11 JS 이벤트

*참고 w3school

1) 이벤트 참고

[JAVASCRIPT] > [JS Events]

https://www.w3schools.com/js/js_events.asp

[HTML DOM] > [DOM Events]

https://www.w3schools.com/jsref/dom_obj_event.asp

https://www.w3schools.com/js/js_htmldom_events.asp

2) File API 참고

JS HTML 이벤트(1/4)

- 마우스를 클릭 하거나, 키보드의 키를 누르거나, HTML 요소를 선택하거나 등 여러 경우에 대한 이벤트가 존재함

이벤트명	이벤트 설명	사용 예시
onload	객체가 로드 되었을 때 발생	<body onload="proc()”>
onclick	마우스로 요소를 클릭했을 경우	<img onclick="prorc()"
ondbclick	마우스로 요소를 연속 클릭했을 경우	<img ondbclick="prorc()"
onmouseover	마우스를 요소 위에 올렸을 경우	<img onmouseover="prorc()"
onmousemove	마우스 요소 위에 움직이는 경우	
onmouseout	마우스 요소 위에 있다가 벗어난 경우	<img onmouseout="prorc()">

  - onload는 웹 페이지의 모든 컨텐츠가 완전히 로드 된 후 수행. 스크립트를 수행하기 위해 가장 자주 사용하는 이벤트임.

JS HTML 이벤트(2/4)

이벤트명	이벤트 설명	사용 예시
onkeydown	키보드의 키를 눌렀을 때 발생(누르고 있을 때 한번만 실행)	<input type="text" onkeydown="prorc()">
onkeypress	키보드의 키를 눌렀을 때 발생(누르고 있을 때 계속 실행)	<input type="text" onkeypress="prorc()">
onkeyup	키보드의 키를 눌렀다가 뗐을 때 발생	<input type="text" onkeyup="prorc()">
onfocus	요소에 입력 커서가 왔을 때 발생(포커스를 받았을 때)	<input type="text" onfocus="proc()">
onblur	입력 커서가 요소에서 나갔을 때 발생(포커스를 잃었을 때)	<input type="text" onblur="proc()">
onchange	요소의 값이 변경되었을 때 발생	<select onchange="proc()“>

- keyDown, keyPressed, keyUp 순으로 이벤트가 발생 함

JS HTML 이벤트(3/4)

- HTML 요소에 이벤트 할당하는 방법
  - HTML 태그에서 이벤트 할당

```
<button onclick="proc()">클릭</button>
```

  - 스크립트에서 요소에 이벤트 할당

```
<script>
document.getElementById("myBtn").onclick = proc;
</script>
```

