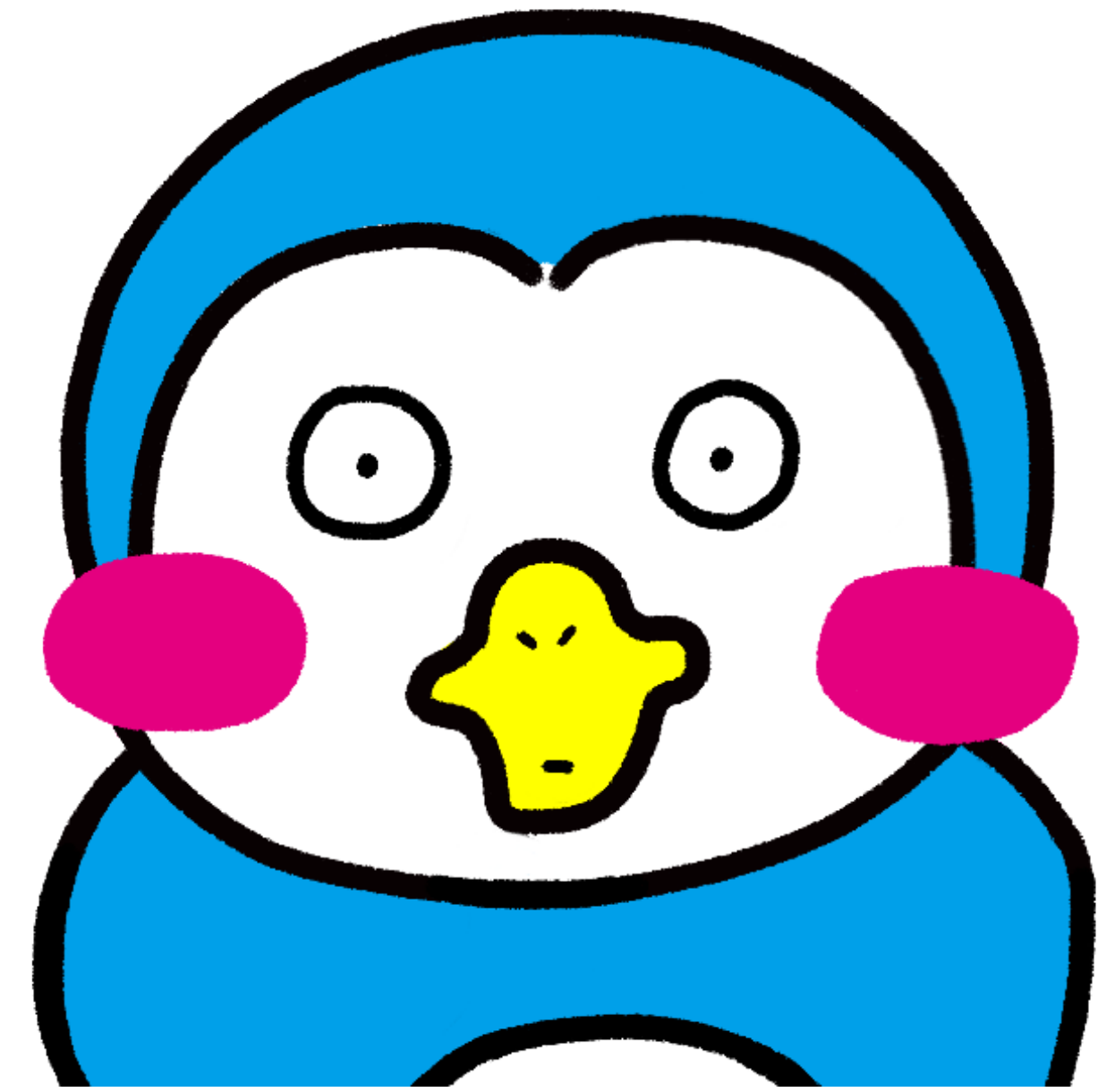


# 이벤트 핸들링 제대로(?) 해보자



---

addEventListener & 이벤트 객체

# addEventListener



onclick, onkeydown 과 같은 이벤트 속성을 통해 이벤트 핸들러를 등록하는 것보다 현대적인 방법은, addEventListener 메소드를 활용하는 것이다.

