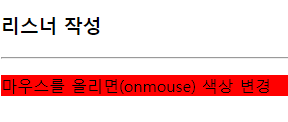
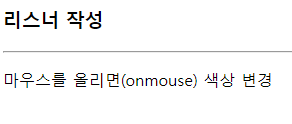
23-11-09

* html과 javascript를 연결하는 것 학습

**마우스 올리면 배경색상 변경되는 예제(exm01>11-09-01.html**)

마우스 올렸을 때 마우스 올리지 않았을 때

**방법 #1**

html 직접하는 방법

    <body>

        <p onmouseover = "this.style.backgroundColor = 'red'"

            onmouseout = "this.style.backgroundColor='white'">

            <!-- this를 하는 이유 스크립트 단이 아니라 여기서 처리하겠다는 것-->

            마우스를 올리면(onmouse) 색상 변경

        </p>

    </body>

This.style.backgroundColor -> C가 대문자로 해야지 변경사항이 적용됨  
onmouseover : 마우스가 위로 올라오면 ‘backgroundColor red’ 적용  
onmouseout : 마우스가 나가면 ‘backgroundColor를 white’ 적용  
this하면 여기에서 처리하겠다는 말이다 스크립트 단에서 하는 것이 아니라

**방법 #2**

명령어 처리 function처리  
head태그 안에 script를 만들었다. <script>

 <script>

            let p;

            function init(){ //init은 문서가 로드되었을 때 호출해라

            p=document.getElementById("p")//하위 문서에 있는 아이디p를 찾겠다는 것

            }

            function over(){

                p.style.backgroundColor = "red"

            }

            function out(){

                p.style.backgroundColor = "white"

            }

        </script>

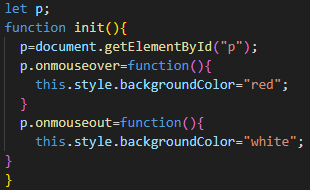
Init은 문서가 로드 되었을 때 호출하라는 것  
이렇게 써주면 html에 써주었던 onmouseover과 onmouseout을 제외해도 작동됨

<p id="p">마우스를 올리면(onmouse) 색상 변경</p>

그 대신 html에 p의 id=”p”를 넣어줘야지 작동됨

**스크립트안에 넣어주면 html의 작성양이 적어진다=>속도가 빨라진다**(이것 때문에 이렇게 하는 것)

<script>

  
function 을 활용함

**방법 #3**

EventListener을 사용하여 처리하는 방법

<script>

let p;

            function init(){

          p=document.getElementById("p")

p.addEventListener("mouseover", over);

                p.addEventListener("mouseout", out);

            function over(){

                p.style.backgroundColor = "red"

            }

            function out(){

                p.style.backgroundColor = "white"

            }

addEventListener을 사용하여 mouseover과 mouseout을 해준다.   
작업할 것들을 eventListener에 저장하는 것. 양이 많아지면 리스너 사용하는 것이 좋다.  
addEventListener은 어떠한 이벤트가 발생할 때 사용하며, 누적이 가능하다 (위의 2개 사용가능)

**방법 #4-1**

익명으로 처리하는 방법 -> 코드는 가장 짧은 코드이다  
그런데 익명으로 처리하다보니 나중에 어떤 이벤트 처리했는지 몰라서 찾을 수 없을 수도 있다.

<script>  
      function init(){ //init은 문서가 로드되었을 때 호출해라

                p=document.getElementById("p")

                p.onmouseover=function(){

                    this.style.backgroundColor ="red";

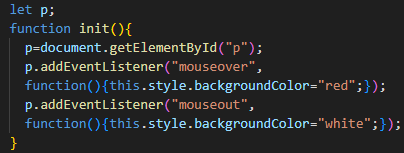
                }

                p.onmouseout = function(){

                    this.style.backgroundColor ="red";

                }

**방법 #4-2**  
익명으로 처리하는 것 중에서 eventListener사용하는 방법  
EventListener 와 function 을 활용하여 만들어줌

<script>

 function init(){

* init

스크립트 안에 있지만 처음에 이 파일을 읽지 않고 건너 뛴다.  
그리고 바디로 오면 그 내용들이 있다.   
그거를 먼저 읽어 들이고 init을 맨 나중에 읽으라고 body에 쓴 것