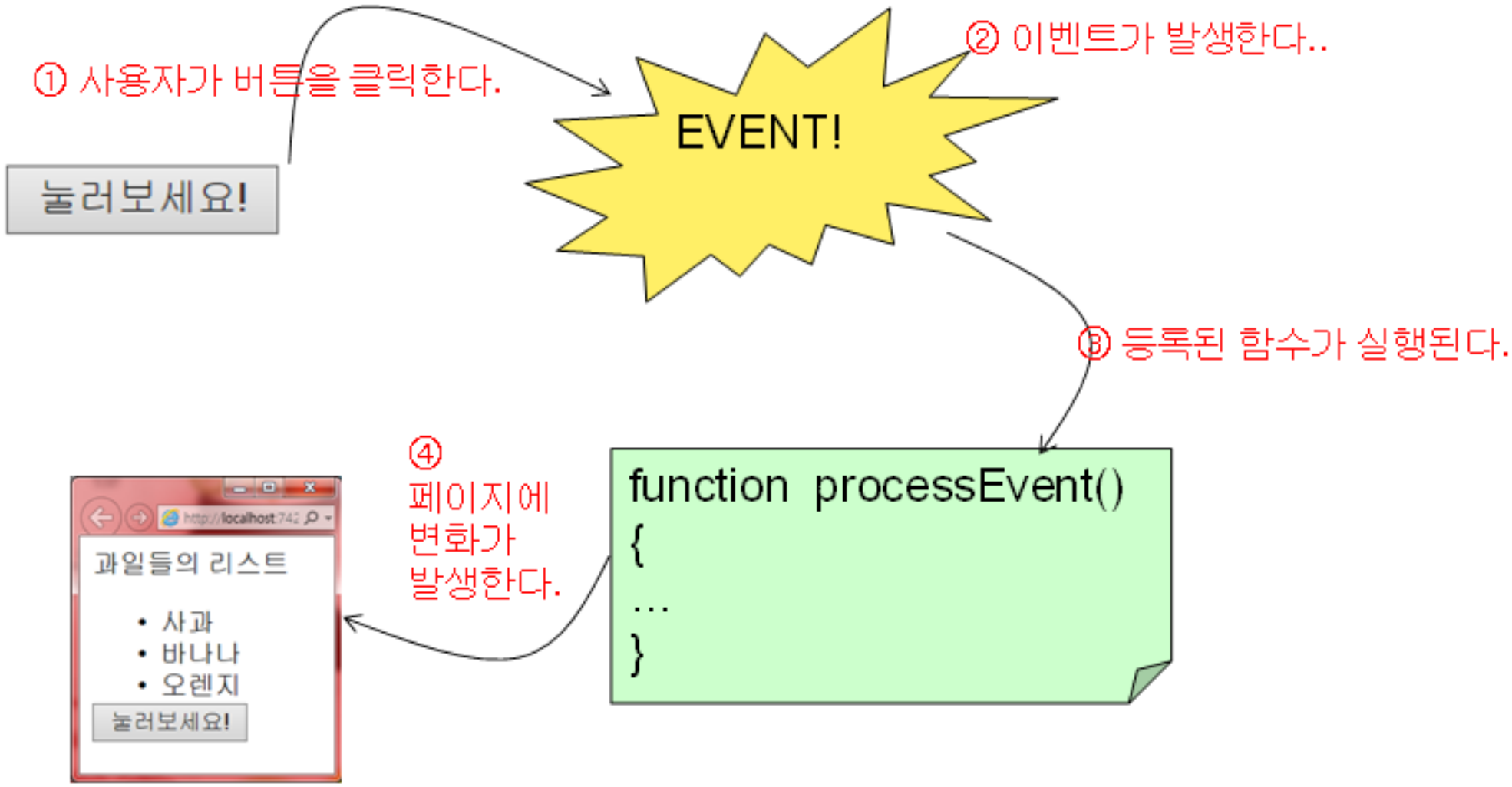
수행 될 함수 명

  - 스크립트에서 요소에 이벤트 리스너를 추가해서 이벤트 할당

```
document.getElementById("myBtn").addEventListener("click",
proc);
```

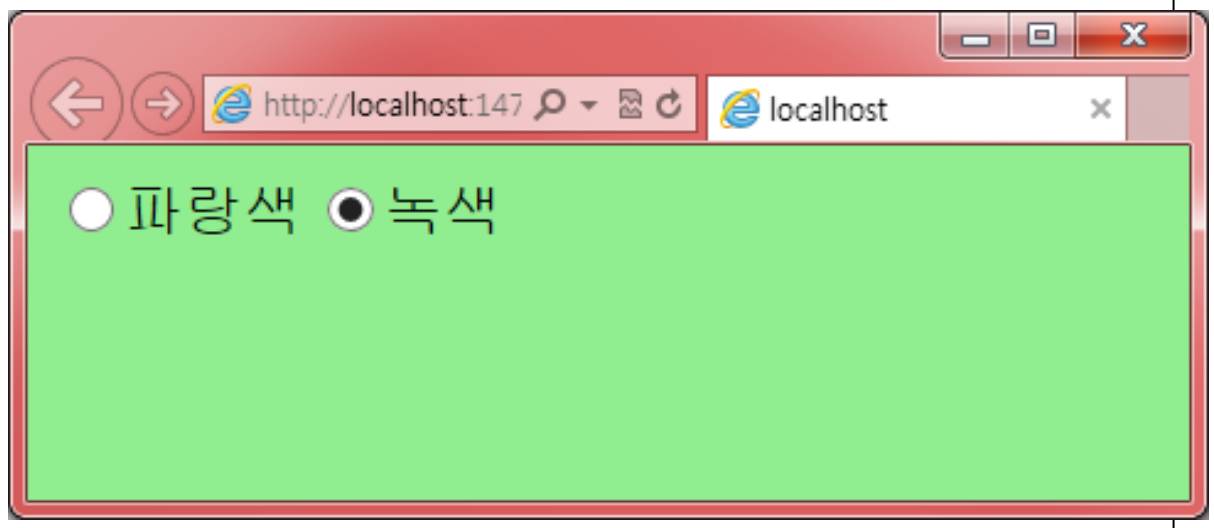
*removeEventListener() 메서드를 사용해서 이벤트 제거 가능

JS HTML 이벤트(4/4)



onclick 이벤트

```
<script>
  function changeColor(c) {
    document.getElementById("target").style.backgroundColor
    = c;
  }
</script>
```

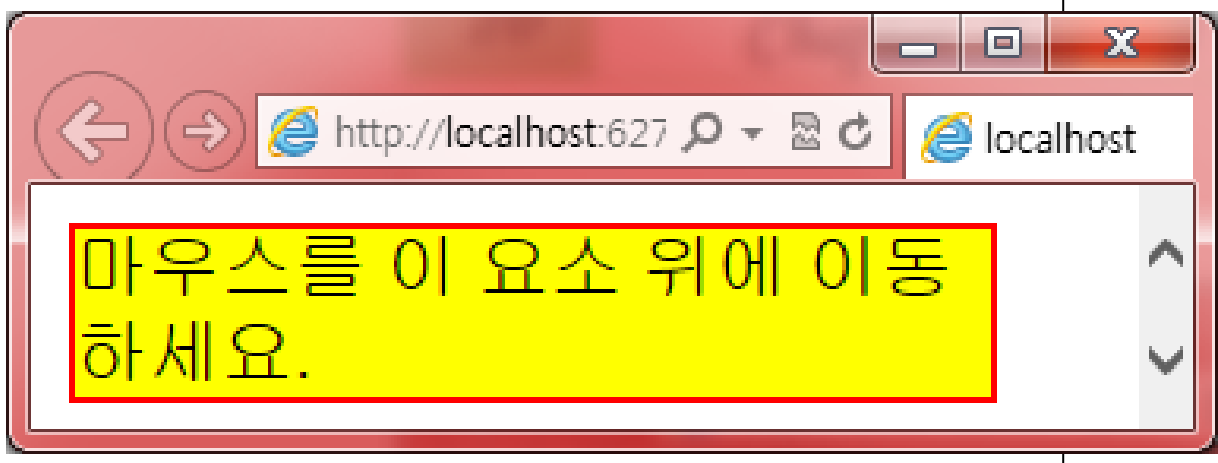


…

```
<body id="target">
  <form method="POST">
    <input type="radio" name="C1" value="v1"
           onclick="changeColor('lightblue')">파랑색
    <input type="radio" name="C1" value="v2"
           onclick="changeColor('lightgreen')">녹색
  </form>
</body>
```

