

2023-11-09

```
<!DOCTYPE html>
<html lang="en">

  <head>
    <meta charset="UTF-8">
    <title>title</title>
    <script>
      let p;
      function init(){ /* 자바스크립트를 불러와라, 문서 load 완료되면 script를 읽어라 */
        p=document.getElementById("p"); /*getElementById id는 p를 찾겠다*/
        // p.onmouseover = over;
        // p.onmouseout = out;
        p.addEventListener("mouseover", over);
        p.addEventListener("mouseout", out); /*작업해야 할 내용들의 기능들을 addEventListener라는 임시 기록장치에 기록 시켜놓음, 위와 같지만 작
      }
      function over(){
        p.style.backgroundColor="red";
      }
      function out(){
        p.style.backgroundColor="white";
      }
    </script>
  </head>

  <body onload="init()">
    <h3>리스너 작성</h3>
    <hr>
    <p id="p">마우스를 올리면(onmouse) 색상 변경</p>
  </body>
</html>
```

```
<p OnMouseOver="this.style.backgroundColor='red'"
  OnMouseOut="this.style.backgroundColor='white'">
  마우스를 올리면(onmouse) 색상 변경</p>
```

아래는 html에 직접 꽃아 넣은거고 위는 스크립트 처리해서 넣음 (작업할게 적어서 스크립트에 넣지 않아도 될 때 사용)

<body onload="init()"> init에 관련된 내용을 마지막 읽어라. (스크립트를 임시로 읽고 이후 body내용도 읽기 때문에 onload="init()"을 안 쓰면 body가 먼저 처리됨)

```
//익명으로 처리하는 이 방법이 코드가 가장 짧다. 익명이라 찾을수 없을수도 있는 단점이 있음, 이벤트 기록이 없으면 뒤로가기가 안된다.
p.addEventListener("mouseout",
  function(){this.style.backgroundColor="red";})

p.onmouseout=function(){
  this.style.backgroundColor="white";
}
```

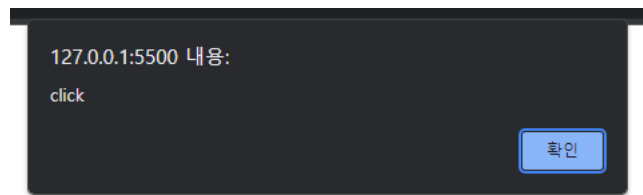
```
<body>
  <p id="p">마우스를 올려보세요.</p>
  <button onclick="f(event)">클릭해라</button> <!-- onclick -> 애플 누르면 실행 -->

  <script> //html 상에서 작업할때는 body 맨 마지막에 넣어도 된다.
    function f(e){
      alert(e.type);
    }
    document.getElementById("p").onmouseover=f; //-> //f일때 작동한다, 객체를 전달한다.
  </script>
</body>
```

document → 지금 내 html문서

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

지금 내 html문서에서 p를 찾는다. f일때 onmouseover 작동한다(객체를 전달한다.)



객체를 전달하는 중인것.

