



웹프로그래밍

Lec11. 이벤트처리와 동적 웹 문서

1. 이벤트 처리하기
2. 폼 다루기
3. 동적 웹 문서 만들기
4. 다양한 방법으로 폼 다루기

상명대
소프트웨어학과
배동성

1. 이벤트 처리하기

- 이벤트 처리 개요

- 이벤트

- 사용자가 웹 브라우저를 사용하는 중에 발생시키는 키보드, 마우스 등의 입력

- 이벤트 처리

- 이벤트가 입력 되었을때 미리 구현된 자바스크립트 코드를 수행

- 1. 이벤트의 정의

- 2. 이벤트 핸들러

- ✓ 이벤트가 발생할때마다 호출되는 자바스크립트 코드

- 3. 이벤트 등록 (registration)

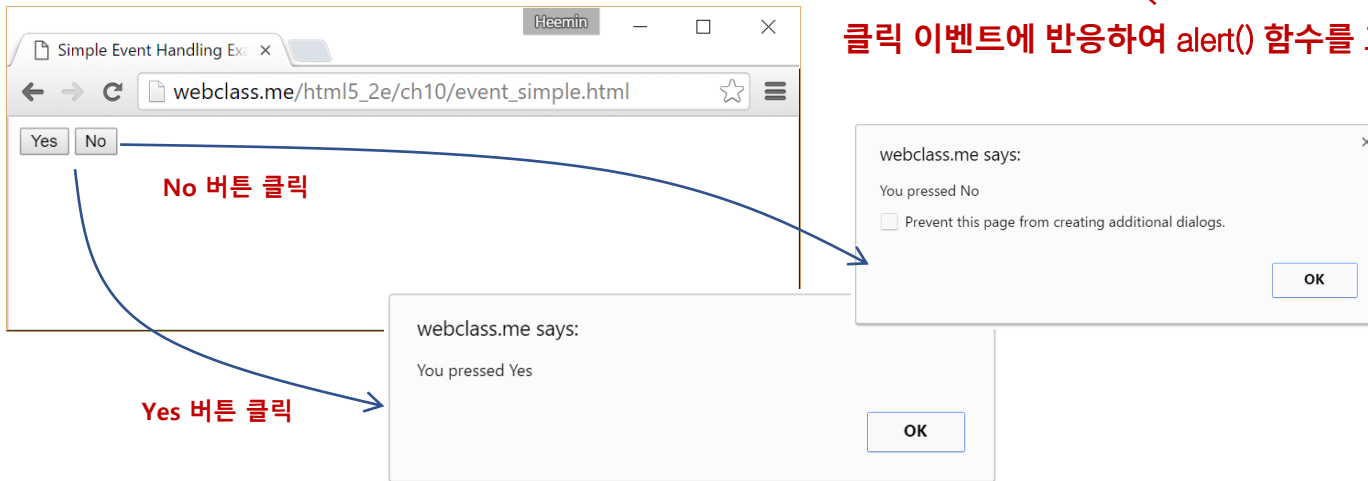
- ✓ 이벤트와 이벤트 핸들러를 연결시키는 과정

1. 이벤트 처리하기

- 간단한 이벤트 처리 예제

```
1 <body>
2   <form>
3     <input type="button" value="Yes" onclick="alert('You pressed Yes');"/>
4     <input type="button" value="No" onclick="alert('You pressed No');"/>
5   </form>
6 </body>
```

클릭 이벤트에 반응하여 alert() 함수를 호출하는 부분



1. 이벤트 처리하기

- 이벤트의 종류
 - 마우스 이벤트

이벤트 이름	태그 속성	설명
click	onclick	HTML 문서내의 요소를 클릭했을때 발생한다.
dblclick	ondblclick	HTML 문서내의 요소를 더블클릭했을때 발생한다.
mousedown	onmousedown	마우스 커서를 HTML 문서내의 요소 위에 위치시키고 마우스 버튼을 누를때 발생한다.
mousemove	onmousemove	마우스 커서를 HTML 문서내의 요소 위에서 이동시킬때 발생한다. 마우스 커서를 움직이는 동안에는 계속해서 이벤트가 발생한다.
mouseup	onmouseup	사용자가 마우스 커서를 HTML 문서내의 요소 위에 위치시키고 마우스 버튼을 떼때 발생한다.
mouseover	onmouseover	마우스 커서가 해당 요소 위에 위치할 경우에 발생한다. 요소 위에 위치할때 1회만 발생하며 연속해서 발생하지 않는다.
mouseout	onmouseout	마우스 커서가 해당 요소 위를 벗어날때 발생하는 이벤트이다.

1. 이벤트 처리하기

- 이벤트의 종류
 - 키보드 이벤트

이벤트 이름	태그 속성	설명
keypress	onkeypress	키보드를 타이핑할때 발생하는 이벤트이다. 키보드를 누를 때 1회 발생하고 손을 떼기 전까지 주기적으로 계속 이벤트가 발생한다.
keydown	onkeydown	키보드를 누를때 발생하는 이벤트이다. 키보드를 눌러서 내려갈때 1회 발생한다.
keyup	onkeyup	키보드를 누른 후 땔때 발생하는 이벤트이다. 키보드에서 손을 땔때 키보드가 올라올때 1회 발생한다.

