Javascript_이벤트

자바스크립트 이벤트(event)는, 사용자의 마우스 조작이나 키보드 조작을 이벤트 라고 합니다. 이벤트는 말 그대로 어떤 사건을 뜻하는 단어이긴 합니다.

1. 이벤트란?

- 어떤 버튼 클릭시 무언가가 실행
- 문서가 로드되고 나면 3초후 팝업이 나타남
- 클릭이나 키보드 조작을 통해 어떠한 요소를 제어

2. 이벤트 타입

웹 서비스에서는 주로 마우스 이벤트와 키보드 이벤트를 사용하는것이 일반적입니다.사실, 이벤트의 종류는 굉장히 많습니다. 아래 링크를 통해 그 목록을 확인해보세요

▼ 마우스 이벤트 관련

★ 마우스 이벤트들
onauxclic : 마우스 왼쪽버튼을 제외한 다른 버튼을 클릭했을 때 작동한다. (오른쪽 버튼, 휠버튼, 다른 매크로 버튼 등)
onclick : 요소 위에서 클릭했을 때 이벤트 발생
oncontextmenu : 컨텍스트 메뉴를 열기 위해서 마우스 오른쪽 버튼을 눌렀을 때 이벤트 발생
ondblclick : 요소 위에서 더블 클릭 했을 때 이벤트 발생
onmousedown : 요소 위에서 마우스 버튼을 누룰 때 발생
onmouseenter : 포인터가 요소로 들어갈 때 이벤트 발생
onmouseleave : 포인터가 요소 밖으로 나갈 때 이벤트 발생
onmousemove : 요소 위에서 포인터가 이동 중일 때 이벤트 발생
onmouseover : 포인터가 요소 위로 올라왔을 때 또는 자식 요소 위로 올라왔을 때 이벤트 발생
onmouseout : 마우스 포인터가 요소 밖으로 나가거나 자식 요소 밖으로 나갔을 때 이벤트 발생
onmouseup : 마우스 버튼을 떼었을 때 이벤트 발생
onselect: text가 선택됐을 때(드래그) 작동된다. 이 event는 element가 <input>태그 중 type이 text, textarea일 때만 작동한다.
wonheel: 사용자가 휠을 작동시킬 때 작동한다.

★ 휠 이벤트 객체
deltaX : 마우스 휠의 수평 스크롤의 양 반환 (x-축)
deltaY : 마우싀 휠의 수직 스크롤의 양 반환 (y-축)
deltaZ : 마우스 휠의 z-축에 대한 스크롤의 양 반환
deltaMode : 델타 값의 측정 단위 표시 (픽셀, 라인, 페이지)

★ 마우스 이벤트 객체
altKey : 마우스 이벤트가 발생했을 때 ALT 키가 눌렸는지 반환
button : 마우스 이벤트가 발생했을 때 마우스 버튼이 눌렸는지 반환
buttons : 마우스 이벤트가 발생했을 때 마우스 버튼들이 눌렸는지 반환
clientX : 마우스 이벤트가 발생했을 때 현재 윈도우에 대한 마우스 포인터의 수평 좌표 반환
clientY : 마우스 이벤트가 발생했을 때 현재 윈도우에 대한 마우스 포인터의 수직 좌표 반환
ctrlKey : 마우스 이벤트가 발생했을 때 CTRL 키가 눌렸는지 반환
detail : 마우스가 몇번이나 클릭되었는지 수 반환
metaKey : 마우스 이벤트가 발생했을 때 META 키가 눌렸는지 반환
relatedTarget : 마우스 이벤트를 일으킨 요소와 관련된 요소 반환
screenX : 마우스 이벤트가 발생했을 때 화면에 대한 마우스 포인터의 수평 좌표 반환
screenY : 마우스 이벤트가 발생했을 때 화면에 대한 마우스 포인터의 수직 좌표 반환
shiftKey : 마우스 이벤트가 발생했을 때 SHIFT 키가 눌렸는지 반환
which : 마우스 이벤트가 발생했을 때 어떤 마우스 버튼이 눌렸는지 반환

★ 드래그 & 드랍 이벤트
ondrag : 드래그되고 있을 때

ondragend : 드래그를 끝마쳤을 때
ondragenter : 드래그되는 요소가 드랍 타겟에 들어갔을 때
ondragstart : 드래그를 시작 했을 때
ondragleave : 드래그되는 요소가 드랍 타겟에서 나왔을 때
ondragover : 드래그되는 요소가 드랍 타켓 위에 있을 때
ondrop : 드래그되는 요소가 드랍 타켓 위에 떨어졌을 때