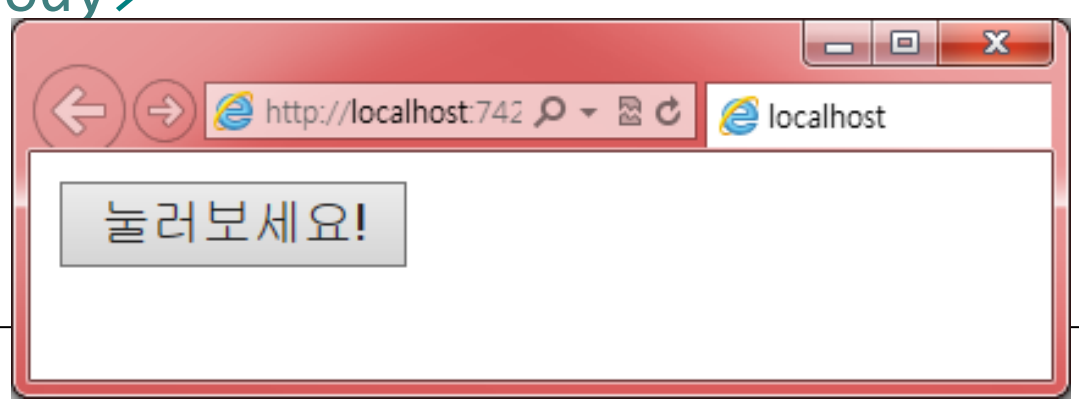
onmouseover 이벤트

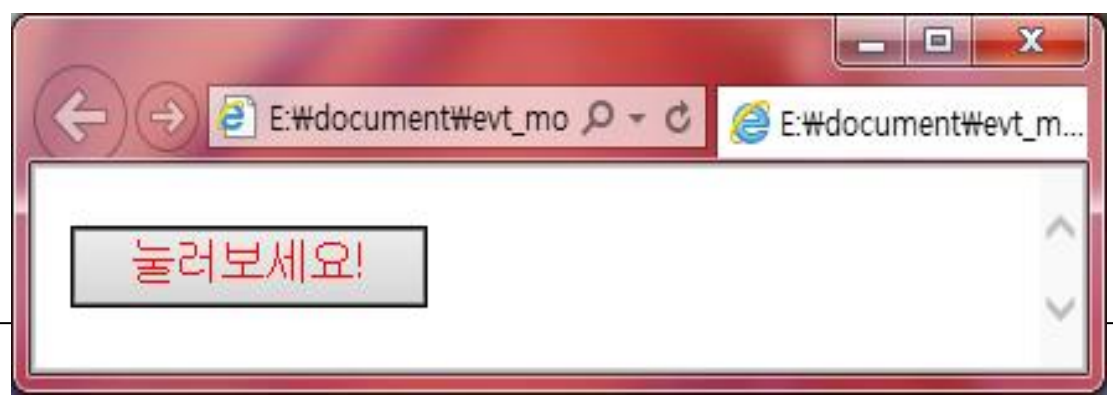
```
<script>
  function setBorder(ele) {
    ele.style.border = "2px solid red";
  }
  function removeBorder(ele) {
    ele.style.border = "";
  }
</script>

…

<body>
  <div style="background-color: yellow; width: 200px;"
        onmouseover="setBorder(this)"
        onmouseout="removeBorder(this)">
       마우스를 이 요소 위로 이동하세요.
  </div>
</body>
```



onmousedown 이벤트

```
<script>
  function changeColor1(button) {
    button.style.color = "#ff0000";
  }
  function changeColor2(button) {
    button.style.color = "#000000";
  }
</script>
…
<body>
  <button onmousedown="changeColor1(this)" onmouseup="changeColor2(this)">
  눌러보세요!
  </button>
</body>
```

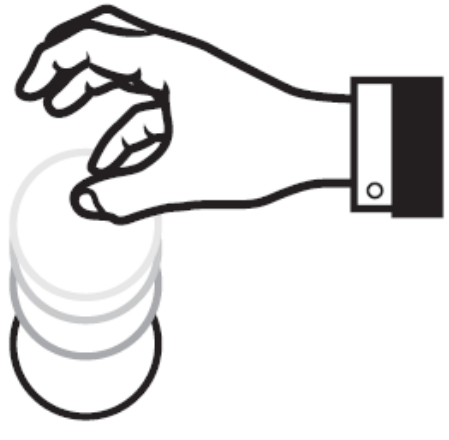




Drag&Drop(1/4)