<script>부분

        <script>

            function init(){ //init은 문서가 로드되었을 때 호출해라

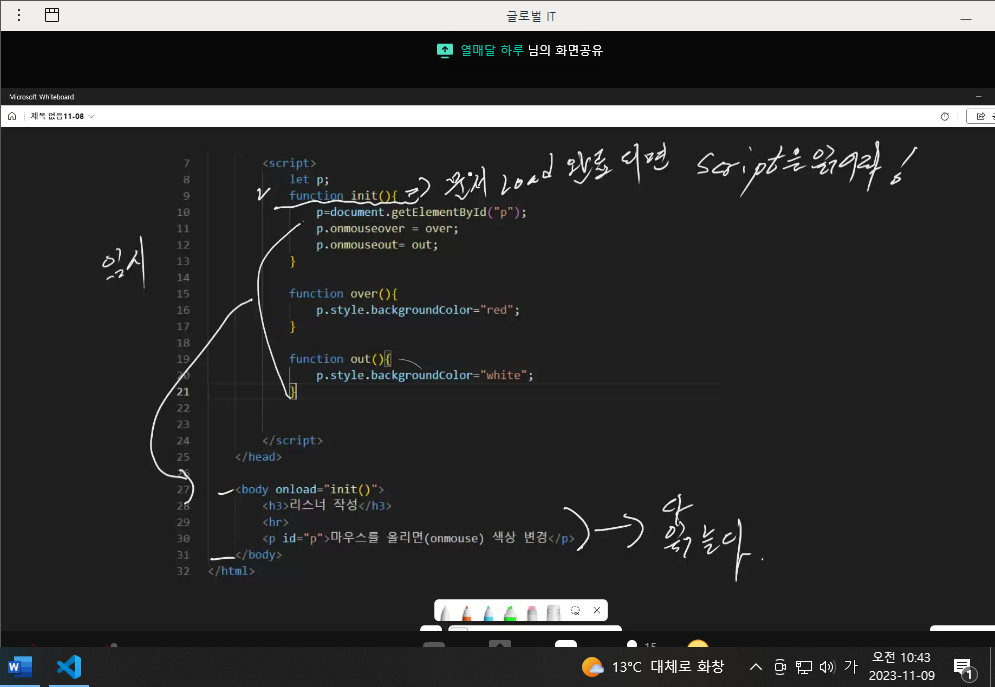
                p=document.getElementById("p")

                //하위 문서에 있는 아이디 p를 찾겠다는 것

            }

<body>부분

    <body onload="init()">



* Function init() => 문서 Load가 완료되면 Script를 읽어라!
* 스크립트 아래 부분은 임시로 읽은 것
* Body 부분을 다 읽는다 지금은 세줄밖에 없지만
* 그 다음에 onload=”init()”하면 임시로 읽었던 부분을 가지고 와서 작동하는 것

**마우스 올릴 때, 버튼 클릭할 때 확인창 뜨게 하기(exm01>11-09-02.html)**

<!DOCTYPE html>

<html>

    <head>

        <meta charset="utf-8">

        <title></title>

    </head>

    <body>

        <p id="p">마우스를 올려 보세요</p>

        <button onclick="f(event)">클릭해라</button>

        <script>

            function f(e) {

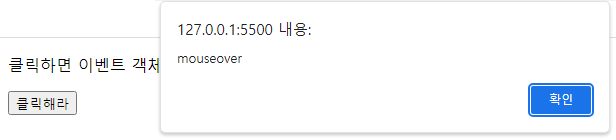
                alert(e.type);}

             document.getElementById("p").onmouseover=f;

         </script>

    </body>

</html>

마우스 올렸을 때 -> id=p를 지정한 것을 스크립트에서 불러내어(document.getElementById(“p”)  
onmouseover(마우스 올려지면)일 때 f함수 실행 => alert(e.type) : 파라미터의 타입을 보여주는 알림창을 띄우겠다는 것.

