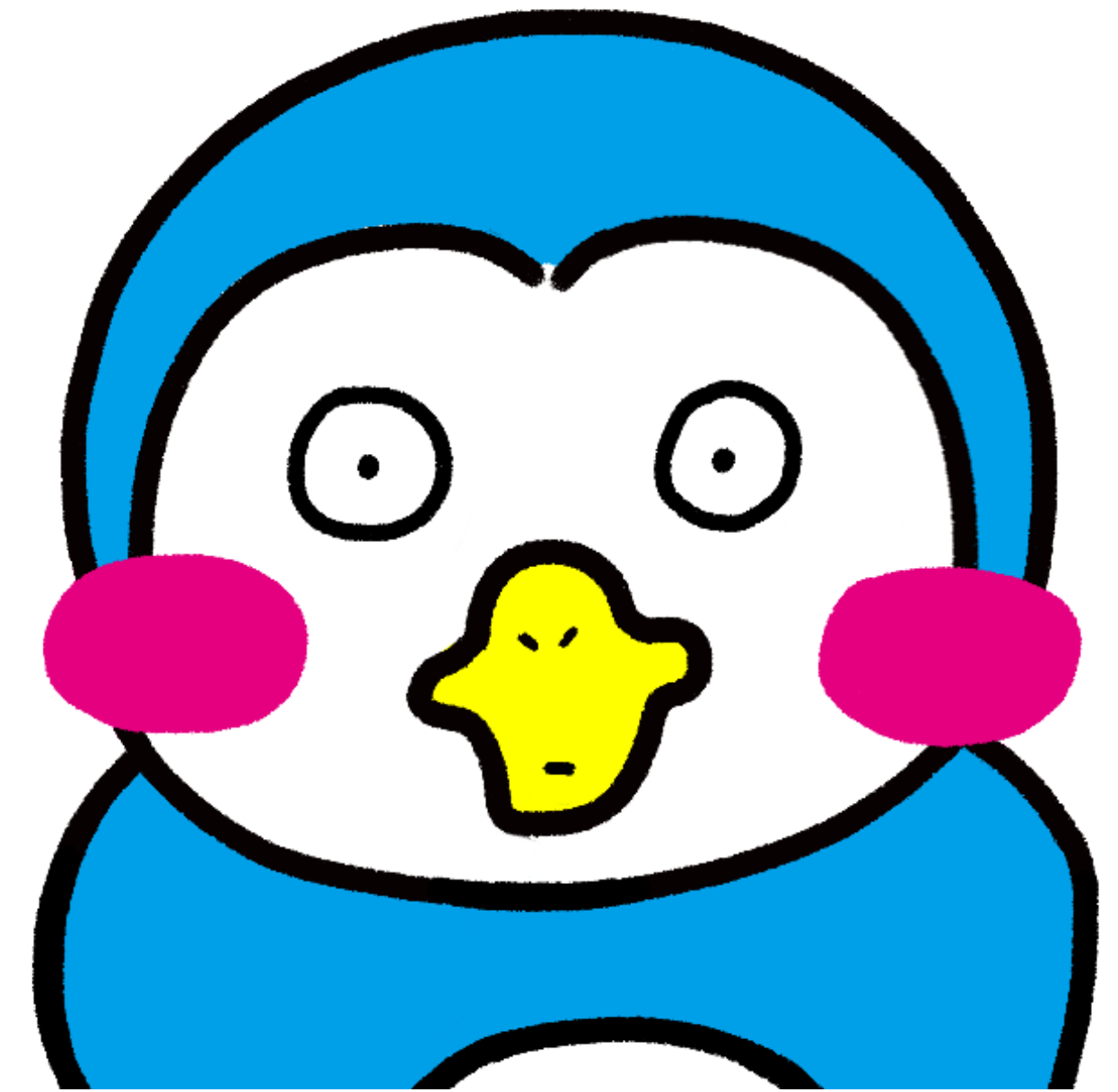


입력된 값을
읽어봅시다



value 속성 외

입력 요소 값 읽기

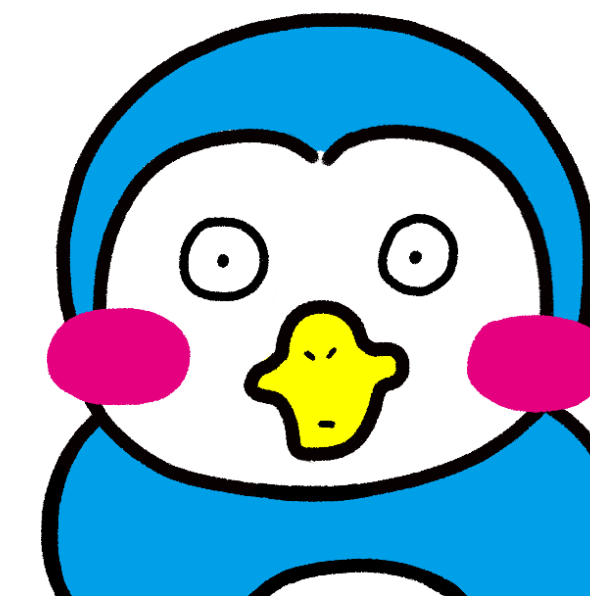


〈input〉, 〈select〉 처럼 사용자로 부터 입력을 받는데 사용 되는 요소들이 있다.
여기에서 사용자가 입력한 값을 읽어들이는 때는 요소의 value 속성에 접근하자.