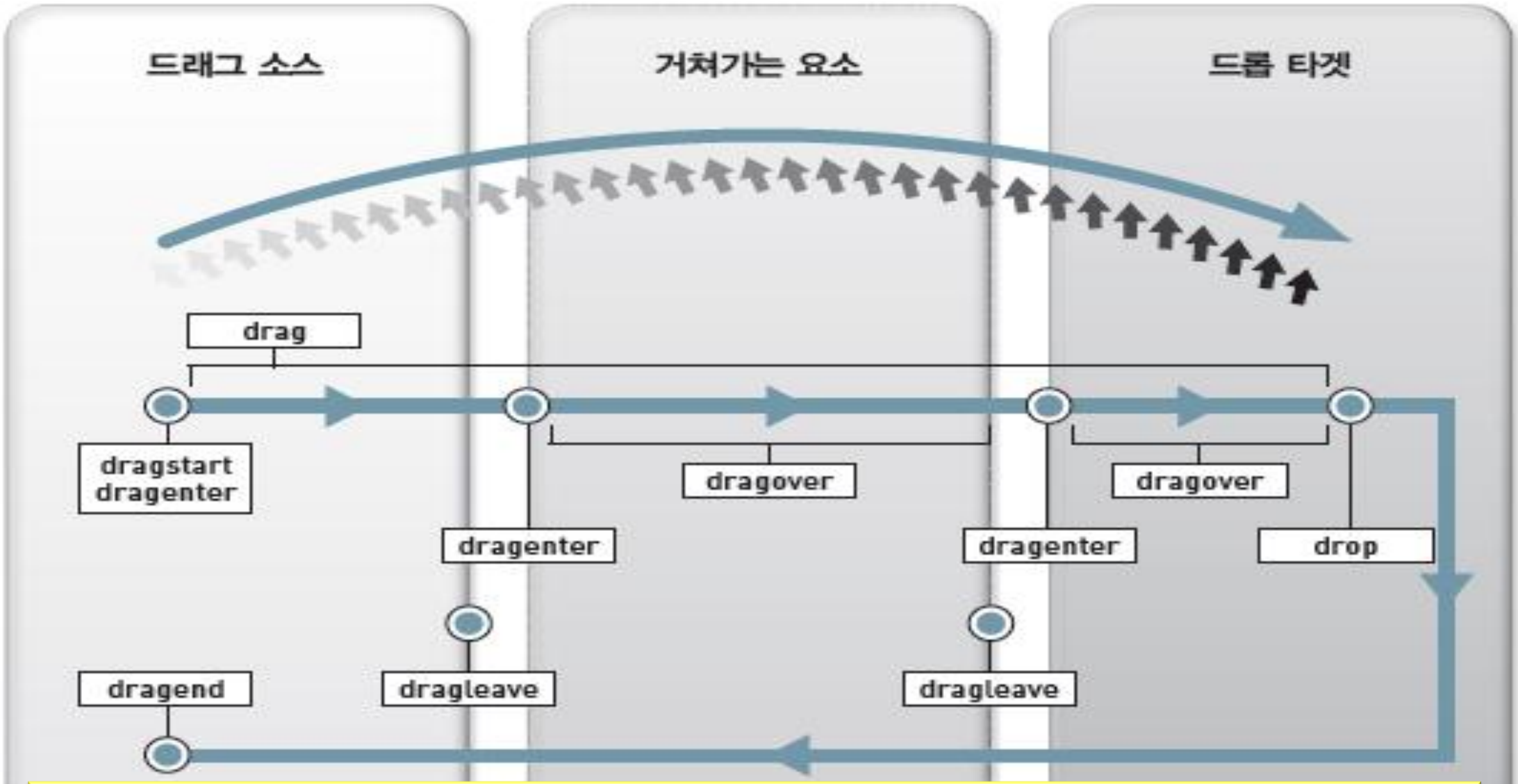
- **드래그(drag)와 드롭(drop)** - 윈도우에서 아주 많이 사용하는 사용자 인터페이스 중의 하나
- 객체를 마우스로 끌어서 다른 애플리케이션에 놓는 것
  - w3school.com의 [?] > [HTML DOM > DOM Events > drag~drop] 참고 (https://www.w3schools.com/jsref/event_ondrag.asp)

드래그



드롭

Drag&Drop(2/4)



소스:Drag**Start** > 소스:Drag**Enter** > 소스:Drag**Over** > 요소1:Drag**Enter** > 소스:Drag**Leave** > 요소1:Drag**Over** > 요소1:Drag**Leave** > 요소2:Drag**Enter** > 요소2:**Drop** > 소스:Drag**End**

Drag&Drop(3/4)

- 드래그 되는 요소의 **draggable 속성을 true로 설정**
- **dragstart** : dataTransfer 객체에 setData() 호출 데이터 설정
- drag : 드래그 도중 계속 발생. 특별히 처리할 내용 없음.
- dragenter : 드래그 중 새로운 요소 안으로 들어가면 발생. 새로운 요소가 타겟 요소인지 검사해서 타겟 요소이면 drop 이벤트 처리
- dragleave : 드래그 중 요소를 빠져 나가면 발생. 특별히 처리할 내용 없음.
- **dragover** : 드래그 도중 마우스가 다른 요소 위에 있을 때 발생. 만약 타겟 요소에서 dragover 이벤트가 발생하면 드롭 허용
- **drop** : 마우스 버튼을 놓았을 때. 반드시 처리 할 내용. dataTransfer 객체에서 getData() 메서드를 이용해 필요한 데이터를 꺼내야 함.