1. 이벤트 처리하기

- 이벤트의 종류
 - 프레임/객체 이벤트

이벤트 이름	태그 속성	설명
load	onload	문서, 프레임, 객체 등이 웹 브라우저상에 로드가 완료 되었을때 발생하는 이벤트이다.
resize	onresize	문서 창, 문서 뷰의 크기가 리사이즈(resize) 되었을 경우 발생한다.
scroll	onscroll	문서 창, 문서 뷰가 스크롤 되었을 경우 발생한다.

1. 이벤트 처리하기

- 이벤트의 종류
 - 폼 이벤트

이벤트 이름	태그 속성	설명
change	onchange	<input>, <selection>, <textarea> 등 폼 요소 콘텐츠의 내용이 변경되었을 때 발생하는 이벤트이다.
focus	onfocus	요소가 포커스 되었을 때 발생하는 이벤트이다. 마우스로 선택되거나 입력 커서가 해당 요소에 위치할 때 발생한다.
blur	onblur	focus 이벤트의 반대 개념으로 요소에서 포커스가 없어질 때 발생하는 이벤트이다. 즉, 요소가 마우스 선택이 해제되거나 입력 커서가 다른 곳으로 이동할 때 발생한다.
select	onselect	<input> 과 <textarea> 요소 내의 텍스트의 일부 혹은 전부가 선택 되었을 때 발생하는 이벤트이다.

1. 이벤트 처리하기

- 이벤트 핸들링 및 이벤트 등록
 - 이벤트 핸들러(handler)
 - 이벤트가 발생시 실행하고자 하는 자바스크립트 함수나 코드
 - 사용자가 입력한 내용이 맞는지 검사하거나 입력한 내용에 따라 웹 문서를 수정하는 등의 작업을 통해 동적 웹 문서를 만든다
 - 이벤트 등록
 - 이벤트의 종류와 이를 처리할 이벤트 핸들러를 연결시키는 작업
 - 두가지 등록 방법
 1. 태그 속성에 직접 이벤트 핸들러 기술
 2. 객체의 이벤트 속성 값에 이벤트 핸들러 함수 기술

1. 이벤트 처리하기

- 요소 속성에 이벤트 핸들러 기술
 - 요소의 이벤트 태그 속성에 직접 이벤트 핸들러 기술
 - 이벤트 핸들러: 자바스크립트 코드 혹은 함수 이름

```
<form action="">
  <input id="username" type="text" value="Name of User"
    onclick="alert('Please type your full name');" />
</form>
```

```
<form action="">
  <input id="username" type="text" value="Name of User"
    onclick="myEventHandler();" />
</form>

<script type="text/javascript">
  function myEventHandler() {
    alert("Please type your full name");
  }
</script>
```

1. 이벤트 처리하기

- DOM 인터페이스를 이용한 이벤트 핸들러 함수 등록
 - DOM 인터페이스를 이용하여 요소 객체의 해당 이벤트 속성에 직접 이벤트 핸들러 함수를 지정
 - 이벤트 핸들러는 반드시 함수 형태로 미리 구현

```
<form action="">
  <input id="username" type="text" value="Name of User" />
</form>

<script type="text/javascript">
  function myEventHandler() {
    alert("Please type your full name");
  }

  var dom = document.getElementById("username");
  dom.onclick = myEventHandler;
</script>
```

- 주의: 객체의 이벤트 속성에 ()없이 함수 이름만 적어야 함

1. 이벤트 처리하기

- setTimeout()을 이용한 예제
 - setTimeout() 함수와 버튼 click을 이용한 스톱워치 예제
 - 1초마다 start() 함수를 호출하여 시간 값을 주기적으로 증가

```
1 <script type="text/javascript">
2   var stopped = false;
3
4   function start() {
5       dom = document.getElementById("sec");
6       dom.value = parseInt(dom.value) + 1;
7       if (!stopped) setTimeout(start,1000);
8   }
9 </script>
10
11 <h2> Stopwatch </h2>
12 <form action="">
13   <input type="button" value="Start" onclick="stopped=false; setTimeout(start,1000);"
14 />
15   <input type="button" value="Stop" onclick="stopped=true;" />
16   <input type="reset"/> <br><br>
17   <input id="sec" type="text" value="0" size="2"/> seconds
18 </form>
```

Stopwatch

seconds

1. 이벤트 처리하기

- 이벤트 핸들러 함수 등록 방법

```
function myEventHandler() {  
    alert("Please type your full name");  
}
```

1. 요소의 이벤트 태그 속성에 직접 이벤트 핸들러 함수 호출

```
<form action="">  
    <input id="username" type="text" onclick="myEventHandler();" />  
</form>
```

2. DOM을 이용하여 요소 객체의 이벤트 속성에 이벤트 핸들러 함수를 지정

```
document.getElementById("username").onclick = myEventHandler;
```

3. DOM을 이용해 요소 객체의 addEventListener 메소드 이용

```
document.getElementById("username").addEventListener("click", myEventHandler);
```

1. 이벤트 처리하기

- 이벤트 핸들러에 파라미터 넘기는 방법

```
function myEventHandler(num, name) {  
    document.getElementById("txt").innerHTML = "이름: " + name + " 번호:" + num;  
}
```

1. 요소의 이벤트 태그 속성에 직접 이벤트 핸들러 함수 호출

```
<form action="">  
    <input id="username" type="text" onclick="myEventHandler(12534, '김상명');" />  
</form>
```

