

## 02-1 이벤트 등록 메서드

# 이벤트 등록 메서드란?

기본형

`<button id="btn">버튼</button>`

이벤트 적용 대상

① 이벤트 대상

② 이벤트 종류

+ 이벤트 등록 메서드  
(click)

```
$("#btn").click(function(){  
    자바스크립트 코드;  
});
```

③ 이벤트 핸들러

이벤트 처리

# 이벤트 등록 방식 알아보기 — 단독 이벤트 등록 메서드

## 기본형

```
$("#이벤트 대상 선택").이벤트 등록 메서드(function( ) {  
    자바스크립트 코드;  
});
```

```
<button id="btn1">버튼</button>  
$(".btn1").click(function( ) {  
    alert("welcome");  
});
```

# 이벤트 등록 방식 알아보기 — 그룹 이벤트 등록 메서드

## 기본형

### ① on() 메서드 등록 방식 1

```
$("#이벤트 대상 선택").on("이벤트 종류1 이벤트 종류2 ... 이벤트 종류n",  
function() {  
    자바스크립트 코드;  
});
```

### ② on() 메서드 등록 방식 2

```
$("#이벤트 대상 선택").on({  
    "이벤트 종류1 이벤트 종류2 ... 이벤트 종류n" : function() {  
        자바스크립트 코드;  
    }  
});
```

### ③ on() 메서드 등록 방식 3

```
$("#이벤트 대상 선택").on({  
    "이벤트 종류1" : function() { 자바스크립트 코드;1 },  
    "이벤트 종류2" : function() { 자바스크립트 코드;2 },  
    ...  
    "이벤트 종류n" : function() { 자바스크립트 코드;n }  
});
```

# 마우스 이벤트 — click( ) 이벤트 메서드

## 기본형

### ❶ click 이벤트 등록

```
$("#이벤트 대상 선택").click(function( ) { 자바스크립트 코드; });
```

```
$("#이벤트 대상 선택").on("click", function( ) { 자바스크립트 코드; });
```

### ❷ click 이벤트 강제 발생

```
$("#이벤트 대상 선택").click( );
```

# 이벤트 등록 방식 알아보기 — 강제 이벤트 발생

기본형

- ❶ `$("이벤트 대상").단독 이벤트 등록 메서드( );`
- ❷ `$("이벤트 대상").trigger("이벤트 종류");`

# 이벤트 등록 방식 알아보기 — 강제 이벤트 발생

```
07: <script>
08: $(function() {
09:     $(".btn1").click(function() {
10:         $(".btn1").parent().next()
11:         .css({"color" : "#f00"});
12:     });
13:
14:     $(".btn2").on({
15:         "mouseover focus" : function() {
16:             $(".btn2").parent().next()
17:             .css({"color" : "#0f0"});
18:         }
19:     });
20:
21:     $(".btn1").click();
22:     $(".btn2").trigger("mouseover");
23: });
24: </script>
```

[버튼1]에 click 이벤트가 발생하며 이벤트 핸들러가 실행됩니다.

[버튼2]에 마우스 포인터를 올리면 이벤트 핸들러(function() { 자바스크립트 코드 })가 자동으로 실행됩니다.

# 이벤트 등록 방식 알아보기 — 이벤트 제거 메서드

**기본형** | `$("이벤트 대상").off("제거할 이벤트 종류");`

```
20:
21:     $(".btn1").off("click");
22:     $(".btn2").off("mouseover focus");
23: });
24: </script>
```

[버튼1]에 등록된 클릭 이벤트를 제거합니다.

[버튼2]에 등록된 마우스 오버, 포커스 이벤트를 제거합니다.



## ready() / load() 이벤트 메서드

ready() 이벤트 메서드는 사용자가 사이트를 방문할 때 요청한 HTML 문서 객체(document)의 로딩이 끝나면 이벤트를 발생시킵니다. load() 이벤트 메서드는 외부에 연동된 소스(iframe, img, video)의 로딩이 끝나면 이벤트를 발생시킵니다. 다음은 ready()와 load() 이벤트 메서드의 기본형입니다.

### 기본형

- ❶ \$(document).ready(function() { 자바스크립트 코드; });
- ❷ \$(document).on("ready", function() { 자바스크립트 코드; });
- ❸ \$(window).load(function() { 자바스크립트 코드; });
- ❹ \$(window).on("load", function() { 자바스크립트 코드; });

## <a>, <form> 태그에 클릭 이벤트 적용 시 기본 이벤트 차단하기

### 기본형

#### ① return false를 이용한 차단 방식

```
$("#a 또는 form").이벤트 메서드(function( ) {  
    자바스크립트 코드;  
    return false;  
});
```

#### ② preventDefault( ) 메서드를 이용한 차단 방식

```
$("#a 또는 form").이벤트 메서드(function(e) {  
    e.preventDefault( );  
    자바스크립트 코드;  
});
```

prevent는 막다를, Default는 기본 이벤트를 의미합니다.