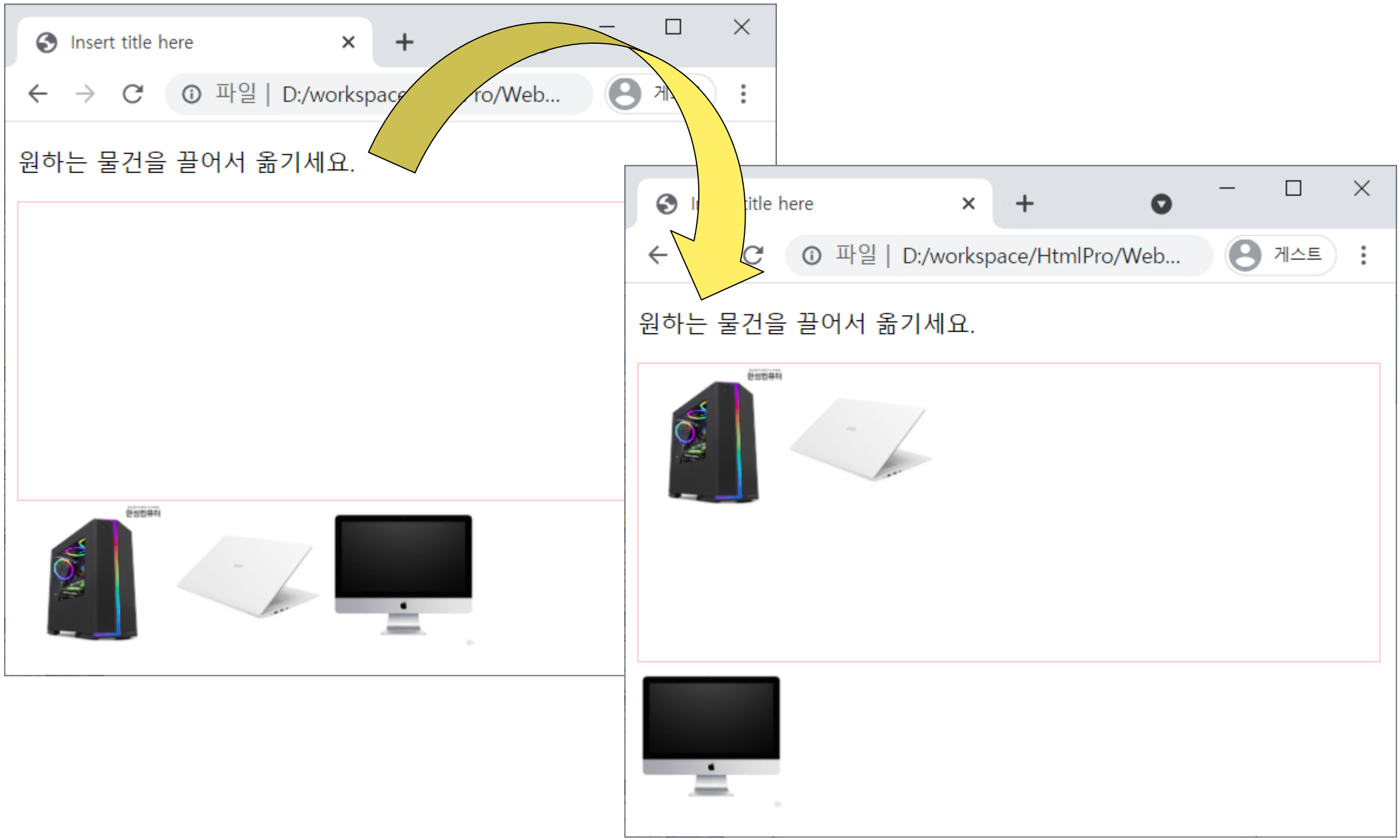
*예제는 https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_ondragenter 참고

Drag&Drop(4/4)

dataTransfer 사용 방법

```
<script>
function allowDrop(ev) {
  //기본 동작을 막는 함수(작성 코드와의 충돌을 피하기 위해)
  ev.preventDefault();//드랍을 허용하도록 함
}
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>
…
<div ondrop="drop(event)" ondragover="allowDrop(event)"></div>
<br>
<img id="drag1" src="../../images/flowers.jpg" width="100"
draggable="true" ondragstart="drag(event)">
```

Drag&Drop 예제(1/2)



Drag&Drop 예제(2/2)

dataTransfer 사용하지 않는 방법

```
<script type="text/javascript">
var obj;// 드래그 대상 객체

function allowDrop(ev){
  ev.preventDefault();
}

function drop(ev){
  ev.preventDefault();
  ev.target.appendChild(obj);

  obj = null;
}
</script>
…
<body>
  <p>원하는 물건을 끌어서 옮기세요.</p>
  <div ondragover="allowDrop(event)" ondrop="drop(event)"></div>
  <img alt="" src="../../images/com1.jpg" ondragstart="obj = this;">
  <img alt="" src="../../images/com2.jpg" ondragstart="obj = this;">
  <img alt="" src="../../images/com3.jpg" ondragstart="obj = this;">
</body>
```

File API

- File API : 웹 브라우저가 사용자 컴퓨터에 있는 로컬 파일들을 읽어올 수 있도록 해주는 API
  - PC에서 실행되는 일반적인 프로그램처럼 동작(웹 애플리케이션)
  - 파일 API의 가장 전형적인 응용 분야는 사용자가 파일을 선택해서 원격 서버로 전송하는 작업

- File API에서 사용되는 객체는 File, FileReader
  - File 객체는 로컬 파일 시스템에서 얻어지는 파일 데이터를 나타낸다.
  - FileReader 객체는 이벤트 처리를 통하여 파일의 데이터에 접근하는 메소드들을 제공하는 객체이다.