클릭해라 button을 눌렀을 때 -> html의 onclick은 클릭한 상황 f(event)는 클릭한 이벤트가 발생했을 때 f(e)의 스크립트 부분을 실행 => 알림창이 뜨면서 e.type 출력 (클릭할 때 ‘click’ 뜸)

**alert : 알림창을 띄운다**

스크립트는 head에 넣어도 되지만 body의 맨 마지막에 넣어도 됨

 document.getElementById("p").onmouseover=f;

(head에 넣을 경우 mouseover이 작동되지 않음 => 객체만 불러오는 것만 되기 때문.)  
\* 이 부분에서 onmouseover=f를 onmouseover-f라고 쓰고 있었다 f의 값을 onmouseover실행되었을 때 실행한다는 것이다.

작동하는 것은 새로운 객체를 불러오고 있는 것

클릭해라 하고 창 뜨는 것 -> 자료전송의 시작이다

스크립트에서 자료 이벤트 발생한 것에 대한 내용을 전달하고 있는 것

새로운 창이 새로운 객체를 불러오고 있는 중인 것

<onclik과 addEventListener차이>

<https://velog.io/@won11/JS-Onclick%EA%B3%BC-Click-Event-%EC%B0%A8%EC%9D%B4%EC%A0%90>

<https://blog.naver.com/isaac7263/221739562878>

    <body>

        <p id="p">클릭하면 이벤트 객체 출력</p>

        <button onclick="f(event)">클릭해라</button>

        <script>

            function f(e){ //작업할 e를 만들고 객체 생성함

                let text = "type : " + e.type + "<br>"

                + "target : " + e.target + "<br>"

                + "currentTarget : " + e.currentTarget + "<br>"

                + "defaultPrevented : " + e.defaultPrevented;

                let p = document.getElementById("p");

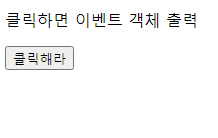
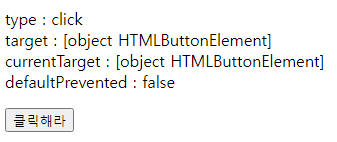
                p.innerHTML = text;

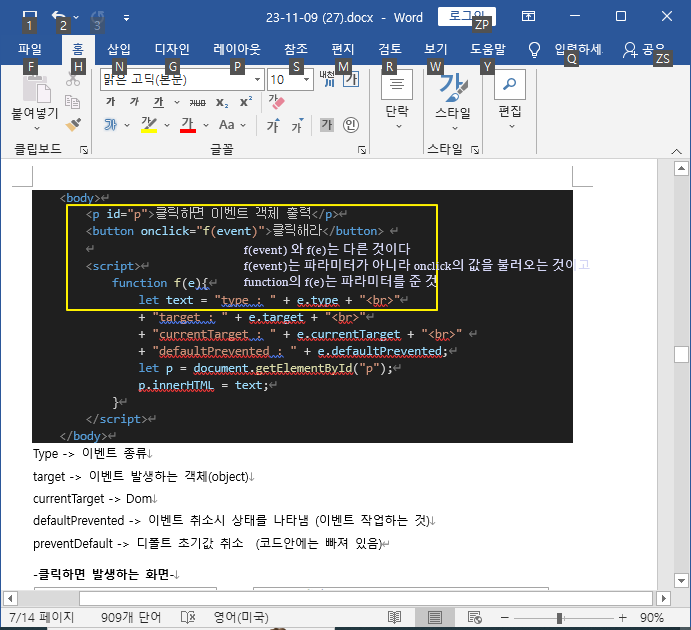
            }

        </script>

    </body>

Type -> 이벤트 종류  
target -> 이벤트 발생하는 객체(object)  
currentTarget -> Dom (뒤에서 다룬 내용)  
defaultPrevented -> 이벤트 취소시 상태를 나타냄 (이벤트 작업하는 것)  
preventDefault -> 디폴트 초기값 취소 (코드안에는 빠져 있음)