2. DOM을 이용하여 요소 객체의 이벤트 속성에 이벤트 핸들러 함수를 지정

```
document.getElementById("username").onclick = function() {  
    myEventHandler(12534, '김상명');  
};
```

3. DOM을 이용해 요소 객체의 addEventListener 메소드 이용

```
document.getElementById("username").addEventListener(  
    "click",  
    function() { myEventHandler(56789, '이사슴'); } );
```

1. 이벤트 처리하기

- 이벤트 등록 예제 코드

```
<!DOCTYPE html>
<html>
<body>

<button id="btn1">Button1</button>
<button id="btn2">Button2</button>

<p id="txt">aa </p>

<script>
    document.getElementById("btn1").onclick = function() { myEventHandler(12345, '김상명'); };

    document.getElementById("btn2").addEventListener("click", function() { myEventHandler(56789, '이사슴'); } );

    function myEventHandler(num, name) {
        document.getElementById("txt").innerHTML = "이름: " + name + " 번호:" + num;
    }
</script>

</body>
</html>
```

2. 폼 다루기

- 폼 다루기와 이벤트 처리 예제

- 자바스크립트를 이용해 폼의 값을 읽어내거나 계산하여 수정하는 것이 가능
- `<input>`요소를 DOM 인터페이스로 접근하여 `value` 속성값을 읽거나 저장

```
<script type="text/javascript">
```

```
function add() {
```

```
    var a = document.getElementById("op1").value;
```

```
    var b = document.getElementById("op2").value;
```

```
    document.getElementById("result").value = parseInt(a) +
```

```
    parseInt(b);
```

```
}
```

```
</script>
```

```
<form>
```

```
    <input id="op1" type="text" size="2"/>+
```

```
    <input id="op2" type="text" size="2"/>
```

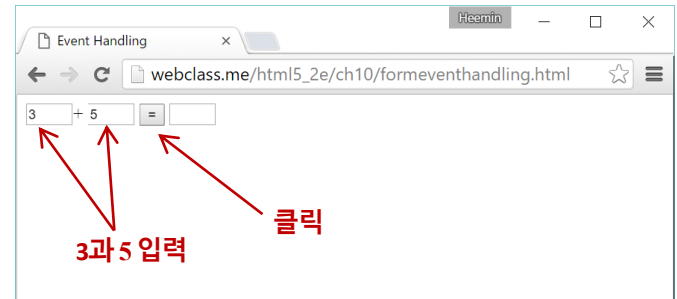
```
    <input type="button" value="=" onclick="add();"/>
```

```
    <input id="result" type="text" size="2"/>
```

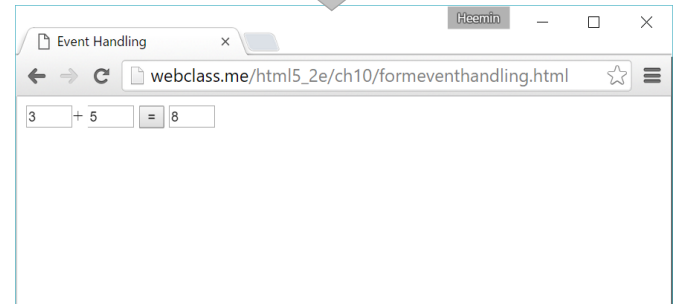
```
</form>
```

폼의 값을 읽어 낸다

폼의 `<input>` 요소에 값을 쓴다



add() 함수가 호출됨



2. 폼 다루기

- 폼의 입력 값 읽고 쓰기
 - 자바스크립트를 이용해 폼의 값을 읽거나 수정한다

The diagram illustrates a form state transition. On the left, the initial state of the '서적 주문 양식' (Book Order Form) is shown. The '주문서' (Order Form) table has all quantity and total columns set to 0. On the right, after a calculation, the quantities are updated to 2, 3, 5, and 10, and the totals are updated to 50000, 81000, 125000, and 256000 respectively. A large gray arrow points from the initial state to the updated state.

서적 주문 양식
주문서

책 제목	가격	수량	합계
멀티미디어 배움터2.0	25,000원	0	0 원
모바일 멀티미디어	27,000원	0	0 원
자바입문: 이론과 실습	25,000원	0	0 원
합계		0	0 원

합계계산 초기화

서적 주문 양식
주문서

책 제목	가격	수량	합계
멀티미디어 배움터2.0	25,000원	2	50000 원
모바일 멀티미디어	27,000원	3	81000 원
자바입문: 이론과 실습	25,000원	5	125000 원
합계		10	256000 원

