**Ch11 이벤트**

11.1 이벤트 관련 용어 정리

* 이벤트란 키보드를 이용해 버튼을 입력하거나 마우스 클릭과 같이 다른 것에 영향을 미치는 것. 자바 스크립트에는 다음과 같은 이벤트가 있다
* 사전적 정의 : 자바 스크립트에서 DOM을 통해 동적인 화면처리를 위해 처리하는 내용
  + 마우스 이벤트, 키보드 이벤트, HTML 프레임 이벤트, HTML 입력 양식 이벤트, 유저 인터페이스 이벤트, 구조 변화 이벤트, 터치 이벤트

(ex) onclick, onmouseover, onmouseout, onkeyup(키보드의 특정키를 눌렀다가 올라올 때, event.keyCode 해당키의 고유 코드값..)

* 이벤트를 연결한다. Window 객체의 onload 속성에 함수 자료형을 할당하는 것
* load는 이벤트 이름, 이벤트 타입

<head>

<meta charset="utf-8" />

<title></title>

<script>

window.onload = function () {

var header = document.getElementById('header');

function whenClick() { alert('Click'); }

header.onclick = whenClick;

}

</script>

</head>

<body>

<h1 id="header">Click</h1>

</body>

* header 객체의 이벤트 이름은 click, 이벤트 속성은 onclick, 이벤트리스너는 whenClick()함수
* 문서 객체에 이벤트를 연결하는 방법을 이벤트 모델
  + DOM Level 0
    - **인라인 이벤트 모델**

<!DOCTYPE html>

<html lang="en" xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta charset="utf-8" />

<title></title>

<script>

function whenClick() {

alert('click');

}

</script>

</head>

<body>

<h1 id="header"

onclick="whenClick()">Click해봐</h1>

</body>

</html>

* + - **기본 이벤트 모델**

<!DOCTYPE html>

<html lang="en" xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta charset="utf-8" />

<title></title>

<script>

window.onload = function () {

var header = document.getElementById('header');

function whenClick() { alert('click'); }

header.onclick = whenClick;

// click 이벤트 onclick 이벤트 속성명

// whenClick 이벤트 리스너

};

</script>

</head>

<body>

<h1 id="header">Click해봐</h1>

</body>

</html>

* + - **고전적 이벤트 모델**

<!DOCTYPE html>

<html lang="en" xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta charset="utf-8" />

<title></title>

<script>

window.onload = function () {

var header = document.getElementById('header');

header.onclick = () => {

alert('click');

};

};

</script>

</head>

<body>

<h1 id="header">Click해봐</h1>

</body>

</html>

* + DOM Level 2
    - **표준 이벤트 모델**

<!DOCTYPE html>

<html lang="en" xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta charset="utf-8" />

<title></title>

<script>

window.onload = function () {

var header = document.getElementById('header');

header.addEventListener('click', () => {

alert('click');

});

};

</script>

</head>

<body>

<h1 id="header">Click해봐</h1>

</body>

</html>

* this의 사용

<html lang="ko">

  <head>

  <meta charset="UTF-8">

  <title></title>

  <script>

    window.onload = function () {

      var header = document.getElementsByClassName('header');

      for (var idx in header) {

        // 클릭한 header의 글자색과 border변경

        header[idx].onclick = function () {

          // this : 클릭이벤트가 발생된 컴포넌트 객체.this사용시 화살표 함수 사용 불가

          this.style.color = 'orange';

          this.style.border = '1px solid red';

        };

      }

    };

  </script>

</head>

<body>

  <h1 class="header">click1</h1>

  <h1 class="header">click2</h1>

</body>

</html>

11.2 이벤트 강제 실행

<html lang="en" xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta charset="utf-8" />

<title></title>

<script>

window.onload = function () {

var btnA = document.getElementById('btnA');

var btnB = document.getElementById('btnB');

var counterA = document.getElementById('counter-a');

var counterB = document.getElementById('counter-b');

btnA.onclick = function () {

counterA.innerHTML = Number(counterA.innerHTML) + 1;

};

btnB.onclick = function () {

counterB.innerHTML = Number(counterB.innerHTML) + 1;

**btnA.onclick(); //강제 이벤트 발생**

};

};

</script>

</head>

<body>

<button id="btnA">Button A</button>

<button id="btnB">Button B</button>

<h1>Button A - <span id="counter-a">0</span></h1>

<h1>Button B - <span id="counter-b">0</span></h1>

</body>

</html>

11.3 디폴트 이벤트 제거

* 일부 HTML 태그는 이미 이벤트 리스너가 있다. 이러한 것을 디폴트 이벤트라 한다. 에를들어 a태그를 클릭하면 다른 페이지로 이동하는 것. 이러한 디폴트 이벤트를 제거하는 방법을 살펴보자

