 하는 작업은 없다.
dragend	드래그 종료했을 때 발생한다.
dragenter	드래그 요소가 다른 요소 범위 안에 들어올 때 발생한다.
dragleave	드래그 요소가 요소를 벗어날 때 발생 한다.
dragover	드래그 요소가 다른 요소 위에서 드래그하고 있을 때 또는 드롭 직전에 발생한다. preventDefault(): 기본값을 취소해서 드롭이 허용되도록 지정
drop	드롭하였을 때 발생한다. 전달된 데이터를 가져와서 드롭 요소에 추가하고 기본 값 처리를 무효로 처리하도록 설정하기 위해 preventDefault()를 호출한다.

```

예제
<style> /* css 지정 */
  div {width: 100px; height: 100px; border: 1px solid; float:left; margin-right:10px }
</style>
<script>
/*기본 값을 취소해서 드롭이 허용되도록 지정*/
function dragover(event) {
  event.preventDefault(); }
function dragstart(event) {
  event.dataTransfer.setData("Text", event.target.id); }
    
```

format

드래그 되는 요소의 id

```

function drop(event) {
    var data = event.dataTransfer.getData("Text"); // 드래그 된 요소 ID
    event.target.appendChild(document.getElementById(data));
    // 드래그된 요소를 드롭 요소에 추가
    event.preventDefault();
}
</script>

<body>
<div id="drag_source" ondrop="drop(event)" ondragover="dragover(event)">
    
</div>
<div id="drag_target" ondrop="drop(event)" ondragover="dragover(event)"> </div>
</body>
    
```

2.6 드래그 포인터 변경

- effectAllowed 속성 및 dropEffect 속성을 이용하여 사용자가 포인터의 모양을 지정할 수 있다.

예제

effectAllowed 속성 ⇨ 드래그시 발생하는 dragstart 이벤트에서 지정

```
function dragstart(event) { //ondragstart="dragstart(event)"

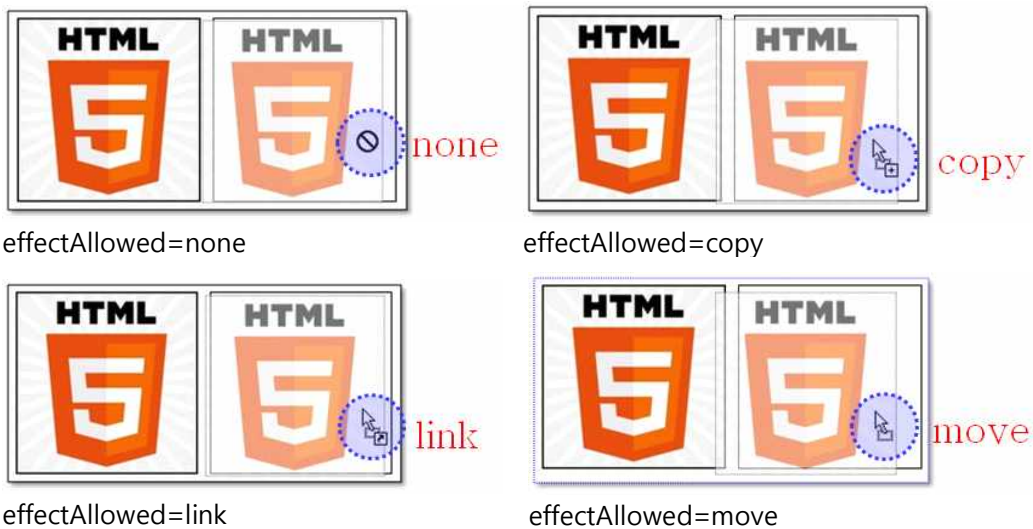
    event.dataTransfer.setData("Text", event.target.id);
    event.dataTransfer.effectAllowed = "copy";
}
```

dropEffect 속성 ⇨ 드롭할 때 발생하는 dragenter 이벤트와 dragover 이벤트에서 사용

```
function dragover(event) { // ondragover="dragover(event)"

    event.preventDefault();
    event.dataTransfer.dropEffect = "copy";
}
```

effectAllowed="copy" 지정하고 dropEffect="move" 지정하면 none에 해당되기 때문에 지정한 영역으로 요소를 드롭할 수 없게 된다.




2.7 드래그 이미지 변경

- 데이터전송 객체 setDragImage(element, x, y)메서드를 사용하여 드래그 할 때의 마우스 커서 위에 표시할 이미지를 변경 할 수 있도록 하고 있다.

예제

```
function dragstart(event) {
    event.dataTransfer.setData("Text", event.target.id);

    var icon = document.createElement("img");
    icon.src = "mini.jpg";
    event.dataTransfer.setDragImage( icon, 0, 0 );
}
```



3. 텍스트 편집

3.1 텍스트 편집 속성

- 텍스트를 편집이 불가능한 요소를 편집이 가능하기 위한 방법으로는 contenteditable 속성 designMode 속성을 이용하는 방법이 있다.
- contenteditable 속성 (문서 영역 편집)
 - ▶ 웹 문서의 특정 요소만을 편집이 가능하도록 하기 위해서 이용한다.