합계계산 초기화

- <input> 요소를 DOM 인터페이스로 접근하여 value 속성값을 읽거나 저장

2. 폼 다루기

• 폼의 입력 값 읽고 쓰기 예제 코드

```
<body>
<h2> 서적 주문 양식 </h2>
<form action="">
  <p> 주문서 </p>
  <table border="1">
    <tbody><tr> <th> 책 제목 </th><th> 가격 </th> <th> 수량 </th>
    <th> 합계 </th> </tr>
    <tr>
      <td> 멀티미디어 배움터2.0 </td> <td> 25,000원 </td>
      <td> <input id="book1" type="text" size="2" value="0"
onclick="this.select();" > </td>
      <td> <input id="book1Total" type="text" size="6" value="0">
원</td>
    </tr>
    <tr>
      <td> 모바일 멀티미디어 </td> <td> 27,000원 </td>
      <td> <input id="book2" type="text" size="2" value="0"
onclick="this.select();" > </td>
      <td> <input id="book2Total" type="text" size="6" value="0">
원</td>
    </tr>
    <tr>
      <td> 자바입문: 이론과 실습 </td> <td> 25,000원 </td>
      <td> <input id="book3" type="text" size="2" value="0"
onclick="this.select();" > </td>
      <td> <input id="book3Total" type="text" size="6" value="0">
원</td>
    </tr>
    <tr>
      <td> 합계 </td> <td> &nbsp;</td>
      <td> <input id="totalNumber" type="text" size="2"
value="0"> </td>
      <td> <input id="totalPrice" type="text" size="6" value="0">
원</td>
    </tr>
  </tbody></table> <br>
  <input type="button" value="합계계산" onclick="updateAll();">
  <input type="reset" value="초기화">
</form>
<script type="text/javascript">
function updateAll() {
  var n1 = document.getElementById("book1").value;
  var n2 = document.getElementById("book2").value;
  var n3 = document.getElementById("book3").value;

  var p1 = 25000 * n1;
  var p2 = 27000 * n2;
  var p3 = 25000 * n3;
  document.getElementById("book1Total").value = p1;
  document.getElementById("book2Total").value = p2;
  document.getElementById("book3Total").value = p3;

  var totalNumber = parseInt(n1) + parseInt(n2) +
parseInt(n3);
  document.getElementById("totalNumber").value = totalNumber;
  var totalPrice = p1 + p2 + p3;
  document.getElementById("totalPrice").value = totalPrice;
}
</script>
</body>
```

2. 폼 다루기

- 폼의 입력 값 읽고 쓰기
 - DOM을 통해 <input> 텍스트 박스 위젯에 입력한 값을 읽어 낸다
 - 요소 객체의 value라는 속성을 통해 읽거나 수정

```
<input id="book1" type="text" size="2" value="0" onclick = "this.select();" />  
<input type="button" value="합계계산" onclick="updateAll();" />
```

```
var n1 = document.getElementById("book1").value;  
var p1 = 25000 * n1;  
document.getElementById("book1Total").value = p1;
```

3. 동적 웹 문서 만들기

- 동적 문서 정의

- 동적 문서

- 웹 문서가 브라우저상에 처음에 표시된 이후에 콘텐츠나 스타일이 변경되어 화면에 표시되는 내용이나 표현 형태가 변경되는 문서
 - 웹 문서의 콘텐츠나 스타일이 변경되면 즉시 변경된 값을 바탕으로 화면의 문서를 갱신

- 웹 문서의 콘텐츠나 스타일의 변경

- 태그 요소, 태그 속성, 태그 콘텐츠, 요소의 CSS 스타일 등의 값을 자바스크립트를 이용해서 변경
 - 태그 요소의 화면내 표시 위치 변경, 애니메이션, 색상 및 글씨체 변경, 인터랙티브 사용자 인터페이스 등 포함

- 동적 문서 구현 방식

- 1. 스타일 속성 변경을 통한 방법
 - 2. 요소의 콘텐츠를 변경시키는 방법

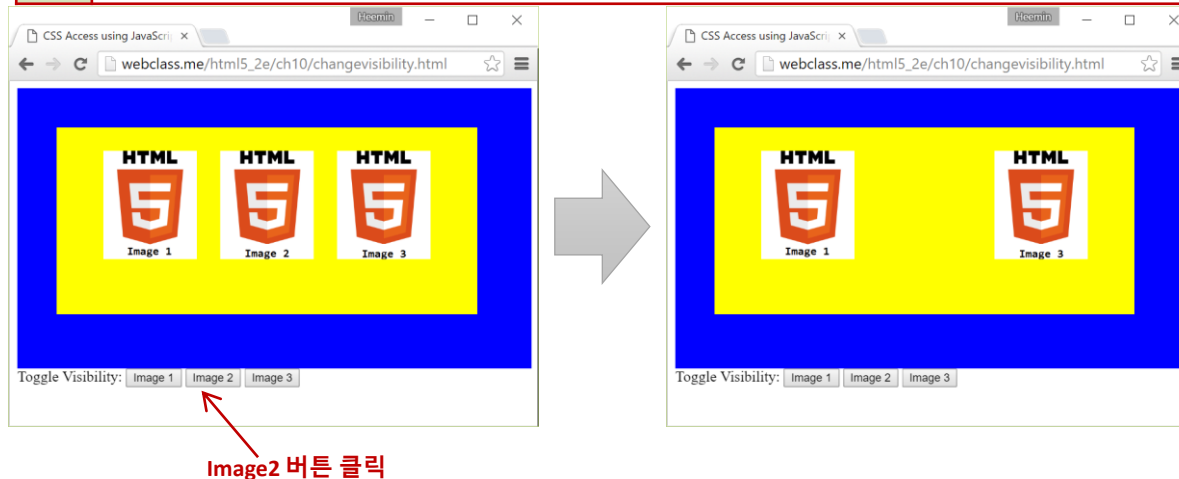
3. 동적 웹 문서 만들기

- 스타일 속성 변경을 통한 동적 문서
 - DOM을 이용하면 CSS 스타일에 접근 가능
 - 일반적인 DOM에 접근하는 방법과 동일
 - CSS 스타일 값 변경으로 웹 문서를 보다 동적으로 제작 가능
 - 보이기/감추기 스타일 속성 변경 예제
 - 화면 표시 여부를 결정
 - ✓ 보이기/감추기 스타일 속성
 - ✓ 스타일 속성: visibility
 - ✓ 속성 값: visible 혹은 hidden
 - hidden으로 설정 되도 웹 문서 내에 요소로는 존재
 - ✓ 화면에 표시만 안될 뿐이다