<!DOCTYPE html>

<html lang="en" xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta charset="utf-8" />

<title></title>

<script>

window.onload = function () {

var btn = document.getElementById('btn');

btn.onclick = function () {

alert('경고창 한번만 떠');

btn.onclick = null; // 이벤트 제거

};

};

</script>

</head>

<body>

<button id="btn">BUTTON</button>

</body>

</html>

<head>

<meta charset="utf-8" />

<title></title>

<script>

window.onload = function () {

document.getElementById('myForm').onsubmit = function () {

var pass = document.getElementById('pass').value;

var passCheck = document.getElementById('passCheck').value;

if (pass == passCheck) {

alert('성공');

} else {

alert('비밀번호가 일치하지 않아요');

document.getElementById('pass').value = '';

document.getElementById('passCheck').value = '';

return false;

}

};

};

</script>

</head>

<body>

<form id="myForm">

<label for="name">아 이 디 : </label><input type="text" name="name" id="name" /><br />

<label for="pass">비밀번호 : </label><input type="password" name="pass" id="pass" /><br />

<label for="passCheck">비번확인 : </label><input type="password" name="passCheck" id="passCheck" /><br />

<input type="submit" value="제출" />

</form>

</body>

※ 인라인 이벤트 모델 – 디폴트 이벤트 제한

<form onsubmit=”return whenSubmit()”>

11.4 이벤트 전달 ; 어떠한 이벤트가 먼저 발생해 어떤 순서로 발생하는지를 이벤트 전달이라 한다. 자바스크립트는 자식 노드에서 부모 노드 순으로 이벤트를 실행한다. 이를 이벤트 버블링 방식이라 한다.

<!DOCTYPE html>

<html lang="en" xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta charset="utf-8" />

<title></title>

<style>

#a {border:2px solid red;padding:5px;}

#b {border:2px solid green;padding:5px;}

#h {border:2px solid blue;padding:5px;}

</style>

</head>

<body>

<div id="a" onclick="alert('a 영역')">

a영역

<div id="b" onclick="alert('a, b 영역')">

a, b 영역

<h1 id="h" onclick="alert('a, b, h영역')">a, b, h영역</h1>

</div>

</div>

</body>

</html>

* **event.keyCode(입력된 키 코드)**
* **event.offsetX와 event.offsetY(클릭한 곳의 offset좌표)**

**event.screenX와 event.screenY(클릭한 곳의 screen좌표)**

<!DOCTYPE html>

<html lang="en" xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta charset="utf-8" />

<title></title>

<script>

window.onkeyup = function () {

alert(event.keyCode);

};

window.onclick = function () {

document.body.innerHTML

= 'screen좌표:' + event.screenX + ',' + event.screenY + '<br>';

document.body.innerHTML

+= 'offset좌표:' + event.offsetX + ',' + event.offsetY;

};

</script>

</head>

<body>

</body>

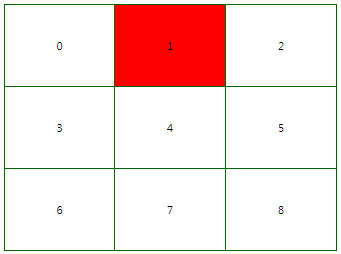
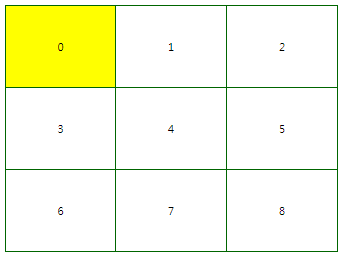
</html>

윗숫자 키보드 아스키코드 :48~56

옆숫자 키보드 아스키코드 : 96~104

ex1. 입력한 숫자 키보드의 해당 칸을 yellow로

ex2. 윗 숫자 키보드를 입력시 red바탕, 오른쪽 숫자 키보드를 입력시 yellow바탕 (s를 누르면 작동 시작,X를 누르면 작동 중지)



ex3. 화살표 키보드를 이용하여 선택된 td의 위치를 옮기고 현재 위치를 아래에 출력한다(힌트. 좌측화살표아스키(37),상화살표아스키(38), 우측화살표아스키(39), 하화살표아스키(40)



ex4. 버튼 위에 마우스를 올리면 버튼의 색상이 바뀌도록 페이지를 구현하시오



ex5. 좌측 아이콘이 마우스다운되면 우측과 같이 되도고, 마우스up되면 다시 원상복귀되는 페이지를 완성하시오.

ex6. 클릭한 곳의 좌표를 출력하고 이미지를 옮겨라

