

## 2. 이벤트

event란 여러분이 프로그래밍하고 있는 시스템에서 일어나는 사건(action) 혹은 발생(occurrence) 이는 여러분이 원한다면 그것들에 어떠한 방식으로 응답할 수 있도록 시스템이 말해주는 것이다. 예를 들면, 사용자가 웹 페이지에 있는 버튼을 고른다면 여러분은 그 사건에 이포메이션 박스를 표시함으로써 응답하기를 원할지도 모른다.

```
<button className='toast__button' onClick={() => console.log('event occurs')}>
  Dismiss
</button>
```

### 이벤트 핸들러

: 이벤트가 발생되면 실행되는 코드 블록(보통 프로그래머가 만드는 자바스크립트 함수)

: 코드 블록이 이벤트에 응답해서 실행되기 위해 정의되었을 때, 이를 이벤트 핸들러 등록(register)했다고 함

```
const Toast = ({ message }) => {
  const handleClickHandler = () => {
    console.log('event occurs');
  };
  return (
    <div className={`toast toast-${message.title}`}>
      <ToastMessage message={message}/>
      <button className="toast__button">
        Dismiss
      </button>
    </div>
  )
};
```

### HTMLButtonElement

: 버튼에는 어떤 이벤트 핸들러가 있을까?

★ 블로그 보다는 공식문서를 참고하는 습관을 가지기

The HTMLButtonElement interface provides properties and methods(beyond the regular HTMLElement interface it also has available to it by inheritance) for manipulating `<button>` elements.

EventTarget > Node > Element > HTMLElement > HTMLButtonElement

## 이벤트 리스너

: 이벤트(사용자의 행동)를 듣는(지켜보는) 메서드

: 이벤트 리스너에는 DOM 요소가 필요

: JavaScript : `.addEventListener()` 메서드

: React : 컴포넌트에 직접 넣어줌(e.g. `onClick={}`)

---

```
const Toast = ({ message }) => {  
  const handleClickHandler = (title) => {  
    console.log(title)  
  };  
};
```

- 순수 함수

어떠한 함수가 내부 로직이 매개변수를 통해서만 변화가 되도록 만들어진 함수