3. 동적 웹 문서 만들기

- 보이기 스타일 속성 변경 예제

```
1 <script type = "text/javascript">
2 function toggleVisibility(id) {
3   var dom = document.getElementById(id);
4
5   if (dom.style.visibility == "visible")
6     dom.style.visibility = "hidden";
7   else
8     dom.style.visibility = "visible";
9 }
10 </script>
11
12 Toggle Visibility:
13 <button onclick = "toggleVisibility('img1');">Image 1</button>
14 <button onclick = "toggleVisibility('img2');">Image 2</button>
15 <button onclick = "toggleVisibility('img3');">Image 3</button>
```



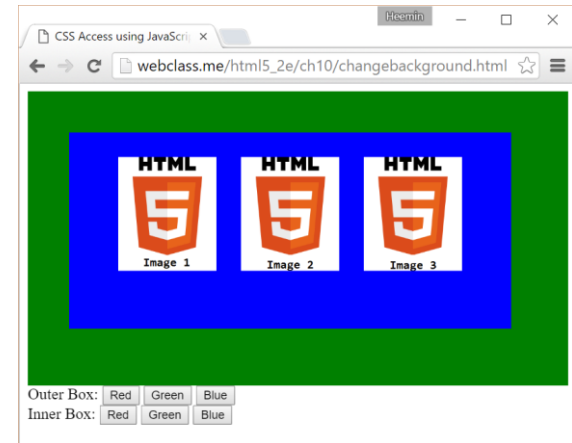
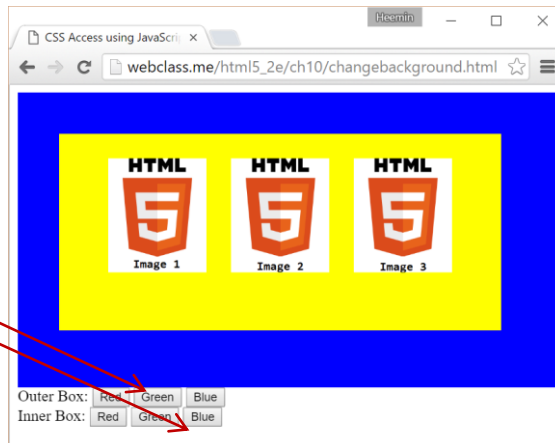
3. 동적 웹 문서 만들기

• 배경색 스타일 속성 변경하기

style.background
속성 접근

```
1 <script type = "text/javascript">
2 function changeColor(id, color) {
3     document.getElementById(id).style.background = color;
4 }
5 </script>
6
7 Outer Box:
8 <button onclick = "changeColor('outerBox', 'red');"> Red </button>
9 <button onclick = "changeColor('outerBox', 'green');"> Green </button>
10 <button onclick = "changeColor('outerBox', 'blue');"> Blue </button>
11 <br />
12
13 Inner Box:
14 <button onclick = "changeColor('innerBox', 'red');"> Red </button>
15 <button onclick = "changeColor('innerBox', 'green');"> Green </button>
16 <button onclick = "changeColor('innerBox', 'blue');"> Blue </button>
```

Green과 Blue
버튼 클릭



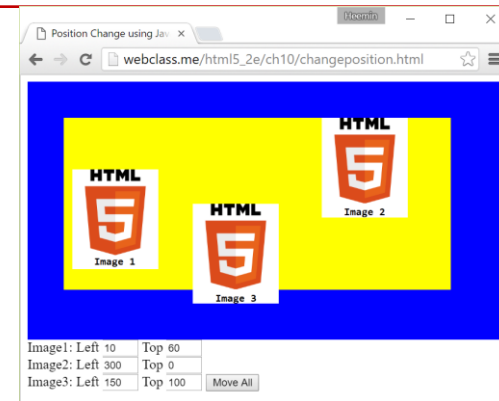
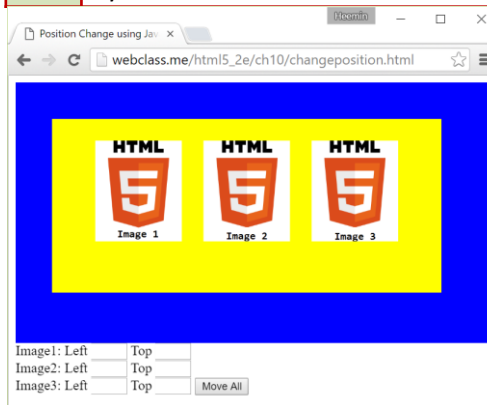
3. 동적 웹 문서 만들기