차이를 기억하자

- 요소의 텍스트에 접근하고 싶다: `textContent` 또는 `innerText`
- 사용자가 요소에 입력한 값에 접근하고 싶다: `value`

접근이라 함은...



입력 요소의 value에 접근하여 할 수 있는 일은 크게 두 가지, 읽기와 쓰기이다.

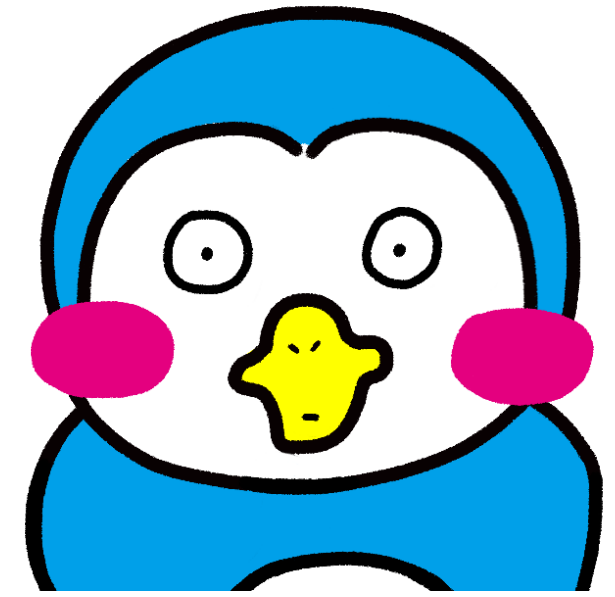
```
// 대상 요소의 사용자 입력값을 읽어 콘솔에 출력하자!
```

```
console.log(target.value)
```

```
// 대상 요소의 사용자 입력값을 "변경값"으로 바꾸자!
```

```
target.value = "변경값"
```

입력 폼을 다루어보자

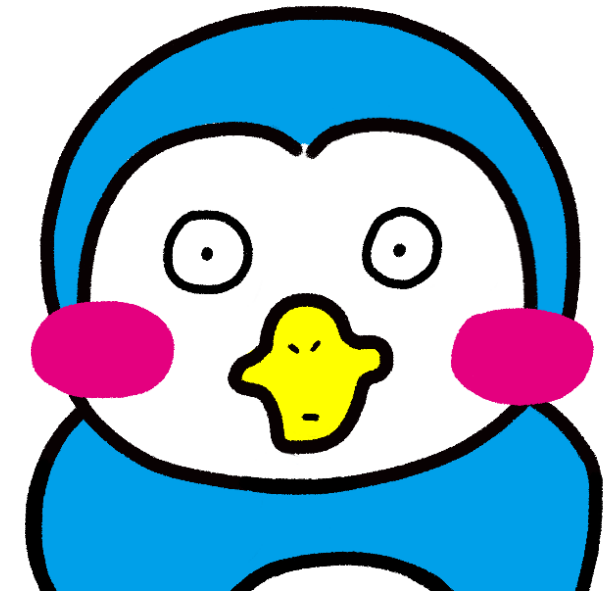


여러 입력 요소를 포함할 수 있는 폼 요소로부터 여러 입력 값을 읽을 수 있다.

```
<form>
  <input name="nickname" placeholder="닉네임" />
  <input name="character" placeholder="특징" />
  <input type="submit" value="입력 완료" />
</form>
```

=> form에서 submit 이벤트 발생하면 입력 값들을 읽자! 앗, 그런데 문제가...

form은 원래 action이 있잖아

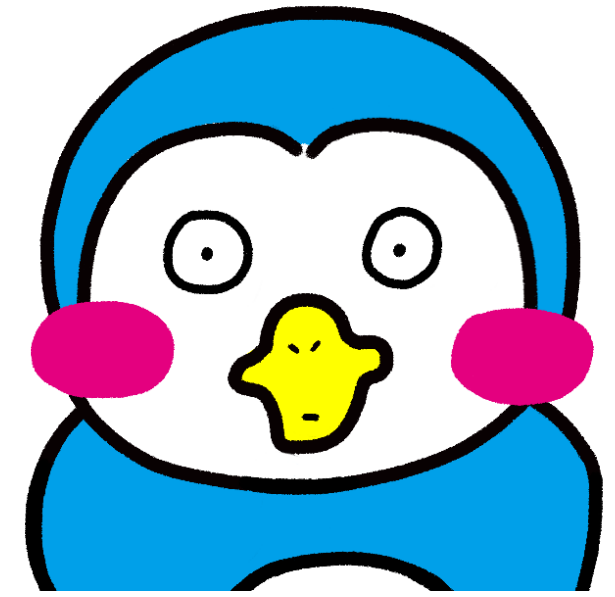


form은 action에 지정한 url로 입력값을 제출한다.

```
<form action="없는데요">  
  <input name="nickname" placeholder="닉네임" />  
  <input name="character" placeholder="특징" />  
  <input type="submit" value="입력 완료" />  
</form>
```

=> action이 없을 땐? 이벤트 객체를 사용해 봅시다.

내용 정리



- 사용자가 입력한 값을 읽어들이는 때는 요소의 value 속성에 접근하면 된다.
- value에 접근함으로써 입력 값 읽기 또는 쓰기를 수행할 수 있다.
- form에 포함된 입력 요소의 name을 통해 각 입력 요소에 접근할 수 있다.
- form에서 이벤트가 제출되는 submit 이벤트가 발생하면 action 속성의 url로 리다이렉트되지만, 이벤트 객체를 통해 기본 기능을 차단할 수 있다.