**-클릭하면 발생하는 화면-**  
 **→** 



**exm01>11-09-03.html**

    <body>

        <h3>이벤트 디폴트 취소</h3>

        <hr>

        <a href="https://www.naver.com">네이버 이동</a>

        <!-- 네이버로 갈지 안갈지 물어보는 아이 만들거다 -->

        <hr>

        <form>

            <input type="checkbox">빵<br>

            <input type="checkbox" onclick="noAction(event)">밥

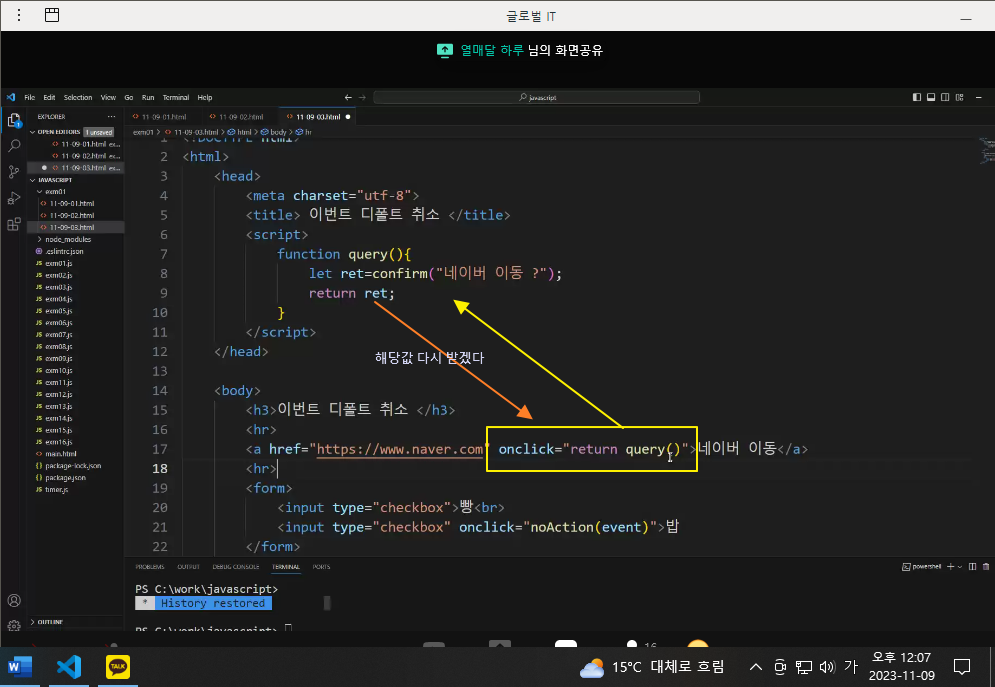
        </form>

    </body>

onclick="return query()"

갈건지 안갈건지 물어보는 아이

Query를 스크립트안에 함수로 지정해준다.



        <script>

            function query(){

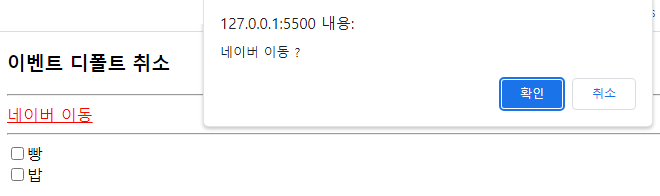
                let ret=confirm("네이버 이동 ?");

//메세지 창에 네이버이동할건지 물어보는 것

                return ret; //return을 해주지 않으면 취소눌러도 네이버로 이동됨

            }

        </script>



확인을 누르면 True이고 홈페이지로 이동하고, 취소를 누르면 false로 취소된다.

로그인창 만들 때 활용가능 !

//<script>부분

function noAction(e){

                e.preventDefault();