File API

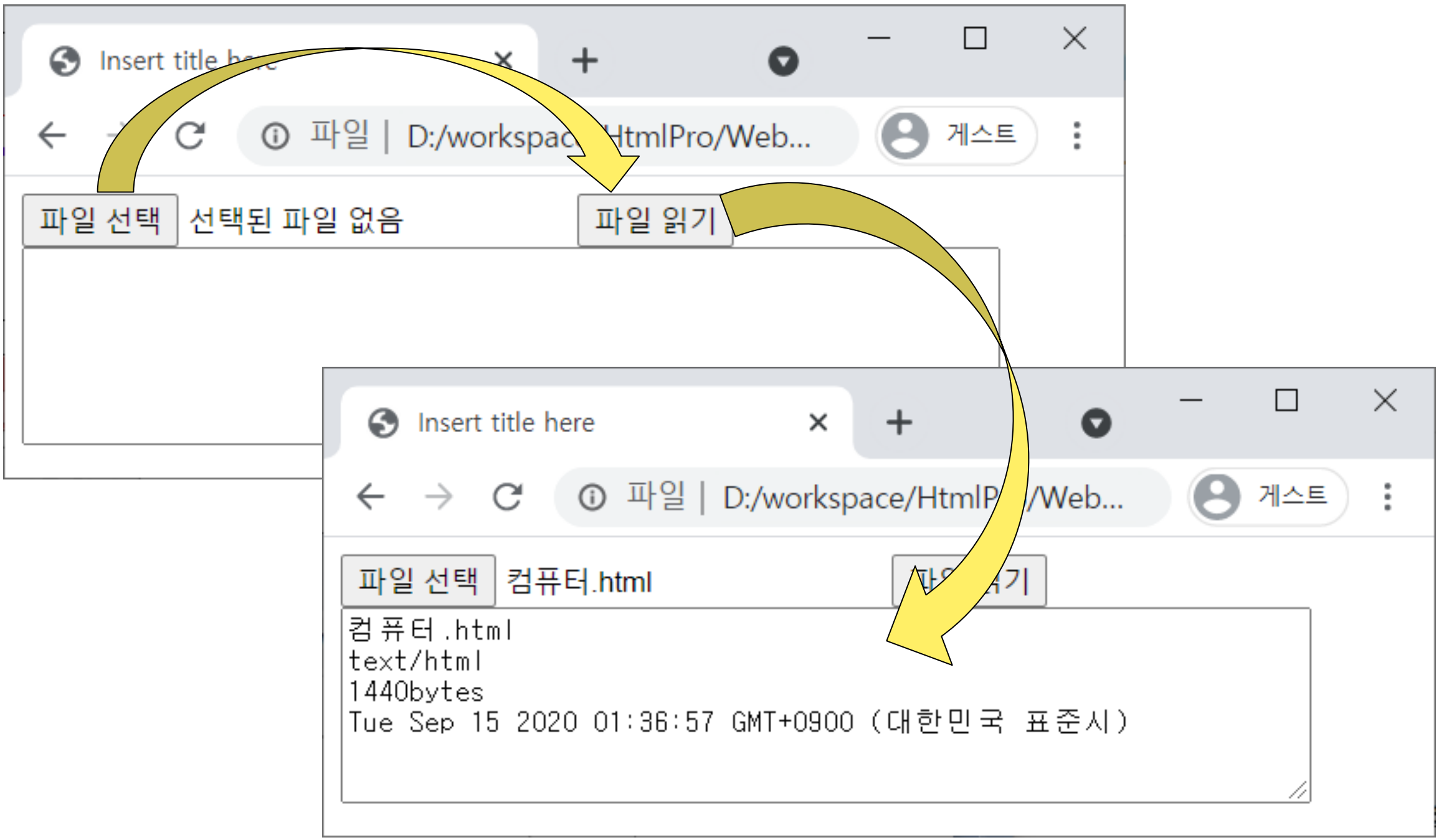
```
<input type="file" id="attachFile" name="attachFile">
```

document.getElementById('attachFile').files	FileList 반환
file.name	파일명 반환
file.type	파일 유형 반환
file.size	파일 사이즈 반환
file.lastModifiedDate	최종 수정 시간

```
var reader = new FileReader();
```

reader.readAsText(*file*); reader.readAsDataURL(file)	-file의 컨텐츠를 문자열로 읽어서 reader의 result에 설정 -DataURL형식으로 읽어서 result에 설정
reader.result	파일의 컨텐츠 반환
reader.onload = function (){…}	읽기가 완료되었을 때 발생

