

유노코딩과 함께하는 자바스크립트 기초 강의

이벤트 &
이벤트 핸들링





이벤트(event)가 뭔데

웹 프로그래밍에서 이벤트란, DOM에서 발생하는 다양한 액션 또는 상호작용 동작을 나타내는 프로그래밍 인터페이스이다.

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



이벤트가 발생했다! 어떻게 할까?

각각의 이벤트에 대한 이벤트 핸들러(handler)를 정의할 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.

다음 예는 웹 사용자가 버튼(button) 요소를 클릭 했을 때, 경고 다이얼로그 박스를 띄워 환영의 메시지를 보여주도록 이벤트를 처리한 예다.

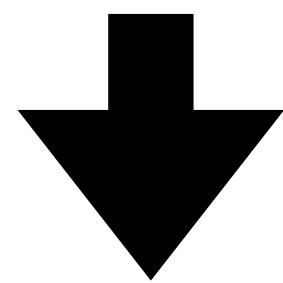
```
const handleClick = function(){  
  window.alert("환영합니다^^")  
}  
  
const button = document.querySelector("button")  
  
button.onclick = handleClick // 여기가 포인트!
```



구문 기본 형태

이벤트를 처리할 타겟을 선택하고, 이벤트 핸들러 속성에 이벤트 핸들러를 대입한다.
주의! 이벤트 핸들러를 등록하기 위해 이벤트 속성에 함수를 대입하는 것과 함수 호출
문을 대입하는 것은 엄연히(!) 다르다.

```
타겟.on이벤트명 = 이벤트핸들러함수
```



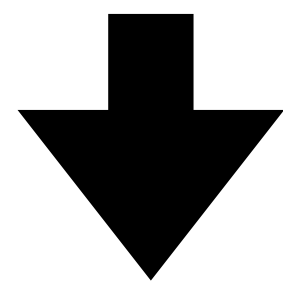
```
button.onclick = handleClick
```



구문 기본 형태

이벤트를 처리할 타겟을 선택하고, 이벤트 핸들러 속성에 이벤트 핸들러를 대입한다.
주의! 이벤트 핸들러를 등록하기 위해 이벤트 속성에 함수를 대입하는 것과 함수 호출
문을 대입하는 것은 엄연히(!) 다르다.

```
타겟.on이벤트명 = 이벤트핸들러함수
```



```
button.onclick = handleClick
```

말씀 중에 죄송합니다.
뒤에 괄호 붙이는 거 아닙니다!



addEventListener() 사용하자

onclick, onkeydown 과 같은 이벤트 속성을 통해 이벤트 핸들러를 등록하는 것보다 편하고 유용한 이벤트 처리 방법은, addEventListener 메소드를 활용하는 것이다. 이 메소드는 다음과 같은 장점들을 제공한다.

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

addEventListener(이벤트명, 이벤트핸들러)



이벤트 객체까지!

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){})
```



수고하셨습니다 :))