

# HICC 프론트엔드 세미나\_6주차

## JavaScript(2)

By. 오성빈, 최서연

# Today's study

Event  
이벤트 핸들러  
제어할 태그 선택  
event 활용  
모듈  
실습

# Event

- 웹에서 일어나는 사건
- 마우스이동, 클릭, 엔터, 입력, 스크롤, 포커스등

1. 사용자가 Page 에서 어떠한 상호작용을 함



2. 이벤트(Event) 발생



4. 페이지에서 어떠한 변화가 생김



3. 자바스크립트 코드 실행

```
function myCode() {  
  // do something  
  return;  
}
```

## Event 종류

### 1) 마우스 이벤트

이벤트	설명
click	요소에 마우스를 클릭했을 때 이벤트가 발생
dblclick	요소에 마우스를 더블클릭했을 때 이벤트가 발생
mouseover	요소에 마우스를 오버했을 때 이벤트가 발생
mouseout	요소에 마우스를 아웃했을 때 이벤트가 발생
mousedown	요소에 마우스를 눌렀을 때 이벤트가 발생
mouseup	요소에 마우스를 떼었을 때 이벤트가 발생
mousemove	요소에 마우스를 움직였을 때 이벤트가 발생
contextmenu	context menu(마우스 오른쪽 버튼을 눌렀을 때 나오는 메뉴)가 나오기 전에 이벤트 발생

### 2) 키 이벤트

이벤트	설명
keydown	키를 눌렀을 때 이벤트가 발생
keyup	키를 떼었을 때 이벤트가 발생
keypress	키를 누른 상태에서 이벤트가 발생

### 3) 폼 이벤트

이벤트	설명
focus	요소에 포커스가 이동되었을 때 이벤트 발생
blur	요소에 포커스가 벗어났을 때 이벤트 발생
change	요소에 값이 변경 되었을 때 이벤트 발생
submit	submit 버튼을 눌렀을 때 이벤트 발생
reset	reset 버튼을 눌렀을 때 이벤트 발생
select	input이나 textarea 요소 안의 텍스트를 드래그하여 선택했을 때 이벤트 발생

| Event

# 이벤트 핸들러

- 사용자가 실제 이벤트를 발생시켰을 때  
그 이벤트에 대응하여  
처리하는 것
- 앞에 'on'을 붙여 주고  
이벤트에 대한 동작(함수)을  
처리

```
<button onclick=  
"alert('button pressed')">button</button>
```

| Event

# 이벤트 핸들러

- HTML 속성을 이용한 이벤트 핸들러 할당

```
<button onclick=  
"alert('button pressed')">button</button>
```

# 이벤트 핸들러

- JavaScript를 이용한 이벤트 핸들러 할당
  - 권장되는 방식
  - 유지보수 용이
- 여러 이벤트 핸들러 추가 가능

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

```
<script>
```

```
    var targetBtn =
```

```
    document.getElementById('btn');
```

```
    targetBtn.addEventListener('click',
```

```
    function(event){
```

```
        alert('button pressed');
```

```
    });
```

```
</script>
```

| Event

# Arrow 함수

## 화살표 함수(Arrow function)

화살표 함수를 사용할 경우,  
익명함수를 사용할 때 function 키워드를 사용하지 않아도 됨.

```
export default () => {  
  console.log('Hello');  
}
```

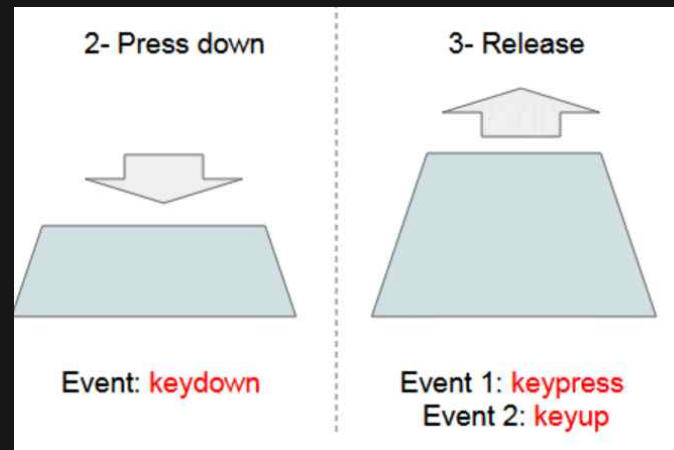
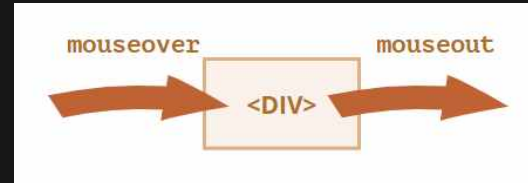
```
<script>  
  var targetBtn = document.getElementById('btn');  
  targetBtn.addEventListener('click', () => {  
    alert('button pressed');  
  });  
</script>
```



## Event

# 주 이벤트

- click / dblclick
- mouseover/mouseout
- keydown / keypress /keyup (키보드)
  - submit
  - change
- load (페이지 로딩)
  - scroll



