

입력 요소 제어하기

#01. 입력 요소 값 가져오기

1. 값 설정하기

```
$("#선택터").val("내용");
```

2. 값 읽어오기

text, select(드롭다운)

```
const input = $("#선택터").val();
```

드롭다운요소의 경우 사용하지 않는 `<option>`태그라 하더라도 value속성값을 부여해야 한다.

```
<select name='foo'>
  <option value=''>--- 선택하세요 ---</option>
  <option value='bar'>item 1</option>
</select>
```

선택된 항목의 인덱스 확인하기

```
$(e.currentTarget).prop("selectedIndex")
```

라디오 버튼의 경우

- 선택터가 여러개를 의미하는 경우이므로 그 중에서 선택된 항목만 지정하는 선택터를 사용해야 한다.
- 라디오는 하나만 선택 가능하므로 가져온 결과 자체는 단일 값이다.

```
const input = $("#선택터:checked").val();
```

체크박스의 경우

- `checked`가상클래스를 통해 가져온 대상이 컬렉션 상태가 되므로 이를 반복문으로 처리해야 한다.
- 컬렉션의 요소는 선택된 체크박스들에 대한 HTML 태그가 된다.
- 그러므로 각 HTML 태그를 다시 jQuery 객체로 변환하여 사용해야 한다.

```
const input = $("선택터:checked");

for (let i=0; i<input.length; i++) {
  // 배열에 포함된 원소(HTML태그)를 jQuery객체로 변환
  const tmp = $(input[i]);
  ...
}
```

배열의 탐색 함수를 사용할 경우 컬렉션을 배열로 변환한다.

```
const input = $("선택터:checked");

Array.from(input).forEach((v, i) => {
  // 배열에 포함된 원소(HTML태그)를 jQuery객체로 변환
  const tmp = $(v);
  ...
});
```

#02. 입력요소 관련 이벤트

1. submit 버튼의 이벤트

submit 버튼이 클릭되면 <form>태그 내의 모든 항목에 입력된 내용을 백엔드 페이지로 전송하게 된다. 즉, submit 이벤트는 버튼 자체의 이벤트가 아니라 <form>태그에 대한 이벤트라는 것이다.

<form>태그는 기본적으로 모든 입력 내용을 백엔드 페이지에 전송하려고 한다. 만약 action 속성이 없다면 HTML 스스로를 전송 대상으로 삼기 때문에 입력값을 현재 HTML로 전송하기 위해 페이지가 새로고침 된다.

그러므로 Javascript가 정상적으로 동작하기 위해서는 submit 이벤트 첫 행에서 e.preventDefault()를 호출하여 <form>태그가 원래 수행해야 하는 동작(입력값 전송)을 수행하지 못하도록 차단해야 한다.

```
<form id="myform" action="백엔드페이지URL" method="GET 혹은 POST">
  ...
  <button type="submit">전송하기</button>
</form>

<script src="jquery 경로">
<script>
  $("#myform").on("submit", (e) => {
    e.preventDefault();

    //... 전송 버튼을 눌렀을 때의 처리 ...

    // 백엔드 페이지에게 데이터를 전송해야 할 경우 off() 함수로 submit이벤트를 끄고
    강제 전송 해야 한다.
    //$(e.currentTarget).off('submit').submit();
```

```
});  
</script>
```

2. 포커스 관련 이벤트

포커스가 들어온 경우

```
$("#선택터").on("focus", (e) => {  
    ...  
});
```

포커스가 빠져나간 경우

```
$("#선택터").on("blur", (e) => {  
    ...  
});
```

포커스 강제 지정

포커스 처리는 그 자체가 메서드 역할도 수행한다.

```
$("#선택터").focus();
```

3. 입력 상태가 변경된 경우

change 이벤트는 주로 라디오, 체크박스, 드롭다운에 대하여 활용도가 높다.

```
$("#선택터").on("change", (e) => {  
    ...  
});
```

텍스트를 입력하는 요소일 경우

내용이 변경되고 포커스가 빠져 나올 때 이벤트가 발생한다.

체크박스나 라디오 버튼, 드롭다운의 경우

선택 상태가 변경된 즉시 발생한다.

ex) 체크박스의 checked 속성이 적용되어 있는지 여부 확인

```
<input type="checkbox" class='foo' disabled />
```

```
// 속성이 적용되어 있으므로 true  
const ok = $(".foo").prop('disabled');
```