▼ 키보드 이벤트 관련

★ 키보드 이벤트

onkeydown : ANY key is pressed(키보드를 누루고 있을 때)

onkeypress : ANY key (except Shift, Fn, or CapsLock) is in a pressed position (fired continuously). (키보드를 지속적으로 누르고 있을때)

onkeyup : ANY key is released(키보드를 누른 후 떼었을 때)

발생 순서 : 1. onkeydown 2. onkeypress 3. onkeyup

★ 키보드 이벤트 객체

altKey : 키보드 이벤트가 발생했을 때 ALT 키가 눌렸는지 반환

ctrlKey : 키보드 이벤트가 발생했을 때 SHIFT 키가 눌렸는지 반환

charCode : onkeypress 이벤트를 발생시킨 키의 유니코드 문자 코드 반환

key : 그 이벤트로 표시되는 키 값 표시

keyCode : onkeypress를 발생시킨 키의 유니코드 문자 코드 반환, onkeydown 또는 onkeyup 을 일으킨 유니코드 키 코드 반환

location : 키보드 또는 디바이스 위의 키의 위치 리턴

metaKey : 키보드 이벤트가 발생했을 때 META 키가 눌렸는지 반환

shiftKey : 키보드 이벤트가 발생했을 때 SHIFT 키가 눌렸는지 반환

which : onkeypress를 발생시킨 키의 유니코드 문자 코드 반환, onkeydown 또는 onkeyup 을 일으킨 유니코드 키 코드 반환

이벤트 참조

DOM 이벤트는 발생한 흥미로운 것을 코드에 알리기 위해 전달됩니다. 각 이벤트는 Event 인터페이스를 기반으로한 객체에 의해 표현되며 발생한 것에 대한 부가적인 정보를 얻는데 사용되는 추가적인 커스텀 필드 또는 함수를 가질수도 있습니다. 이벤트는 렌더링 모델에서 기본적인 사용자 인터렉션부터 발생한 것에대한 자동 알림까지 모든 것을 나타낼 수 있습니다.

https://developer.mozilla.org/ko/docs/Web/Events

3. 이벤트 처리방법

이벤트 발생시 이것을 처리하는 함수를 이벤트 리스너 또는 이벤트 핸들러 라고 합니다.

- 이벤트 리스너(listener)
- 이벤트 핸들러(handler)

이벤트 등록 방식은 크게 3가지로 구분합니다. 장, 단점을 파악하고 사용에 유의하기 바랍니다.

(1) 인라인 방식

```
<!doctype html>
<html lang="ko">
<head>
  <meta charset="utf-8">
</head>
<body>
  <button onclick="test()">인라인 방식 핸들러</button>
<!--  <button onclick="alert('hello javascript');">인라인 방식 핸들러</button> -->
  <script>
  function test(){
    alert('hello javascript');
  }
```

```
  </script>
</body>
</html>
```

(2) 프로퍼티방식 : 자바스크립트 코드에서 프로퍼티로 등록해서 사용하는 방식이며, 인라인 방식과 다름 또한 하나의 이벤트 핸들러 프로퍼티에 하나의 이벤트 핸들러만 바인딩이 가능합니다.

```
<!doctype html>
<html lang="ko">
<head>
  <meta charset="utf-8">
</head>
<body>
  <button id="test">프로퍼티 방식 핸들러</button>
  <script>
    // 이벤트 바인딩(binding) : 부착
    // DOM 객체에 접근하기 위해 .querySelector()를 사용했습니다.
    let testBtn = document.querySelector("#test");

    testBtn.onclick = function(){
      alert("hello javascript1");
    }
    // 하나의 이벤트 핸들러만 바인딩이 가능하므로, 아래의 함수가 실행됩니다.
    testBtn.onclick = function(){
      alert("hello javascript2");
    }
  </script>
</body>
</html>
```

(3) addEventListener(), attachEvent() 방식 - 하나 이상의 이벤트 핸들러를 바인딩 할 수 있습니다.

```
/* 기본형
대상객체.addEventListener('이벤트 타입', 함수명[,이벤트 전파방식]);
*/

// 주의1. IE9 이상에서 동작하는 방식(IE8 이하는 attatchEvent를 사용해야 함)
// 주의2. 프로퍼티 방식과 다르게 이벤트 타입을 작성, 예)onclick --> click 으로 작성
```