속성	설 명
contenteditable	특정 요소의 내용을 편집할 수 있도록 문자열 형식으로 지정한다. •true → 편집 가능하도록 한다. •false → 편집 불가능하도록 한다. •inherit → 부모 요소로부터 값을 물려받는다.

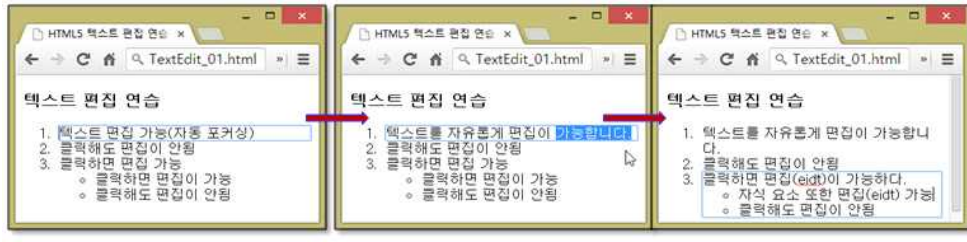
예제

```
<script>
    window.onload = function() {
        var editor = document.getElementById("edit1");
        if ( editor.isContentEditable ) { editor.focus(); }
    };
</script>
<ol>
```

```

<li id="edit1" contenteditable>텍스트 편집 가능(자동 포커싱)</li>
<li id="edit2">클릭해도 편집이 안됨</li>
<li id="edit3" contenteditable>클릭하면 편집 가능
  <ul>
    <li id="edit4" spellcheck=false>클릭하면 편집이 가능</li>
    <li id="edit5" contenteditable=false>클릭해도 편집이 안됨</li>
  </ul>
</li>
</ol>

```



- designMode 속성 (문서 전체 편집)
 - ▶ 웹 문서를 전체를 편집하고자 할 때 사용한다. 문서의 현재 상태를 바꿀 수 있도록 설정할 수 있기 때문에 문서가 편집 가능하도록 하기 위해서는 "on"을 지정하고 그렇지 않으면 "off"를 지정하면 된다.

예제

```

<script>
  window.onload = function() { document.designMode = "on"; };
</script>

<h3>웹 문서 편집 연습</h3>
<div>웹 문서의 편집이 가능합니다</div>

```

HTML5 텍스트 편집 연습

웹 문서 편집 연습

웹 문서의 편집이 가능합니다

편집전

HTML5 텍스트 편집 연습

제목을 편집하고 있습니다.

designMode 속성: 웹 문서의 편집이 가능합니다

편집후

3.2 텍스트 편집 API

- execCommand() 메서드를 사용하면 텍스트를 다양하게 편집을 할 수 있게 된다.

- 사용방법

```
document.execCommand(commandId [, showUI [, value ]])
```

선택한 텍스트 영역에 대한 명령어 ←

지정한 commandId에 대한 UI 표시 여부 ←

지정한 commandId가 필요로 하는 추가 정보 지정 ←

```
document.execCommand("bold", false, false)
```

예제

```
<head>
<style> ① style 적용
    div { width: 700px; height: 200px; border: 1px solid;
          float: left; margin-right:10px; }
    input { width: 100px; height: 30px; }
</style>
<script type="text/javascript">
window.onload = function() ② html 문서가 로드될 때 호출되는 함수
{
    var editor = document.getElementById("editor"); ③ id=edior를 찾아 변수에 저장
    if (editor.isContentEditable) ④editor가 편집가능하면
    {
        editor.focus(); ⑤ editor에 포커스가 지정된다.
    }
};
function setBold()
{
    document.execCommand("bold"); ⑥ 텍스트를 진하게
}
function setItalic()
{
    document.execCommand("italic");⑥ 이탤릭체
}
function setSubscript()
{
    document.execCommand("subscript"); ⑥ 아래첨자
}
function setSuperscript()
{
    document.execCommand("superscript");⑥ 윗첨자
}
}
```

```

function setSelectAll()
{
    document.execCommand("selectAll"); ⑥ 모든 텍스트 선택
}
function setUnselect()
{
    document.execCommand("unselect");⑥ 선택취소
}
function setUndo()
{
    document.execCommand("undo"); ⑥ 실행취소
}
</script>
</head>
<body>⑦ button을 클릭하면 setBold() 함수(각 함수)를 호출한다.
    <input type="button" value="굵게" onclick="setBold();">
    <input type="button" value="기울게" onclick="setItalic();">
    <input type="button" value="아래첨자" onclick="setSubscript();">
    <input type="button" value="윗첨자" onclick="setSuperscript();">
    <input type="button" value="모두 선택" onclick="setSelectAll();">
    <input type="button" value="선택 취소" onclick="setUnselect();">
    <input type="button" value="되돌리기" onclick="setUndo();"><br><br>
    <div id="editor" contenteditable></div>
</body>

```