# 제어할 태그 선택

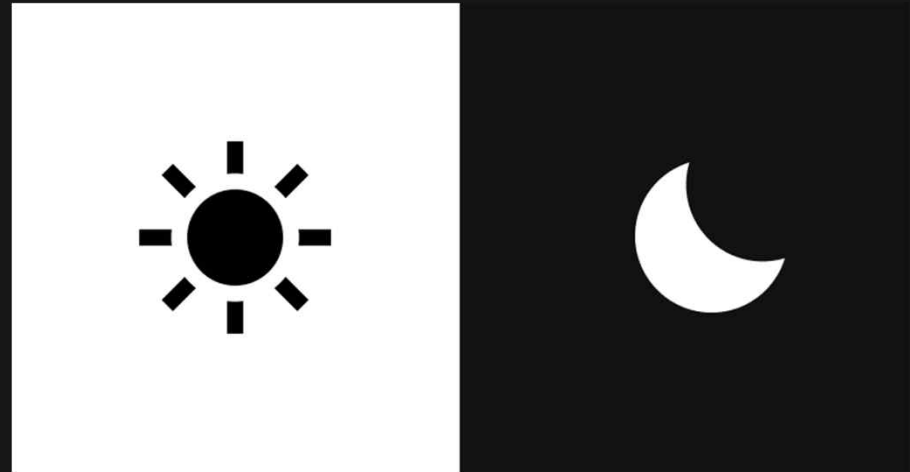
- querySelector가 문서내 첫번째 해당 태그를 찾아서 선택해줌
  - 클래스를 선택하고 싶다면  
querySelector('.class명')
  - id를 선택하고 싶다면  
querySelector('#id명')
- querySelectorAll로 여러 클래스 선택 가능

```
<script>
    var targetBtn = document.getElementById('btn');
    targetBtn.addEventListener('click', () => {
        document.querySelector('body').style.backgroundColor
            = 'black';
    });
</script>
```

| Event

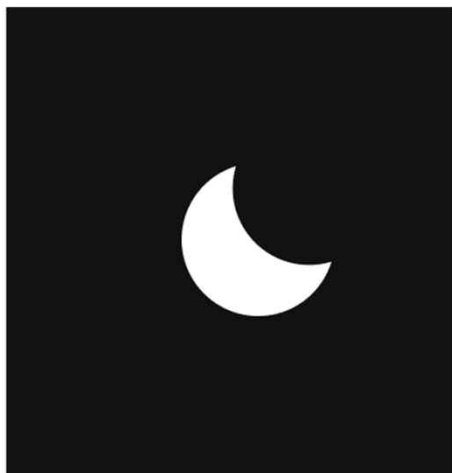
# event 활용

- (1) light 모드일 때는 버튼의 값을 dark,  
dark 모드일 때는 light로 표시하기
- (2) light 모드일 때는 글자가 검은색,  
dark 모드일 때는 글자를 흰색으로 표시



Event

# event 활용

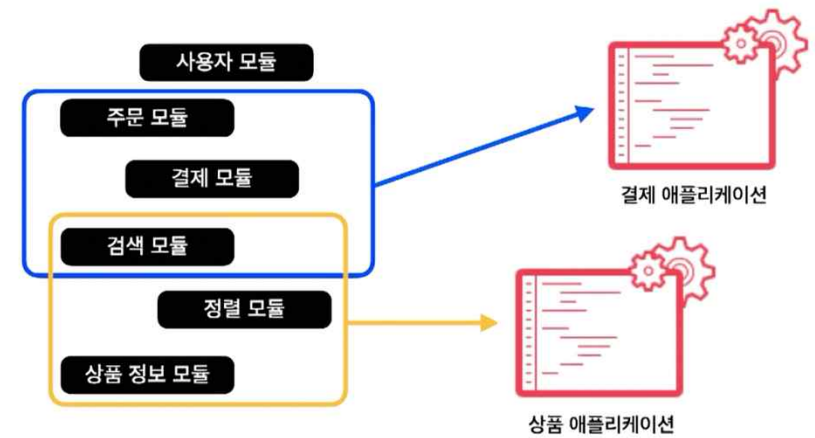


```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5    </head>
6    <body>
7      <input type="button" value="dark" id="btn">
8      <p>hello</p>
9      <script>
10         var targetbtn = document.getElementById('btn')
11         targetbtn.addEventListener('click',()=>{
12             if(targetbtn.value=='dark'){
13                 targetbtn.value='light';
14                 document.querySelector('body').style.backgroundColor='black';
15                 document.querySelector('body').style.color = 'white';
16             }
17             else{
18                 targetbtn.value='dark';
19                 document.querySelector('body').style.backgroundColor='white';
20                 document.querySelector('body').style.color = 'black';
21             }
22         })
23     </script>
24 </body>
25 </html>
```

# 모듈

- 분리된 파일
- import 와 export로 다른 모듈에서 사용가능
- 확장성 및 재사용성 증가

## 1-1. Module system의 필요성



# 모듈

- 선언시 export 를 붙이면  
내보내기 가능(함수, 클래스, 변수)
- 사용하려는 파일에서 import해서  
가져올 수 있음
- 로컬 서버가 아닌 가상 서버에서  
실행해야 제대로 동작

JS add.js

```
1 export function add(a, b) {  
2   return a + b;  
3 }  
4  
5
```



```
<> console.html > {} "console.html"  
1 <!DOCTYPE html>  
2 <html>  
3 <head>  
4   <meta charset="utf-8">  
5 </head>  
6 <body>  
7   <script type="module">  
8     // 웹 브라우저에서 javascript 모듈 사용  
9     import { add } from './add.js';  
10  
11     alert(add(5, 3));  
12   </script>  
13 </body>  
14 </html>  
15
```

# 실습

- 버튼을 누르면 카운트가 올라가는 것을 이벤트를 통해서 구현해보자

- count를 웹에서 바꾸기 위해서는

```
counter.textContent = 'Click Count: ' + clickCount;
```

코드가 필요함

(counter는 'Click Count:' 단락에 대한 id, textContent는 해당 요소가 표시하는 텍스트 내용을 나타냄, clickCount 는 카운트를 세기 위한 변수)

button

Click Count: 0

button

Click Count: 1

button

Click Count: 2

---

# 감사합니다

reference:

- <https://ko.javascript.info/import-export>
- <https://jenny-daru.tistory.com/17>