# 마우스 이벤트 — mouseover( ) / mouseout( ) / hover( ) 이벤트 메서드

## 기본형

### ① mouseover 이벤트 등록

```
$("#이벤트 대상 선택").mouseover(function( ) { 자바스크립트 코드; });  
$("#이벤트 대상 선택").on("mouseover",function( ) { 자바스크립트 코드; });
```

### ② mouseover 이벤트 강제 발생

```
$("#이벤트 대상 선택").mouseover( );
```

## 기본형

### ① mouseout 이벤트 등록

```
$("#이벤트 대상 선택").mouseout(function( ) { 자바스크립트 코드; });  
$("#이벤트 대상 선택").on("mouseout",function( ) { 자바스크립트 코드; });
```

### ② mouseout 이벤트 강제 발생

```
$("#이벤트 대상 선택").mouseout( );
```

## 기본형

### hover 이벤트 등록

```
$("#이벤트 대상 선택").hover(  
    function( ) { 마우스 오버 시 실행될 코드 },  
    function( ) { 마우스 아웃 시 실행될 코드 }  
);
```

## mouseenter()/mouseleave() 이벤트 메서드

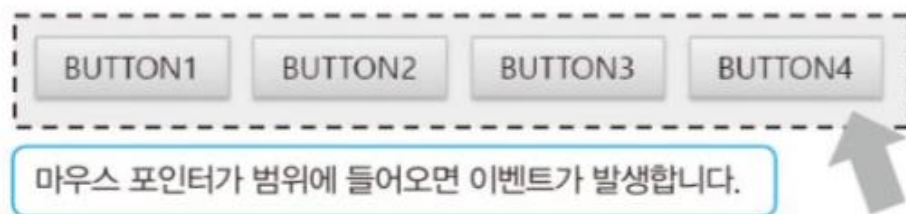
mouseenter() 이벤트 메서드는 대상 요소의 경계 범위에 마우스 포인터가 들어오면 이벤트를 발생시키고, mouseleave() 이벤트 메서드는 대상 요소의 경계 범위에서 마우스 포인터가 완전히 벗어나면 이벤트를 발생시킵니다. 그러면 다음과 같은 궁금증이 생길 것입니다.

“mouseover와 mouseenter 이벤트의 차이점은 무엇이고, mouseout과 mouseleave 이벤트의 차이점은 무엇인가요?”

mouseover는 대상 요소에 마우스 포인터가 올라가면 발생하는 이벤트이고, mouseenter는 이벤트 대상이 포함된 범위에 마우스 포인터가 들어오면 발생하는 이벤트입니다.



VS



마우스 포인터가 이벤트 대상 요소에 올라가면 이벤트가 발생합니다.

마우스 포인터가 범위에 들어오면 이벤트가 발생합니다.

mouseout은 대상 요소에서 마우스 포인터가 벗어나면 발생하는 이벤트이고, mouseleave는 대상이 포함된 범위에서 마우스 포인터가 완전히 벗어나면 발생하는 이벤트입니다.

BUTTON



VS

VS

마우스 포인터가 이벤트 대상 요소에서 벗어나면 이벤트가 발생합니다.

BUTTON1

BUTTON2

BUTTON3

BUTTON4



마우스 포인터가 대상이 포함된 범위에서 벗어나면 이벤트가 발생합니다.

`mouseenter()`, `mouseleave()` 이벤트 메서드는 해당 범위의 하위 요소를 제외한 '해당 범위'에 마우스 포인터가 '들어왔는지'(떠났는지)를 기준으로 이벤트를 발생시킵니다. 하지만 `mouseover()`, `mouseout()` 이벤트 메서드는 범위를 기준으로 하지 않습니다. 요소에 정확하게 마우스 포인터가 '올라왔는지'(또는 벗어났는지)를 기준으로 이벤트를 발생시킵니다.

#### 기본형

##### ① mouseenter 이벤트 등록

```
$("#이벤트 대상 선택").mouseenter(function() { 자바스크립트 코드; });
```

```
$("#이벤트 대상 선택").on("mouseenter", function() { 자바스크립트 코드; });
```

##### ② mouseenter 이벤트 강제 발생

```
$("#이벤트 대상 선택").mouseenter( );
```

#### 기본형

##### ① mouseleave 이벤트 등록

```
$("#이벤트 대상 선택").mouseleave(function() { 자바스크립트 코드; });
```

```
$("#이벤트 대상 선택").on("mouseleave", function() { 자바스크립트 코드; });
```