            }

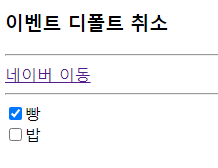
//<body>부분

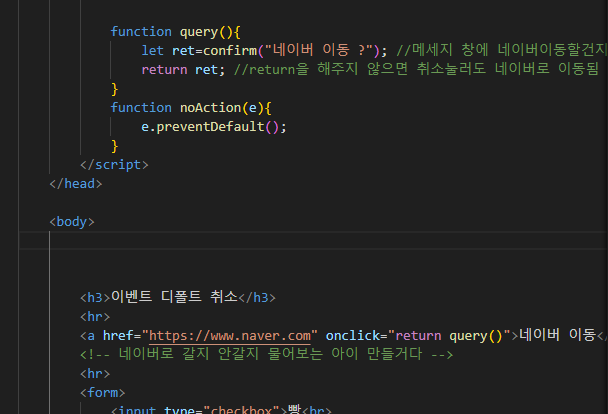
<form>

            <input type="checkbox">빵<br>

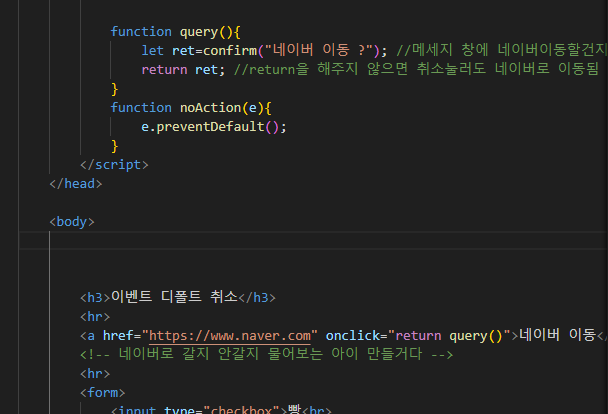
            <input type="checkbox" onclick="noAction(event)">밥

        </form>

noAction해준 부분인 밥이 클릭되지 않음



Return ret; 을 해주면 확인과 취소의 true의 값과 false의 값을 return 반환해준다.



**Return query()**는 그 함수를 실행한다고 생각하기 보다는 query함수의 실행한 ret의 값을 받아와서 확인을 누르면 그 a링크가 활성화 되고 naver로 이동하게 되는 것이다.  
(true,false의 값을 가지고 와서 True가 되면 링크에 걸어둔 네이버 사이트가 연결이 된다.)

**exm01>11-09-04.html**

        <h3>문장검사</h3>

        <hr>

        <p style="color:aqua;">이건<span style="color: red;">문장이다</span></p>

        <form>

            <input type="text">

            <input type="button" value="테스트" id="button">

        </form>

Id를 지정해주면 누군가 찾으려고 할 때 찾을 수 있다.

<개념>

Input click하면 생성된다.

웹브라우저가 작동할 때 document를 생성한다

맨 처음 도큐먼트 생성하는 아이를 DOM이라고 부른다

* DOM : Document Object Model, 문서 객체 모델  
  Dom의 구조는 브라우저와 HTML문서를 이해 할 수 있게 구성된 것  
  Dom의 구조가 만들어져야

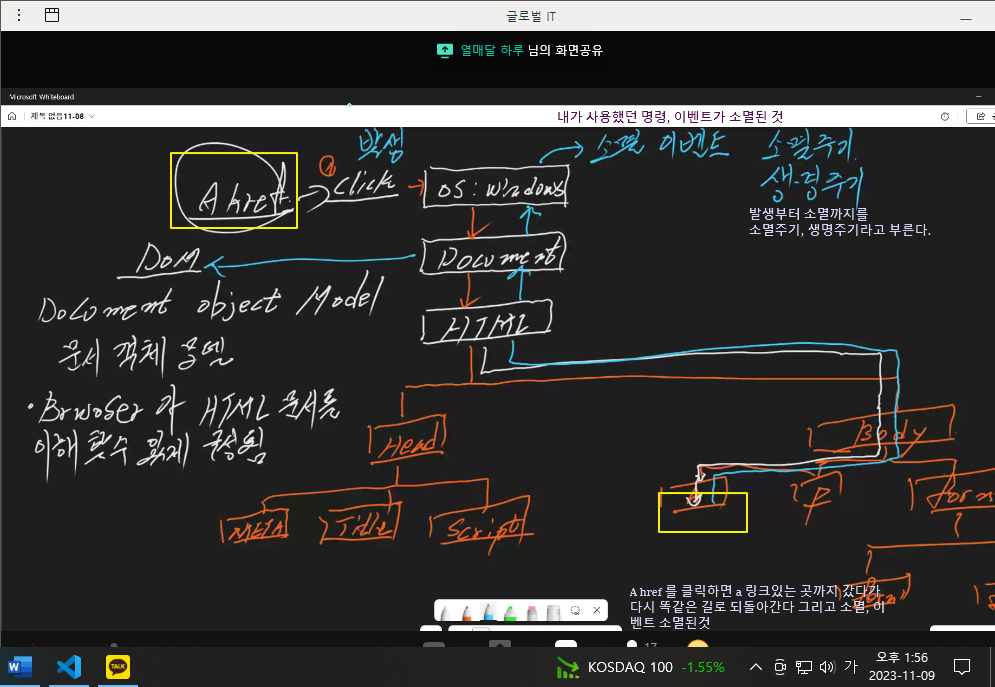
클릭하는 순간 윈도우 가 document를 만든다 => 그건 즉 DOM을 만든다는 것   
-> document가 HTML을 읽는다

HTML 구조는 head와 body로 이루어져 있다.