- 위치 스타일 속성 변경

style.left 속성 접근

style.top 속성 접근

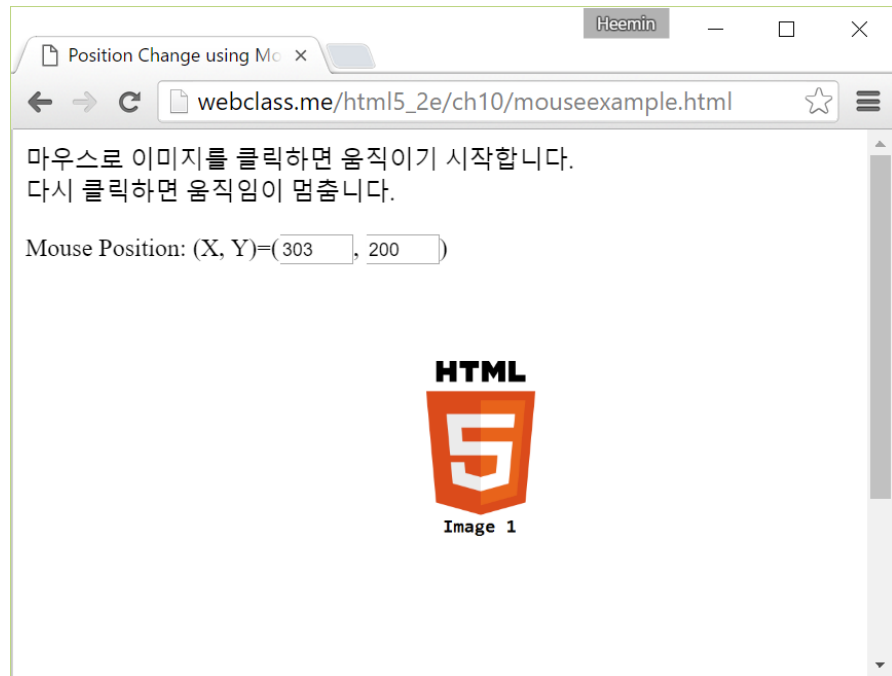
```
1 <script type = "text/javascript">
2   function changePositions() {
3     for(i = 1; i <= 3; i++) {
4       var left = document.getElementById("left" + i).value;
5       var top = document.getElementById("top" + i).value;
6       document.getElementById("img" + i).style.left = left + "px";
7       document.getElementById("img" + i).style.top = top + "px";
8     }
9   }
10 </script>
11
12 <form>
13   Image1: Left <input id = "left1" size = "2" type = "text"/>
14   Top <input id = "top1" size = "2" type = "text"/> <br />
15
16   Image2: Left <input id = "left2" size = "2" type = "text"/>
17   Top <input id = "top2" size = "2" type = "text"/> <br />
18
19   Image3: Left <input id = "left3" size = "2" type = "text"/>
20   Top <input id = "top3" size = "2" type = "text"/>
21
22   <input type = "button" value = "Move All" onclick = "changePositions();" />
23 </form>
```



3. 동적 웹 문서 만들기

- 마우스 이벤트를 이용한 위치 스타일 변경하기
 - 마우스 이벤트를 이용한 위치 스타일 변경
 - 웹브라우저 화면상의 위치 좌표 값을 이용해 요소의 위치 변경
 - 마우스 포인터의 위치
 - window 객체의 event 속성값에서 구할 수 있음 (client, clientY)

```
e = window.event;  
mouse_x = e.clientX;  
mouse_y = e.clientY;
```



3. 동적 웹 문서 만들기

- 마우스 이벤트를 이용한 위치 스타일 변경 예제

```
<body onmousemove="move();" style="width:500px;
height:500px;">
  

  <script type="text/javascript">
    var stopped = true;
    document.getElementById("img1").onclick = start_stop_move;

    function start_stop_move() {
      if (stopped) stopped = false;
      else stopped = true;
    }

    function move() {
      e = window.event;
      mouse_x = e.clientX;
      mouse_y = e.clientY;

      document.getElementById("x").value = mouse_x;
      document.getElementById("y").value = mouse_y;

      if (!stopped) {
        document.getElementById("img1").style.left = (mouse_x-50)
+ "px";
        document.getElementById("img1").style.top = (mouse_y-50) +
"px";
      }
    }
  </script>

  <form>
```

마우스로 이미지를 클릭하면 움직이기 시작합니다.
다시 클릭하면 움직임이 멈춥니다.

Mouse Position:
(X, Y)=(<input id="x" size="3" type="text">,
<input id="y" size="3" type="text">)
</form>
</body>

3. 동적 웹 문서 만들기

- 요소의 콘텐츠 변경을 통한 동적 문서
 - 웹 문서 콘텐츠 변경
 - 폼 위젯 요소의 value 값을 변경
 - 태그 요소의 콘텐츠를 변경
 - HTML 태그 콘텐츠 속성

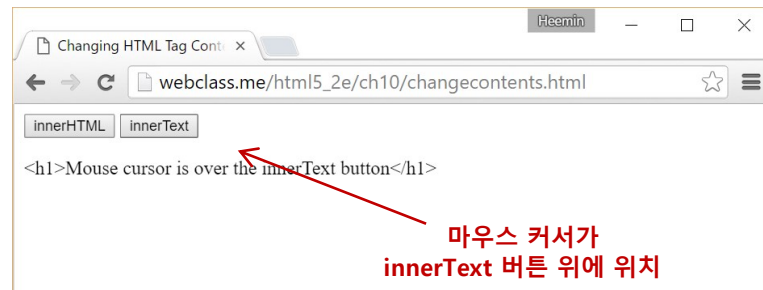
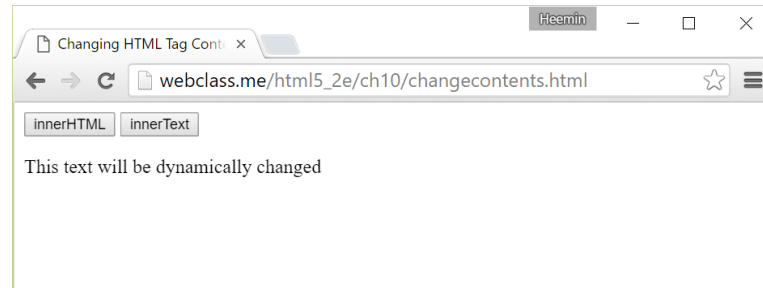
```
<p id = "example"> This is an example content </p>
```