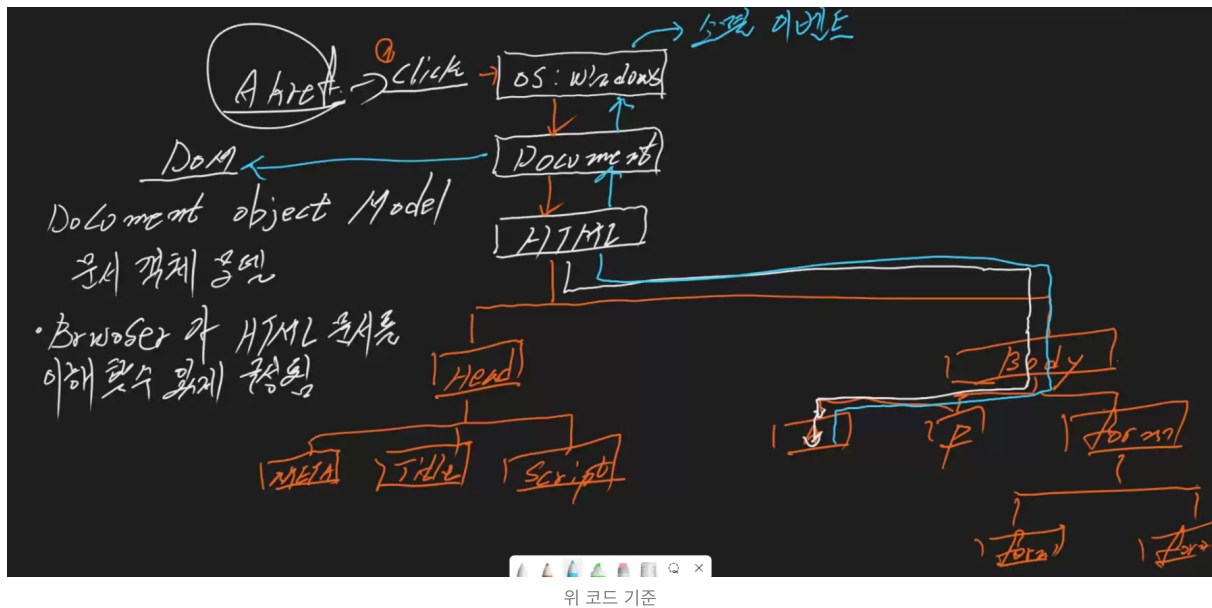
```
<body>
  <p id="p">클릭하면 이벤트 객체 출력</p>
  <button onclick="f(event)">클릭해라</button> <!-- onclick -> 애플 누르면 실행 -->

  <script> //html 상에서 작업할때는 body 맨 마지막에 넣어도 된다.
    // function f(e){
    //   alert(e.type);
    // }
    // document.getElementById("p").onmouseover=f; //-> //f일때 작동한다, 객체를 전달한다.
    function f(e){
      let text = "type : " + e.type + "<br>" //type -> 이벤트 종류(click, load 등)
      + "target : " + e.target + "<br>" //target -> 이벤트 발생하는 오브젝트
      + "currentTarget : " + e.currentTarget + "<br>" //currentTarget -> dom
      + "defaultPrevented : " + e.defaultPrevented; //defaultPrevented -> 이벤트 취소 시 상태
      let p=document.getElementById("p"); //preventDefault -> 디폴트 초기값 취소
      p.innerHTML = text;
    }
  </script>
</body>
```

```
<head>
  <meta charset="UTF-8">
  <title>이벤트 디폴트 취소</title>
  <script>
    function query(){
      let ret=confirm("네이버 이동????????????"); //네이버로 이동할건지 메시지창의 내용
      return ret;
    }
    function noAction(e){
      e.preventDefault();
    }
  </script>
</head>

<body>
  <h3>이벤트 디폴트 취소</h3>
  <hr>
  <a href="https://www.naver.com/" onclick="return query()">네이버 이동</a>
  <hr>
  <form>
    <input type="checkbox">땡<br>
    <input type="checkbox" onclick="noAction(event)">뱌
  </form>
</body>
```

onclick="return query()" → 취소할지 확인하는 창을 만들겠다는 것



document를 생성하는 애를 dom이라 부른다.

dom → document object model(문서 객체 모델이라 부름)

- browser가 html 문서를 이해할 수 있게 구성 된 것

발생부터 소멸까지를 소멸주기, 생명주기 라고 한다.

bom → browser object model

- javascript browser와 소통하기 위해

*windows : 각 프레임 별 하나 씩 존재 (javascript의object)

*location : URL 정보

document : 현재 문서 정보

navigator : 브라우저 간 호환성

*history : 방문 기록

screen : 브라우저 외부 환경

(bom이 존재하기에 spa(싱글 페이지 어플리케이션) 가능하다. (동적 처리 가능)

```
<!DOCTYPE html>
<html lang="en">

  <head>
    <meta charset="UTF-8">
    <title></title>
    <script>
      function changeImage(){
        let sel = document.getElementById("sel");
        let img = document.getElementById("myImg");
        img.onload = function(){ //이미지 크기를 계속 출력하기 위해 객체를 만들
          let mySpan = document.getElementById("mySpan");
          mySpan.innerHTML = img.width + "x" + img.height;
        }

        let index=sel.selectedIndex;
        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">사과</option>
    <option value="img\banana.png">바나나</option>
    <option value="img\mango.png">망고</option>
  </select>
  <span id="mySpan">이미지 크기</span>
</form>
<p></p>
</body>
</html>
```

[코딩 파일명 작성법]

1. Camel - 첫 문자는 소문자 이후 연결된 문자의 첫 글자는 대문자 (많이 사용)
ex) myspan → mySpan
2. Kebab - 모든 문자는 소문자이고 문자 연결은 "-"(하이픈)
ex) myspan → my-span
3. Snake - 모든 문자는 소문자이고 문자 연결은 "_"(언더바)
ex) myspan → my_span
4. Pascal - 모든 문자의 첫 글자는 대문자 (사용X)
ex) myspan → MySpan