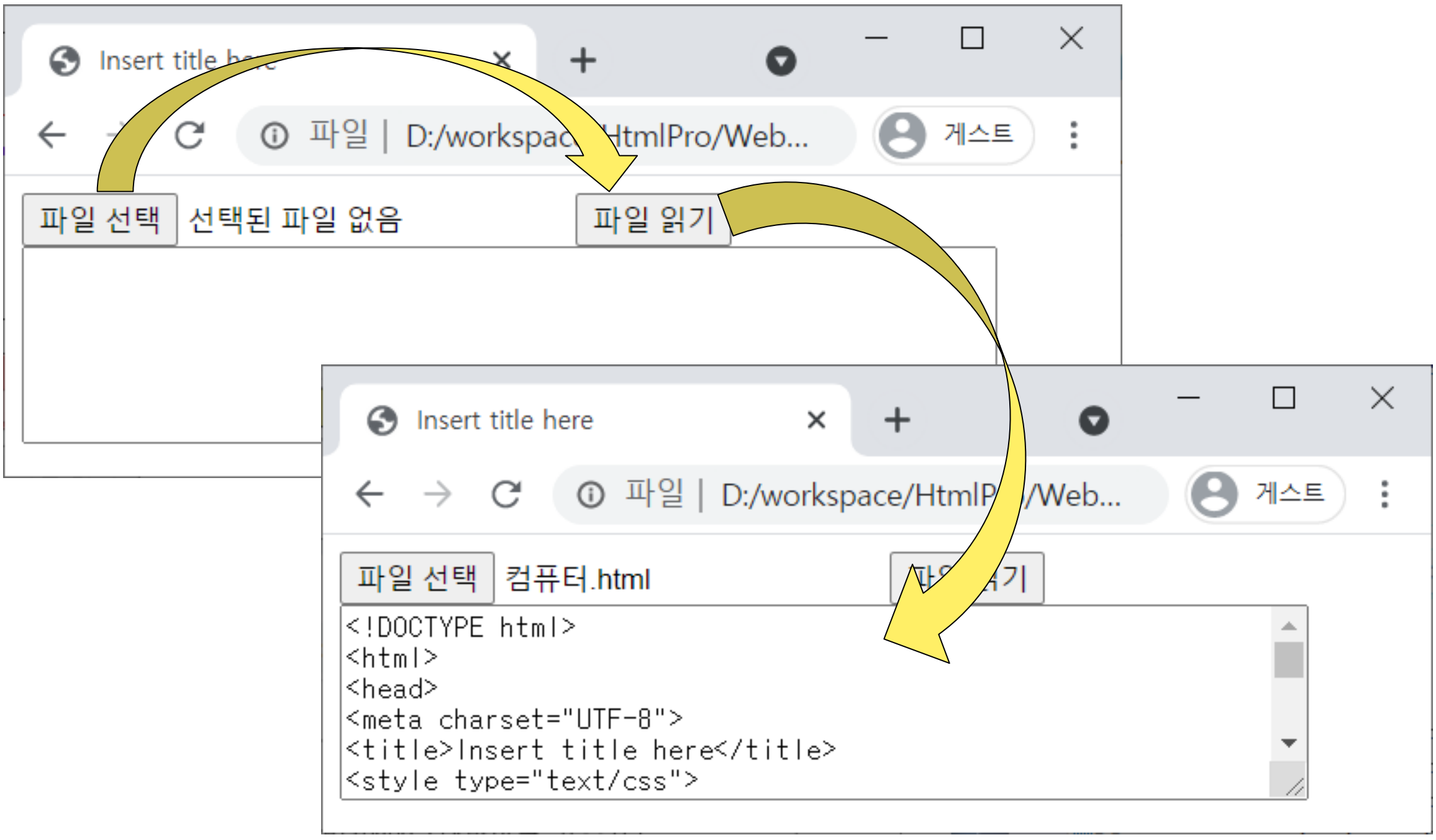
File API 예제1 (1/2)



File API 예제1 (2/2)

```
<script>
function readFile() {
    var files = document.getElementById('attachFile').files;
    var output = "";

    for(var i = 0 ; i < files.length ; i++) {
        var file = files[i];
        output += file.name + "\n";
        output += file.type + "\n";
        output += file.size + "bytes\n";
        output += file.lastModifiedDate + "\n";
    }
    document.getElementById('result').innerHTML = output;
}
</script>
…
<body>
  <input type="file" id="attachFile">
  <button type="button" onclick="readFile()">파일읽기</button><br>
  <textarea id="result" rows="6" cols="60"></textarea>
</body>
```

File API 예제2(1/2)