- “**This is an example content**” 부분이 <p> 요소의 콘텐츠
- 요소의 콘텐츠에 접근하는 속성: innerHTML, innerText
 - ✓ innerHTML
 - 속성에 저장된 값을 HTML 태그로 해석
 - ✓ innerText
 - 속성에 저장된 값을 단순히 문자열로 해석

3. 동적 웹 문서 만들기

- 콘텐츠 변경을 통한 동적 문서 예제

```
<button id="b1">innerHTML</button>
1 <button id="b2">innerText</button>
2 <br/>
3 <p id="text">This text will be dynamically changed</p>
4
5 <script type="text/javascript">
6   document.getElementById("b1").onmouseover =
7   mouseover_innerHTML;
8   document.getElementById("b2").onmouseover =
9   mouseover_innerText;
10
11   document.getElementById("b1").onmouseout = reset_text;
12   document.getElementById("b2").onmouseout = reset_text;
13
14   function mouseover_innerHTML() {
15     document.getElementById("text").innerHTML =
16     "<h1>Mouse cursor is over the innerHTML button</h1>";
17   }
18
19   function mouseover_innerText() {
20     document.getElementById("text").innerText =
21     "<h1>Mouse cursor is over the innerText button</h1>";
22   }
23
24   function reset_text() {
25     document.getElementById("text").innerHTML =
26     "This text will be dynamically changed";
27   }
28 </script>
```



4. 다양한 방법으로 폼 다루기

- 폼 접근하기

- document.getElementById() 메소드 이외의 다른 방법으로 폼에 접근하는 방법들
 - document.폼이름.위젯ID.value;
 - document.forms[index].위젯ID.value;
 - document.forms[index].elements[index].value;

value of form

Access using getElementById: value of form

Access using form_name.id: value of form

Access using form[index].id: value of form

Access using form[index].elements[index]: value of form

4. 다양한 방법으로 폼 다루기

- 폼 제어하기

- 자바스크립트로 폼 위젯을 제어할 수 있는 방법들

- select() 메소드
- submit() 메소드
- reset() 메소드
- checked 속성 설정



```
1 <form name="form1" action="">
2   <input id="input1" type="text" value="value of form" />
3   <input id="input2" type="checkbox"/> <br/>
4   <input type="button" value="select()" onclick="input1.select();"/> <br/>
5   <input type="button" value="submit()" onclick="submit();"/> <br/>
6   <input type="button" value="reset()" onclick="reset();"/> <br/>
7   <input type="button" value="check" onclick="input2.checked=true;"/> <br/>
8   <input type="button" value="uncheck" onclick="input2.checked=false;"/> <br/>
9 </form>
```

기말 프로젝트

- 데이터 공유 사이트 만들기
 - 조건에 맞도록 사이트 구성
 - HTML, CSS, JS 를 이용하여 기능 구현
 - 수업시간 내용 및 개인 공부한 내용 추가 허용
 - 차주 수업까지 팀원 등록한 팀에 한하여 팀프로젝트 허용

I. 데이터 공유 사이트 목표

■ 교육의 場

- 학생들에게 **실습** 가능한 데이터셋을 제공
- 오픈 SW를 통해 자신의 방법과 다른 사람의 방법과 **경쟁**하며 성장

■ 취업의 場

- 산업 현장에서 얻은 데이터셋을 이용한 **실무 경험** 제공
- OSS Resume와 연계한 **산학 연계** 과제 수행 혹은 **인턴쉽** 진행

■ 진학의 場

- **대학원**에서 연구하는 데이터셋을 공유하여 관심있는 학생을 모집
- 최근 **연구적**으로 이슈가 되고 있는 데이터셋을 체험하는 공간 제공



II. 데이터 공유 사이트 구성

상명 Data Wiki

- 현재의 Data Wiki
 - 데이터셋 확보를 위한 기본 틀과 대표적인 60개의 데이터셋 보유

항목	내용
ID(name)	Dataset명
Description	Dataset 설명
Link	URL
Category	Dataset 카테고리
Field	Dataset 분야
Task	Dataset의 목적
Instance	Dataset내의 data 수
Num	Index
Tutorial	Dataset 관련 사이트 (일부)

- 추진 계획
 - 기존 데이터 공유 사이트와의 차별성 확보를 위한 콘텐츠 확보
 - 데이터셋을 이용한 학습이 가능한 사이트
 - 데이터셋의 목적성을 제공하는 사이트
 - 데이터셋에 대해 친절한 설명이 가능한 사이트

II. 데이터 공유 사이트 구성

상명 Data Wiki



- Data Share Hub
 - 데이터 분석을 위한 기초 자료를 제공
 - 기존 교육자료를 활용한 교육 콘텐츠
 - TensorFlow
 - Google 사의 대표 ML 라이브러리
 - 라이브러리 활용을 위한 예제 코드 제공
 - Pytorch
 - Facebook에서 제공하는 ML 오픈SW
 - 라이브러리 활용을 위한 콘텐츠 제공
- 각 데이터셋 별 Open SW 연계
- 설명이 적은 데이터셋 Data Competition 진행 및 공유 자료 확보



초보자용

사용자 친화적 Keras Sequential API로 시작하는 것이 가장 좋습니다. 구성요소를 함께 연결하여 모델을 빌드해 보세요. 이러한 튜토리얼을 완료한 후 [Keras 가이드](#)를 읽어보세요.