클릭 이벤트 핸들러를 등록하는 경우의 예

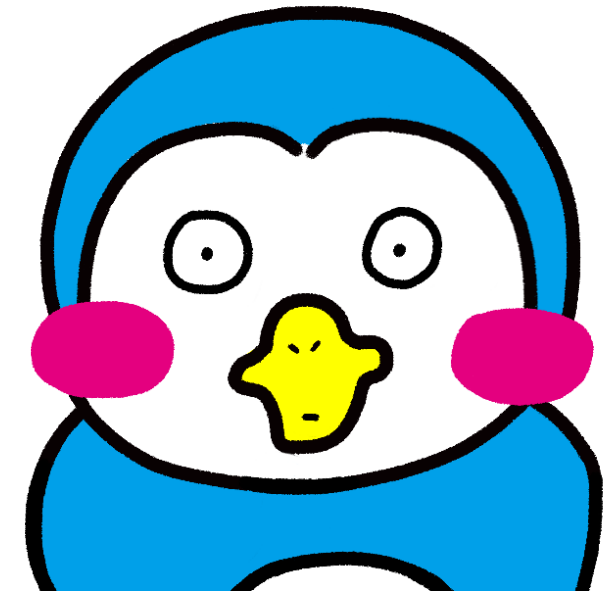
```
// 이전 영상에서 학습한 방식
```

```
target.onclick = function(){} 
```

```
// addEventListener의 방식
```

```
target.addEventListener('click', function(){} )
```

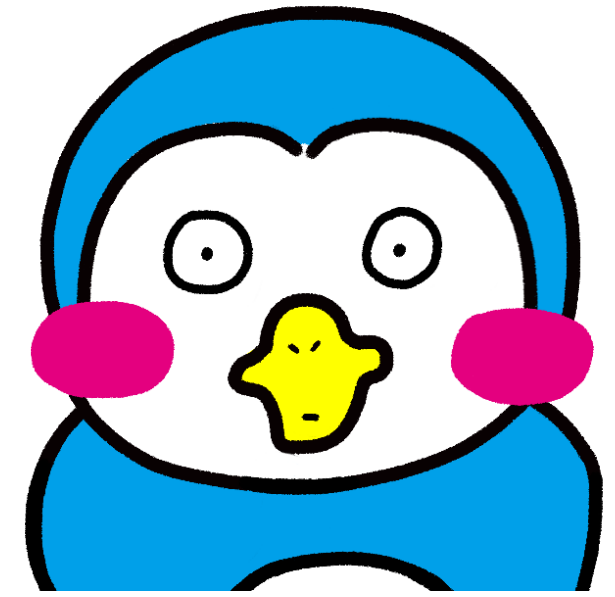
# 똑같은 뭐가 현대적이야



addEventListener 함수를 사용하는 것은 이벤트 핸들러 속성을 사용하는 것에 비해 몇 가지 이점을 제공한다.

- 이전에 추가한 이벤트 핸들러를 제거할 수 있는 대응 메소드가 존재한다.
- 같은 리스너(타겟)에 대해 다수의 핸들러를 등록할 수 있다.
- 추가적인 옵션 사항들이 제공된다.

# 이벤트 객체

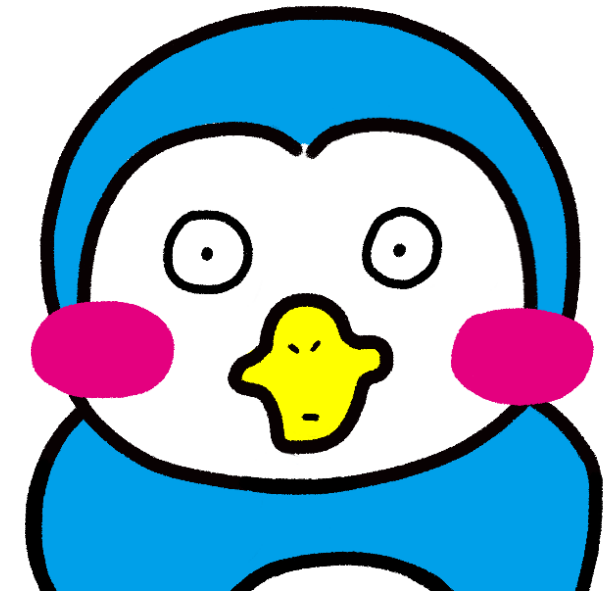


이벤트 객체는 추가적인 기능과 정보를 제공하기 위해 이벤트 핸들러에 자동으로 전달되는 데이터이다. 이를 활용하기 위해서는 이벤트 핸들러 함수에 매개변수를 추가하여 이벤트 발생 시마다 전달받을 수 있도록 해야 한다.

```
// click 이벤트가 발생하면 함수를 호출하겠다!  
target.addEventListener('click', function(){})  
  
// click 이벤트가 발생하면 함수를 호출하겠다!  
// + 이때 자동으로 전달되는 이벤트 객체를 매개변수 event에 대입하겠다!  
target.addEventListener('click', function(event){})
```

# 내용 정리

---



- `addEventListener` 메소드를 활용해 이벤트 핸들러 등록을 할 수 있다.
- 이는 이벤트 핸들러 속성을 사용하는 것보다 현대적인(좋은) 방법이다.
- 이벤트 객체는 추가적인 기능과 정보를 제공하기 위해 이벤트 핸들러에 자동으로 전달되는 데이터이다.
- 이벤트 발생 시에 이벤트 핸들러가 호출될 때 이벤트 객체가 전달되는데, 이때 이벤트 핸들러 함수의 매개변수를 통해 이벤트 객체를 받을 수 있다.