

# React

## Event Handling

# Event Handling

# React의 Event

- HTML Event

```
<button onclick="activeEvent();">  
  Event Button  
</button>
```

- React Event

```
<button onClick={activeEvent}>  
  Event Button  
</button>
```

# React의 Event 주의점

1. React의 이벤트는 소문자가 아닌 **카멜 케이스(camelCase)** 이용
2. JSX를 사용해 이벤트 핸들러 전달

```
<button onclick={activeEvent()}>  
  Event Button  
</button>
```



```
<button onClick={activeEvent}>  
  Event Button  
</button>
```



## 3. 기본 DOM 요소에만 이벤트 설정 가능

직접 만든 컴포넌트에 이벤트 자체 설정 불가능!! (대신, props를 이용해 설정할 수 있음)

Ex) `<MyButton onClick={activeEvent} />` // 이름이 onClick인 props를 MyButton 컴포넌트에게 전달

# React 합성 이벤트

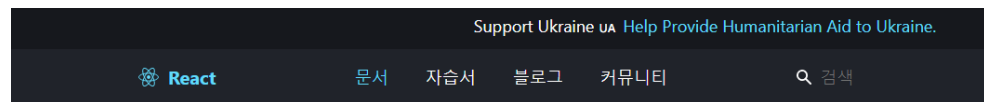
- **Synthetic Event ( 합성 이벤트 )** React에서 사용하는 이벤트는 브라우저의 기본 이벤트가 아니다!
- 모든 브라우저에서 이벤트를 동일하게 처리하기 위한 Wrapper(래퍼) 객체

※ Wrapping이란? 기본 기능을 감싸는 새로운 기능

★ 리액트가 DOM이 아닌 **VirtualDOM**을 사용하는 것처럼  
웹 브라우저의 `nativeEvent`가 아닌  
`nativeEvent`를 감싼 **SyntheticEvent**를 사용

더 많은 이벤트 종류는 공식 문서를 참고하자!

<https://ko.reactjs.org/docs/events.html>



## 합성 이벤트(SyntheticEvent)

이 문서는 React의 이벤트 시스템 일부를 구성하는 `SyntheticEvent` 래퍼를 설명합니다. 더 많은 정보는 [이벤트 처리하기](#) 문서를 보세요.

### 개요

이벤트 핸들러는 모든 브라우저에서 이벤트를 동일하게 처리하기 위한 이벤트 래퍼 `SyntheticEvent` 객체를 전달받습니다. `stopPropagation()` 와 `preventDefault()`를 포함해서 인터페이스는 브라우저의 고유 이벤트와 같지만 모든 브라우저에서 동일하게 동작합니다.

이벤트 객체는 `Event` 인터페이스를 구현하며, `Event` 인터페이스를 상속받은 `Event` 객체입니다.

# 클래스 컴포넌트에서의 이벤트

- 클래스형 컴포넌트에서는 **this**를 사용하여 함수를 찾아갈 수 있다.
  - (state와 props와 동일 )

```
<button onClick={this.printConsole}>printConsole2</button>  
<button onClick={this.handleButton}>handleButton</button>
```

# 함수에서 this 사용하기

in 클래스 컴포넌트

## 1. bind 사용하기 (이전 버전)

- 공식 문서 참고하기!

<https://react.dev/reference/react/Component>

```
constructor(props) {  
  super(props);  
  
  this.eventExample = this.eventExample.bind(this);  
}  
eventExample() {  
  console.log( this )  
}
```

## 2. 화살표 함수 이용하기

```
eventExample2 = () => {  
  console.log( this );  
}
```

# 함수형 컴포넌트에서의 이벤트

- 함수에 인자가 없는 경우

```
<button onClick={increase}>더하기</button>
```

- 함수에 인자가 있는 경우

```
<button onClick={() => alertMsg('helloooo')}>alert 출력</button>  
<button onClick={(e) => consoleMsg(e, 'helloooo')}>console 출력</button>
```