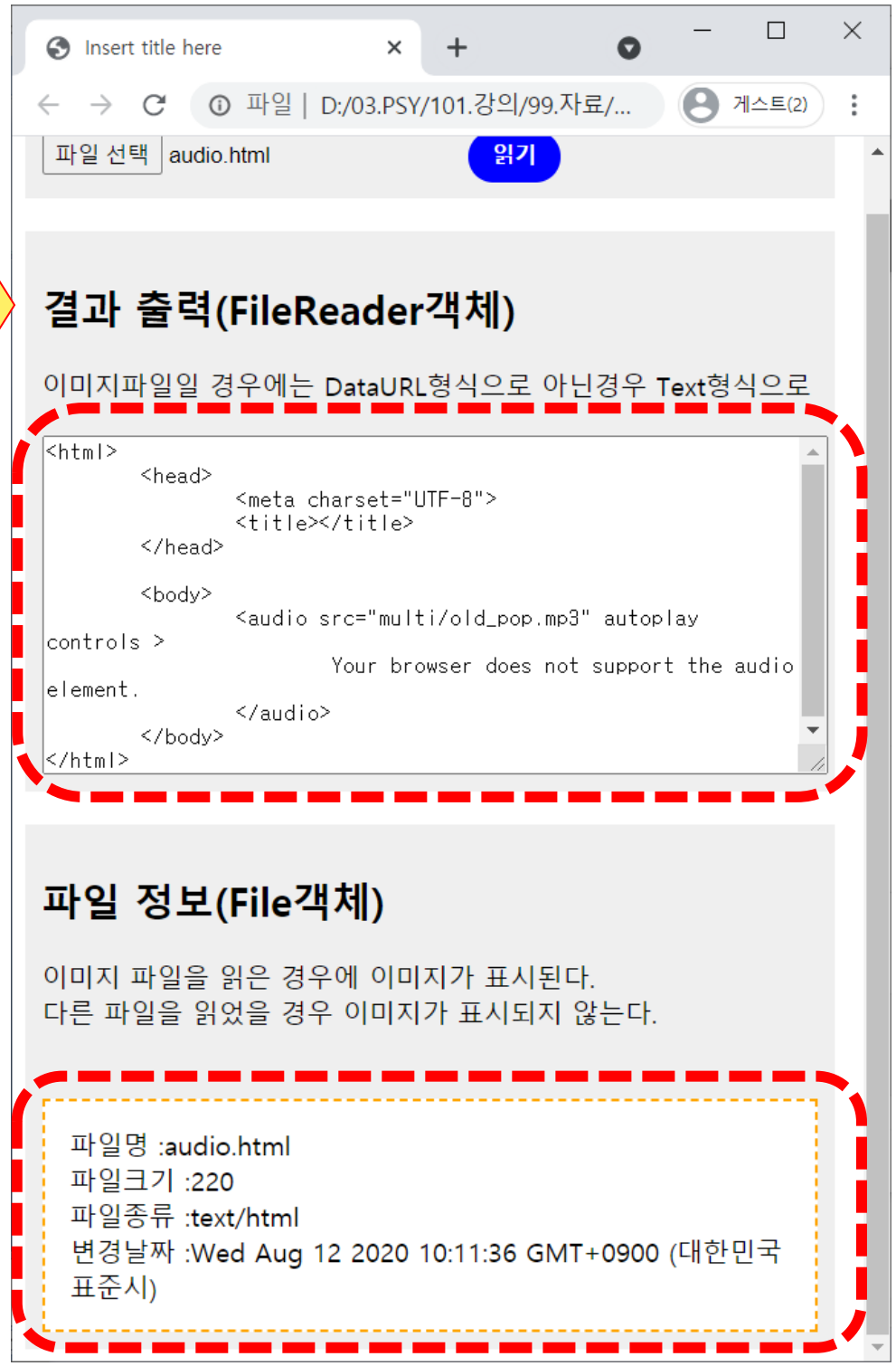


File API 예제2(2/2)

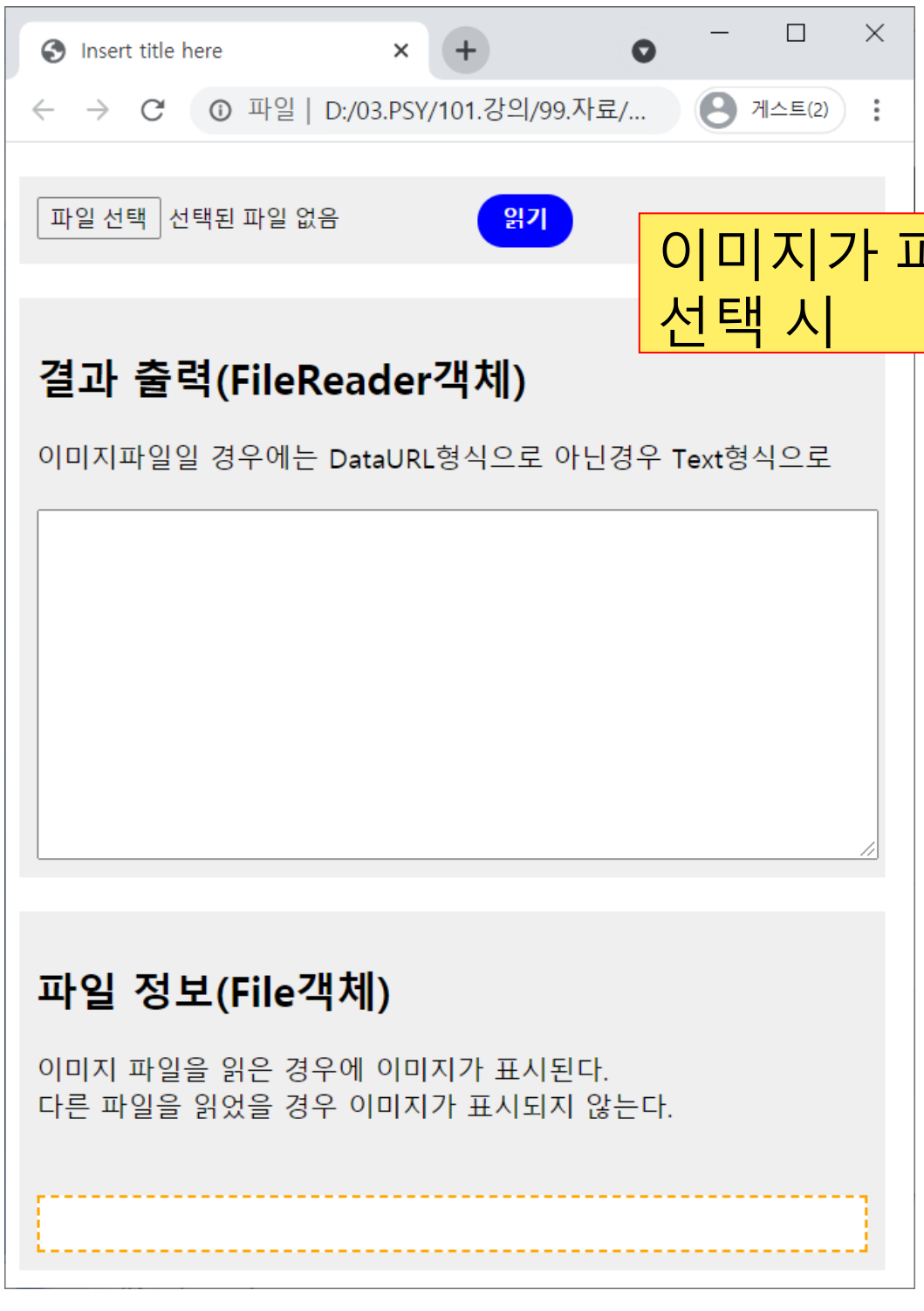
```
<script>
function readFile() {
  if (!window.File || !window.FileReader) {
    alert('File API가 지원되지 않습니다.');
    return;
  }

  var files = document.getElementById('attachFile').files;
  if (!files.length) {
    alert('입력된 파일이 없습니다.');
    return;
  }

  var file = files[0];
  var reader = new FileReader();
  reader.readAsText(file);

  reader.onload = function () {
    document.getElementById('result').innerHTML = reader.result;
  };

}
</script>
```

File API 예제3(1/2)

이미지가 아닌 파일 선택 시



File API 예제3(2/2)



이미지가 파일 선택 시