실제 코드는 아래와 같습니다.

```
<!doctype html>
<html lang="ko">
<head>
  <meta charset="utf-8">
</head>
<body>
  <button id="test">addEventListener 방식</button>
  <button id="removeTest">이벤트 삭제</button>
  <script>
    let testBtn = document.getElementById("test");
    let removeBtn = document.getElementById("removeTest");

    function testAlarm(){
      alert('hello javascript');
    }

    function testAlarm2(){
      alert('HELLO JAVASCRIPT');
    }

    testBtn.addEventListener('click', testAlarm); // 마우스 왼쪽 버튼클릭
    testBtn.addEventListener('contextmenu', testAlarm2); // 마우스 오른쪽버튼 클릭

    // .removeEventListener() 사용시 등록된 이벤트 리스너를 삭제합니다.
    function removeEvent() {
      testBtn.removeEventListener('click', testAlarm);
      testBtn.removeEventListener('contextmenu', testAlarm2);
    }

    removeBtn.addEventListener('click', removeEvent);
  </script>
```

```
</body>
</html>
```

그러면, 이벤트와 함수를 이용해서 웹서비스에서 자주 사용하는 컨텐츠를 제작해보겠습니다.

① 팝업

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>자바스크립트</title>
    <style>
        .popup_main h2 {
            background:#333;
            padding:5px 10px;
            font-size:1.5rem;
            color: white;
            text-align: center;
        }
        #popup {
            display: none;
            width:400px;
            height:500px;
            left:50%;
            top:50%;
            position: absolute;
            padding:50px;
            box-sizing: border-box;
            transform: translate(-50%, -50%);
            border:1px dashed #ddd;
        }
    </style>
</head>
<body>
    <h1>자바스크립트를 이용한 팝업만들기</h1>
    <div id="popup">
        <div class="popup_main">
            <h2>이것은 팝업입니다</h2>
            <p>되려니와, 이상은 스며들어 관현악이며, 장식하는 인생을 뜨고, 이것은 할지라도 운다.
               청춘 그러므로 너의 가치를 때문이다. 그들은 것은 소금이라 하여도 그림자는 위하여
              구하지 황금시대다. 능히 그들은 풍부하게 동력은 풀밭에 위하여, 그것은 듣는다.
              위하여서, 황금시대를 원질이 영락과 영원히 피가 두손을 것이다.
              방황하였으며, 뛰노는 사는가 바이며, 바이며, 끓는다.
              천고에 미인을 길지 불러 무한한 우리 따뜻한 보이는 얼마나 이것이다.
              가는 창공에 풍부하게 현저하게 목숨이 능히 같지 것이다.
              찾아 위하여 얼마나 귀는 끓는 이상을 이상의 피다. </p>
        </div>
    </div>
    <button id="openPop">open</button>
    <script>
        let openBtn = document.querySelector("#openPop");

        openBtn.onclick = function(){
            let popup = document.getElementById("popup");
            popup.style.display="block";
        }
    </script>
</body>
</html>
```

② 모달팝업

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>자바스크립트</title>
    <style>
        .popup_main h2 {
            background:#333;
```

```
            padding:5px 10px;
            font-size:1.5rem;
            color: white;
            text-align: center;
        }
        #popup {
            display: none;
            position:fixed;
            width: 100%;
            height: 100%;
            top:0;
            left:0;
            background:rgba(0,0,0,.5);
            z-index:1000;
        }
        .popup_main {
            width:400px;
            height:500px;
            left:50%;
            top:50%;
            position: absolute;
            padding:50px;
            box-sizing: border-box;
            transform: translate(-50%, -50%);
            border:1px dashed #ddd;
            background:#fff;
        }
    </style>
</head>
<body>
    <h1>자바스크립트를 이용한 팝업만들기</h1>
    <div id="popup">
        <div class="popup_main">
            <h2>이것은 팝업입니다</h2>
            <p>되려니와, 이상은 스며들어 관현악이며, 장식하는 인생을 뜨고, 이것은 할지라도 운다. 청춘 그러므로 너의 가치를 때문이다. 그들은 것은 소금이라 하여도 그림
            <button id="closePop">close popup</button>
        </div>
    </div>
    <button id="openPop">open modal popup</button>
    <script>
        let openBtn = document.querySelector("#openPop");
        let closeBtn = document.querySelector("#closePop");
        openBtn.onclick = function(){
            let popup = document.getElementById("popup");
            popup.style.display = "block";
        }
        closeBtn.onclick = function(){
            let popup = document.getElementById("popup");
            popup.style.display = "none";
        }
    </script>
</body>
</html>
```