초급자용 빠른 시작

이 'Hello, World!' 메모장에는 Keras Sequential API 및 `model.fit`가 표시됩니다.

Keras 기본사항

이 메모장 컬렉션은 Keras를 사용한 기본적인 머신러닝 작업을 보여줍니다.

데이터 로드

이러한 튜토리얼에서는 `tf.data`를 사용하여 다양한 데이터 형식을 로드하고 입력 파이프라인을 빌드합니다.



파이토치(PyTorch) 레시피 [+]

파이토치(PyTorch) 시작하기 [-]

파이토치(PyTorch) 기본 익히기

빠른 시작(Quickstart)

텐서(Tensor)

Dataset과 DataLoader

변형(Transform)

신경망 모델 구성하기

`torch.autograd`를 사용한 자동 미분

모델 매개변수 최적화하기

모델 저장하고 불러오기

II. 데이터 공유 사이트 구성

상명 Data Wiki

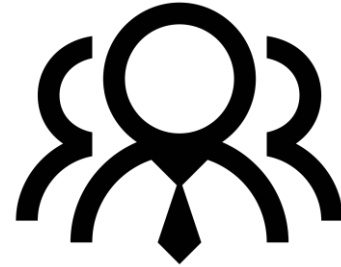


- Data Colosseum
 - 교내 경진대회를 통한 데이터셋 활용
 - 연 2회 Data Wiki 내 데이터셋을 이용한 경진대회 개최
 - 활용사례가 적은 최신 데이터셋에 대한 오픈 SW 자료 확보
 - 경진대회 수상자는 상금과 6개월 간 Data Manager 의무 수행
 - 수업 과제로 Data Wiki 활용
 - 데이터 및 인공지능 수업에서 활용
 - Data Wiki를 활용한 교수법 연구를 통한 활용방법 공유

II. 데이터 공유 사이트 구성

상명 Data Wiki

- Data Manager
 - Data Competition의 우승자에게 부여되는 권한
 - 우승한 데이터셋에 대한 설명 및 수정권한을 부여 받음
 - Data Wiki를 통하여 수상자와 질의 응답이 가능한 통로를 만듦



III. 사이트 구성

■ 상명 Data Wiki

- 사이트 구성



Home

Data Wiki

Share Hub

Colosseum

Forum



■ 사이트 관련 사진 출력 (5장, 5초마다 슬라이드 됨)

■ 공지사항

- 추가 데이터셋
- 경진대회

■ 인기 많은 글

- Data Wiki 중 상위 3개
- Colosseum 중 상위 3개
- Forum 중 상위 3개



III. 사이트 구성

■ 상명 Data Wiki

- 사이트 구성



Home

Data Wiki

Share Hub

Colosseum

Forum



■ Data Wiki

- 연구용 데이터셋 및 공공데이터 검색 및 링크 제공
- 이름순, Category순, task순, tutorial 유/무로 정렬 가능
- 데이터셋은 다음과 연계됨
 - Data Manager (1인)
 - 데이터셋 설명 (관리자 및 Data Manager가 작성 및 수정)
 - 해당 데이터셋을 활용한 Open SW
 - 데이터셋을 설명한 Blog 링크
 - 데이터셋 Tutorial Code

III. 사이트 구성

■ 상명 Data Wiki

- 사이트 구성



Home

Data Wiki

Share Hub

Colosseum

Forum



정렬기준

데이터 리스트

데이터 이름

- 데이터 분류
- 데이터의 수 (instances)
- 데이터의 목적 (task)
- 데이터 설명

데이터 링크 (iframe 연결) / 튜토리얼 링크 (iframe 연결)

-1

-2

-3

-4

-5

-.

-.

-.

-60

III. 사이트 구성

▪ 상명 Data Wiki

- 사이트 구성



Home

Data Wiki

Share Hub

Colosseum

Forum



▪ Share Hub

- 데이터를 다운 받은 사람이 활용하는 사이트
- 데이터셋의 유형에 따른 예제코드 정리
 - 자연어 (csv, text) / 이미지 (jpg, png) / 동영상 (avi, mp3) / 음성 (wav)
 - <https://www.tensorflow.org/tutorials?hl=ko>
 - <https://tutorials.pytorch.kr/>
 - 위 사이트와 연계
- 데이터 분석에 필요한 배경지식 & 실습자료 제공

III. 사이트 구성

■ 상명 Data Wiki

- 예상 사이트 구성



Home

Data Wiki

Share Hub

Colosseum

Forum



■ Forum

- 사이트 이용자들의 정보 공유 공간
- 데나무숲 : Data Wiki의 Data Manager 질의 응답 게시판(자유롭게)

- 이벤트 처리하기
- 폼 다루기
- 동적 웹 문서 만들기
- 다양한 방법으로 폼 다루기