Body의 구조는 header, footer, 이런 것들이 나온다. Tree 구조처럼 만들어진다.

Tree 구조처럼 만들어진 것이 document의 구조이다

* 우리가 만든 코드의 구조

Head > meta, title, script  
body > a(링크), p(태그), form

Form > 1번 form

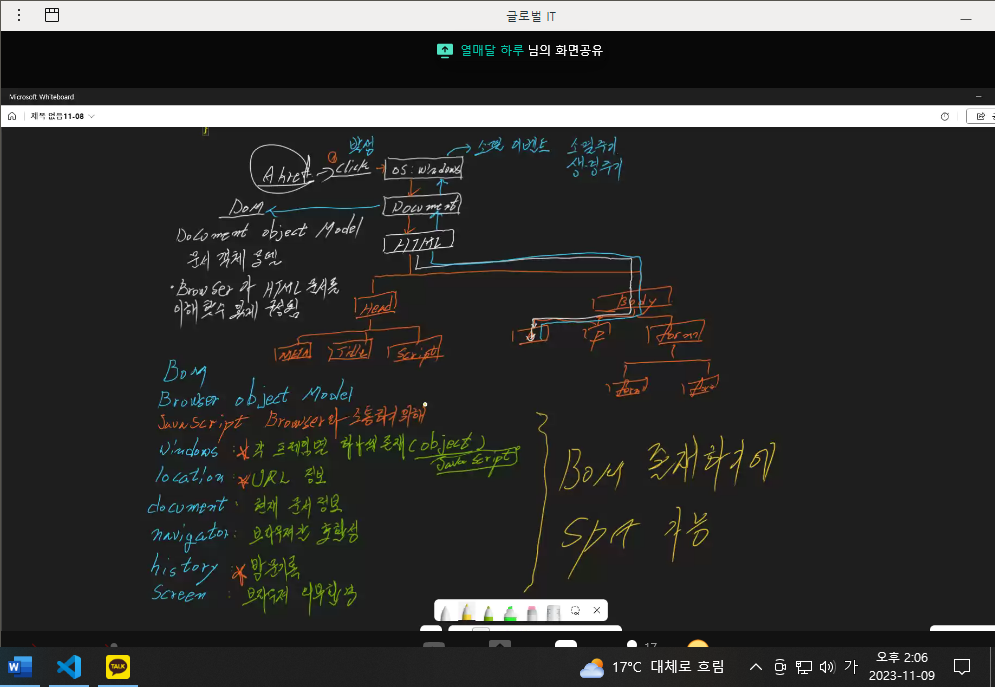
2번 form

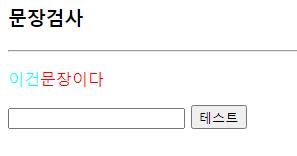
* BOM : Browser Object Model  
  JavaScript가 Browser과 소통하기 위해 필요한 것이 Bom이다.  
  Bom은 Dom이 생성된 이후에 생성된다.

**Bom이 존재하기에 SPA가 가능하다 (동적기능이 가능)**

Bom의 종류

* + Windows : 각 프레임 별 하나씩 존재 (즉, JavaScript의 **object** 말하는 것)
  + Location : URL 정보
  + Document : 현재 문서 정보
  + Navigator : 브라우져간 호환성
  + History : 방문기록
  + Screen : 브라우져 외부환경

  
DOM의 구조

  
텍스트 클릭할 때 밑에 문장이 나오도록 설정하는 것  
클릭할 때 생기는 이벤트 기록을 capture할거야 사진찍을거라는 것

<script>

            let div = document.getElementById("div");

//이벤트 메세지가 출력될 공간, div실행공간, div안에 있는곳에 작업할거다

            let button = document.getElementById("button"); //button실행공간

            document.body.addEventListener("click", capture, true);