③ 슬라이드

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width">
  <title>JS Bin</title>
  <link href="slide.css" rel="stylesheet">
</head>
<body>
  <div id="slide">
      <div class="slide_imgs">
            <img src="//via.placeholder.com/600x200/abc/fff?text=slide1" alt="">
            <img src="//via.placeholder.com/600x200/def/fff?text=slide2" alt="">
            <img src="//via.placeholder.com/600x200/a9c/fff?text=slide3" alt="">
      </div>
      <div id="prev">이전</div>
      <div id="next">다음</div>
      <ul id="pages">
        <li><span>1</span></li>
        <li><span>2</span></li>
        <li><span>3</span></li>
      </ul>
  </div>
  <script src="slide.js"></script>
</body>
</html>
```

```
* {
  margin:0;padding:0;
}
ul { list-style: none; }
span {font-size:0;}
#slide {
  width:600px;
  height:200px;
  overflow:hidden;
  margin:50px auto;
  background:#eee;
  position: relative;
  overflow: hidden;
}
.slide_imgs {
    width: 1800px;
    height: 200px;
    position: absolute;
    left:0; top:0;
    transition: all 0.3s;
}
.slide_imgs > img {
    float:left;
}
#pages {
  position:absolute;
  width:50px;
  height: 10px;
  display:flex;
  justify-content:center;
  gap:10px;
  bottom:10%;
  left:50%;
  transform: translateX(-50%);
  display:none;
}
#pages > li {
  width:10px;
  height:10px;
  background:#333;
  border-radius: 50%;
}
#prev, #next {
    position: absolute;
    top: 50%;
    background: #fff;
    padding:5px 10px;
}
#prev {
    left: 10%;
}
#next {
    right: 10%;
}
```

```
/* variables */
/* 슬라이드 만드는데 필요한 변수 선언
  - 슬라이드 대상
  - 슬라이드 이미지 갯수
  - 슬라이드 이미지 폭
  - 슬라이드 (현재) 인덱스(=유사배열)
  - 다음버튼
  - 이전버튼
*/
let $slide, $slideLength, $slideWidth, $slideCurrent, prevBtn, nextBtn;

// 프로그램 실행, 초기화(tv 리모컨 켜면, 채널정보,볼륨,영상/음성 정보 받고 화면표시 하는것과 같음)
init();

function init(){ // 초기설정
    $slide = document.querySelector(".slide_imgs"); // 슬라이드 이미지 지정
    $slideLength = document.querySelectorAll(".slide_imgs > img").length; // 사진 갯수 알아오기
    $slideWidth = document.querySelector("#slide").clientWidth; // 사진 가로 사이즈 알아오기
    $slideCurrent = 0; // 현재 사진은 0번 슬라이드(배열의 개념)
    prevBtn = document.querySelector("#prev"); // 이전 버튼 지정
    nextBtn = document.querySelector("#next"); // 다음 버튼 지정
}

function slideNext(){ // 슬라이드 다음 넘기는 기능
    let nextIndex = $slideCurrent + 1; // 다음 = 현재 + 1;
    if (nextIndex >= $slideLength) nextIndex = 0; // 계속 증가하면, 슬라이드 없는 값이 나오는 것을 방지
```

```
    let slidePos = -$slideWidth * nextIndex; // 사진은 600px, 한장씩 옮겨지는 간격을 생각해보기
    $slide.style.left = slidePos+"px"; // css 조작시 px 단위 추가
    console.log(slidePos+"px"); // 콘솔 탭에서 확인
    $slideCurrent = nextIndex; // 지역변수 vs 전역변수 - 함수 밖에 있는 값을 업데이트
}
function slidePrev(){ // 슬라이드 이전 기능
    let nextIndex = $slideCurrent - 1;    // 다음 사진 = 현재 - 1;
    if (nextIndex < 0) nextIndex = $slideLength - 1; // 계속 감소하면, 마이너스 나올 수 있음..
    let slidePos = -$slideWidth * nextIndex;
    $slide.style.left = slidePos+"px";
    console.log(slidePos+"px");
    $slideCurrent = nextIndex;
}
nextBtn.addEventListener("click", slideNext); // 다음 버튼에 다음슬라이드 기능 부착(붙이기)
prevBtn.addEventListener("click", slidePrev); // 이전 버튼에 이전슬라이드 기능 부착(붙이기)
```