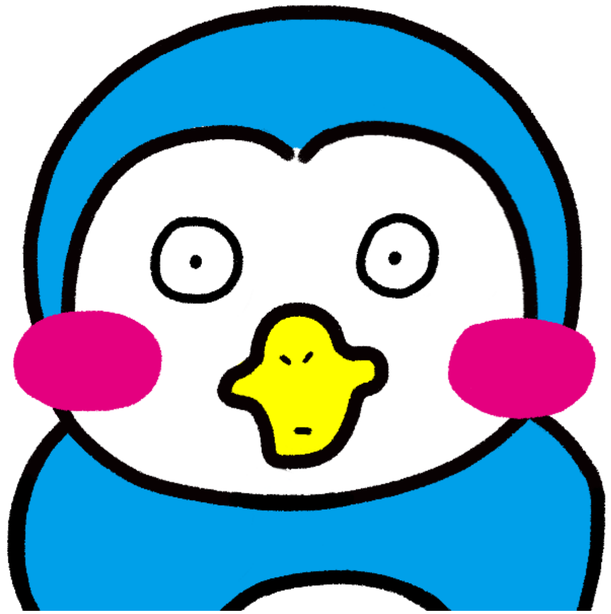
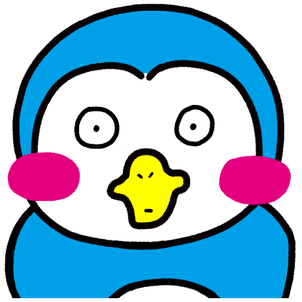
문이 열리네요
그대가 들어오죠



이벤트를 소개합니다

이벤트(event)가 뭔데요

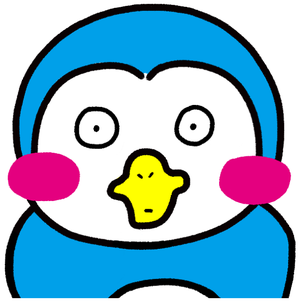


‘사용 중이거나 프로그래밍 중인 시스템 내에서 일어나는 사건’을 뜻한다.
이 강의에서 우리는 웹을 다루고 있고, (당연히)웹에서도 이벤트가 발생한다.

웹에서 발생하는 이벤트의 예

- 웹페이지 사용자가 버튼을 클릭했다, 클릭 이벤트!
- 웹페이지 사용자가 키보드를 눌렀다, 키다운 이벤트!
- 웹페이지 사용자가 입력 폼의 내용을 제출했다, 제출 이벤트!
- 외 다수

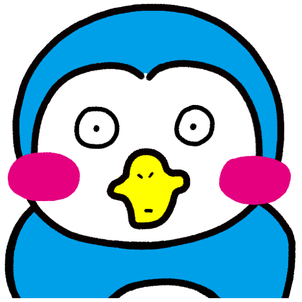
발생하면 어떡하죠?



각각의 이벤트들은 이벤트 핸들러(handler)를 가질 수 있다. 이벤트 핸들러란 이벤트가 발생되면 실행될 코드 블록을 뜻하며, 주로 함수가 이 역할을 담당한다. 이벤트 핸들러 역할을 수행할 함수를 정의하는 것을 이벤트 핸들러 등록이라 한다.

=> event handler register: 이벤트가 발생하면, 이 함수를 호출해라!

예를 들어봅시다



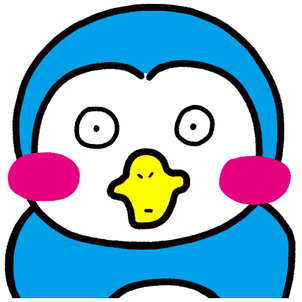
웹 사용자가 버튼(button) 요소를 클릭 했을 때, 경고 다이얼로그 박스를 띄워 환영의 메시지를 보여주고 싶다면?

```
const handleClick = function(){
  window.alert("환영합니다^^")
}

const button = document.querySelector("button")

button.onclick = handleClick // 여기가 포인트!
```

구문 기본 형태

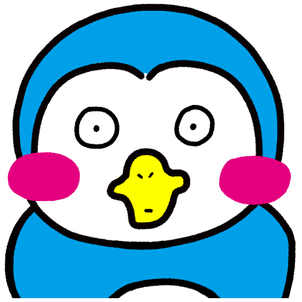


이벤트가 발생할 수 있는(혹은 발생할 예정인) 타겟을 선택하고, 이벤트 핸들러 속성에 이벤트 핸들러를 대입한다.

```
타겟.on이벤트명 = 이벤트핸들러함수
```

```
button.onclick = handleClick
```

이거 헷갈리면 안 돼요

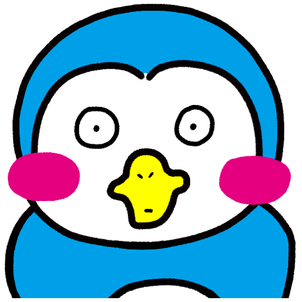


이벤트 핸들러를 등록하기 위해 이벤트 속성에 함수를 대입하는 것과 함수 호출 문을 대입하는 것은 엄연히(!) 다르다.

```
// handleClick 함수를 대입한다 (이벤트 핸들러 등록)
button.onclick = handleClick

// handleClick 호출 후 반환값을 대입한다 (굳이?)
button.onclick = handleClick()
```

내용 정리



- 이벤트란 '시스템 내에서 일어나는 사건'을 뜻한다.

  (마우스 클릭, 키보드 입력, 마우스 휠 스크롤 등 다양해요)
- 각각의 이벤트들은 이벤트 발생 시 대응으로 이벤트 핸들러를 가질 수 있다.
- 이벤트 핸들러 정의하는 작업을 이벤트 핸들러 등록이라 한다.
- 이벤트 속성에 함수를 대입하는 것과 함수 호출문을 대입하는 것은 다르다.