//클릭할 때 생기는 이벤트 기록을 capture할거야 사진찍을거라는 것

            //작업할것을 기억할거야 이벤트리스너 -> 위치 문서에 body에 있는 곳에 이벤트 실행할거야

            document.body.addEventListener("click", bubble, false)

            button.addEventListener("click", bubble, false);

            function capture(e) { //캡처하고 버블했던 것을 출력하는 부분

                let obj = e.currentTarget;

//currentTarget 이벤트 발생한 것을 dom안에 기록하라는 것

                let tagName = obj.tagName;

//태그네임은 obf안에 있는 tagname가지고 해라

                div.innerHTML += "<br> capture 단계 : " + tagName + "태그" + e.type + "이벤트"

//html문서에 작업해 그런데 추가하면셔 ,연속해서 작업

            }

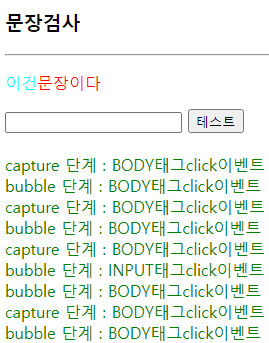
            function bubble(e) {

                let obj = e.currentTarget; //목표점이 무엇이냐고 물어본거다

                let tagName = obj.tagName;

                div.innerHTML += "<br> bubble 단계 : " + tagName + "태그" + e.type + "이벤트"

            }

        </script>

이벤트 리스너의 구조 흐름이다.

**Exm01 > 11-09-05 cal.html**

<!DOCTYPE html>

<html lang="en">

    <head>

        <meta charset="UTF-8" />

        <title>calculate</title>

        <script>

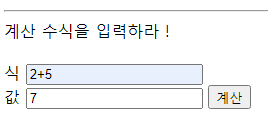
            function calculate(){

                let exp = document.getElementById("exp");

                let result = document.getElementById("result");

                result.value = eval(exp.value);

            }

        </script>

    </head>

    <body>

        <h3></h3><hr>

        계산수식을 입력

        <br><br>

        <form>

        식 : <input type="text" id="exp" value=""><br>

        값 : <input type="text" id="result" value="">

        <input type="button" value="계산" onclick="calculate()">

        </form>

    </body>

</html>

Exp에 html에 id지정한 exp의 값을 받아온다.  
result에 html에 id지정한 result의 값을 받아온다.  
**eval이라는 것은 값을 계산해주는 함수이다.** (찾아보니까 eval의 문제점이 있다는 것을 발견, 그래서 간단한 계산할 때는 괜찮지만 ,안전성의 문제를 기억하고 주의해야 할 것)

해결한점 : 코드를 똑같이 작성했는데 아무리 봐도 틀린 것이 보이지 않았다. 그런데 계산값에 결과가 뜨지 않아서 똑같은 코드를 그대로 다시 입력했더니 결과값 출력됨

궁금한점 : 값에는 입력하는 부분이 없는데 값을 getElementById로 가지고 오는 이유가 무엇인가  
아니면 받은 값을 다시 전달하는 부분인가  
result.valud 값이 나온 것이 어떻게 출력되는 것일까?

**Exm01 > 11-09-06-onload.html**

<!DOCTYPE html>

<html>

    <head>

        <meta charset="utf-8">

        <title></title>

        <script>

            function changeImage(){

                let sel = document.getElementById("sel");

                let img = document.getElementById("myimg");

                img.onload = function() {

//빈 객체 만들거야 이미지 크기를 계속 출력하기 위해서

                    let myspan = document.getElementById("myspan");

                    myspan.innerHTML=img.width + img.height;

                }

                let index = sel.selctionindex;

                img.src = sel.options[index].value;

            }

        </script>

    </head>

    <body onload="changeImage()">

        <h3> 이미지 출력 </h3><hr>

        <form>

            <select id="sel" onchange="changeImage()">

                <option value="img/apple.png">apple</option>

                <option value="img/banana.png">banana</option>

                <option value="img/mango.png">mango</option>

            </select>

            <span id="myspan">이미지 크기</span>

        </form>

        <p><img id="myimg" src="img/apple.png" alt ="."></p>

        <!-- alt를 해줘야 사진이 없어도 x자로 나오지 않음 -->

    </body>

</html>

이 위에 것이 아래와 같이 바뀌니까 작동 되었다.

아이디 표기법 지키기

<!DOCTYPE html>

<html>

    <head>

        <meta charset="utf-8">

        <title></title>

        <script>

            function changeImage(){

                let sel = document.getElementById("sel");

                let img = document.getElementById("myImg");

                img.onload = function() {

//빈 객체 만들거야 이미지 크기를 계속 출력하기 위해서

                    let mySpan = document.getElementById("mySpan");

                    mySpan.innerHTML=img.width + "X" + img.height;

                }

                let index = sel.selectedIndex;

//사과인지 바나나인지 선택한거 들어오기

                img.src = sel.options[index].value;

            }

        </script>

    </head>

    <body onload="changeImage()">

        <h3> 이미지 출력 </h3>

        <hr>

        <form>

            <select id="sel" onchange="changeImage()">

                <option value="img/apple.png">apple</option>

                <option value="img/banana.png">banana</option>

                <option value="img/mango.png">mango</option>

            </select>

            <span id="mySpan">이미지 크기</span>

        </form>

        <p><img id="myImg" src="img/apple.png" alt ="."></p>

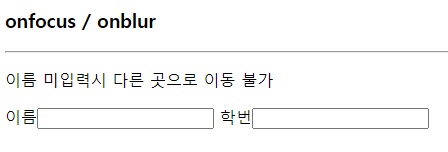
        <!-- alt를 해줘야 사진이 없어도 x자로 나오지 않음 -->

    </body>

</html>

**Exm01 > 11-09-07.html**

Onfocu가 이름에 값을 입력되지 않으면 다른 항목으로 이동되지 않음

// 회원가입 시 필수항목에 사용한다

<!DOCTYPE html>

<html>

    <head>

        <meta charset="utf-8">

        <title></title>

        <script>

            function checkFilled(obj) {

                if(obj.value == ""){ //object의 값이 없으면

                    obj.focus(); } }

            function findChecked() {

//findcheck를 누르면 무엇을 선택을 했는지 메세지창으로 보여주려고 한다

                let found = null;

                let kc = document.getElementsByName("city");

                //아래 아이디 주지 않았기 때문에 name로 쓰는 것, 아래 id를 줬으면 id로 쓰면됨

                for(let i=0; i<kc.length; i++) {

                    if(kc[i].checked==true)

                    found=kc[i]; } //이 값이 될것, 도시는 선택한 것

                if(found != null)

                alert(found.value + "선택됨."); //found에서 찾은 값을

                else

                alert("미선택"); }

        </script>

    </head>

    <body onload="document.getElementById(name).focus()">

        <!-- focus는 초점 맞추라는 것 위치값 들어가지 않음 -->

        <h3>onfocus / onblur</h3><hr>

        <p>이름 미입력시 다른 곳으로 이동 불가</p>

        <form>

            이름<input type="text" id="name" onblur="checkFilled(this)">

            <!-- this쓰는 이유는 스크립트에 있는 값이 아닌 여기에서 바로 사용할 때 -->

            학번<input type="text">

        </form>

        <h3>라디오 버튼</h3><hr>

        <form>

            <input type="radio" name="city" value="seoul" checked>서울

            <input type="radio" name="city" value="busan"> 부산

            <input type="radio" name="city" value="suwon"> 수원

            <input type="button" value="find checked" onclick="findChecked()">

        </form>

    </body>

</html>

<코딩 파일명 작성법>

1. **Camel 표기법** : 첫문자는 소문자, 이후 연결된 문자의 첫 글자는 대문자  
   ex) my span => mySpan 으로 작성하기
2. **Kebab 표기법** : 모든 문자는 소문자이고, 문자 연결은 “-” 하이픈  
   ex) my span => my-span
3. **Snake 표기법** : 모든 문자는 소문자이고, 문자 연결은 “\_” 언더바  
   ex) my span => my\_span
4. **Pascal 표기법** : 모든 문자의 첫 글자는 대문자  
   ex) my span => MySpan  
   \* 이건 지금 사용하지 않음, 모든 첫 글자가 대문자 이기 때문에 어디서부터 시작하는지 구분할 수 없다. 그래서 camel법 사용한다

* **Camel 표기법을 가장 많이 사용한다.**