##### ② mouseleave 이벤트 강제 발생

```
$("#이벤트 대상 선택").mouseleave( );
```

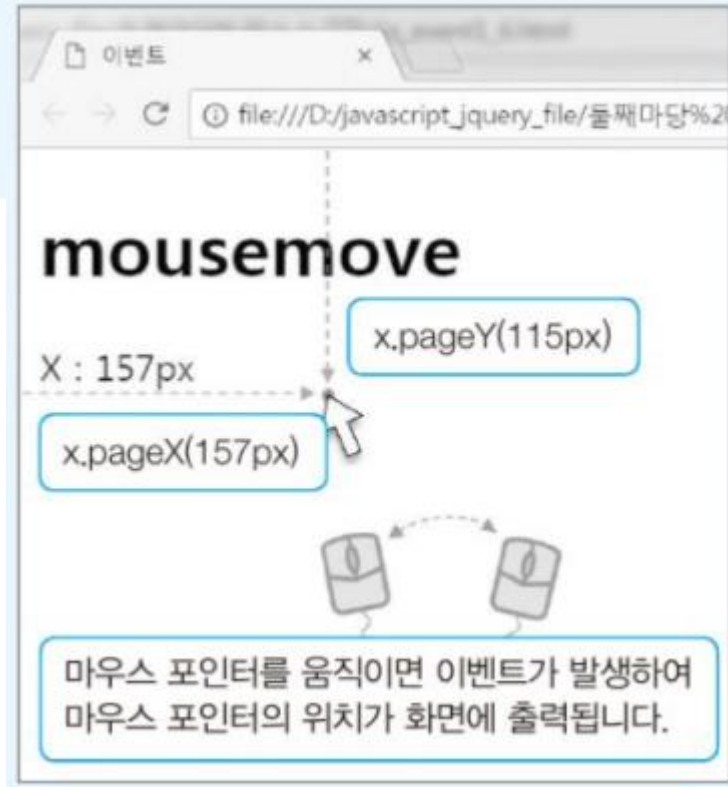
## 기본형

### ① mousemove 이벤트 등록

- ① `$("이벤트 대상 선택").mousemove(function( ) { 자바스크립트 코드; });`
- ② `$("이벤트 대상 선택").on("mousemove", function( ) { 자바스크립트 코드; });`

### ② mousemove 이벤트 발생

`$("이벤트 대상 선택").mousemove( );`





# 포커스 이벤트 — focus( ) / blur( ) 이벤트 메서드

## 기본형

### ① focus 이벤트 등록

```
$("#이벤트 대상 선택").focus(function( ) { 자바스크립트 코드; });  
$("#이벤트 대상 선택").on("focus", function( ) { 자바스크립트 코드; });
```

### ② focus 이벤트 강제 발생

```
$("#이벤트 대상 선택").focus( );
```

### ③ blur 이벤트 등록

```
$("#이벤트 대상 선택").blur(function( ) { 자바스크립트 코드; });  
$("#이벤트 대상 선택").on("blur", function( ) { 자바스크립트 코드; });
```

### ④ blur 이벤트 강제 발생

```
$("#이벤트 대상 선택").blur( );
```



# 포커스 이벤트 — focusin( ) / focusout( ) 이벤트 메서드

## 기본형

### ❶ focusin 이벤트 등록

```
$("#이벤트 대상 선택").focusin(function( ) { 자바스크립트 코드; });
```

```
$("#이벤트 대상 선택").on("focusin", function( ) { 자바스크립트 코드; });
```

### ❷ focusin 이벤트 강제 발생

```
$("#이벤트 대상 선택").focusin( );
```

### ❸ focusout 이벤트 등록

```
$("#이벤트 대상 선택").focusout(function( ) { 자바스크립트 코드; });
```

```
$("#이벤트 대상 선택").on("focusout", function( ) { 자바스크립트 코드; });
```

### ❹ focusout 이벤트 강제 발생

```
$("#이벤트 대상 선택").focusout( );
```

# 키보드로 마우스 이벤트 대응하기

## 마우스 이벤트에 대한 키보드 대응 이벤트

마우스 이벤트	키보드 이벤트
mouseover	focus
mouseout	blur

# 키보드로 마우스 이벤트 대응하기 — 키보드 접근성 X

```
<button class="btn">버튼</button>  
<p class="txt_1">내용1</p>
```

```
$(".btn").mouseover( function( ) {  
    $(".txt_1").hide( );  
} );
```

# 키보드로 마우스 이벤트 대응하기 — 키보드 접근성 ○

```
<button class="btn">버튼</button>  
<p class="txt_1">내용1</p>
```

```
$(".btn").on("mouseover focus", function( ) {  
    $(".txt_1").hide( );  
});
```

## change( ) 이벤트 메서드

change( ) 이벤트 메서드는 선택한 폼 요소의 값(value)을 새 값으로 바꿉니다. 그리고 포커스가 다른 요소로 이동하면 이벤트를 발생시킵니다. change( ) 이벤트 등록 메서드의 기본형은 다음과 같습니다.

### 기본형

#### ❶ focus 이벤트 등록

```
$("#이벤트 대상 선택").change(function( ) { 자바스크립트 코드; });
```

```
$("#이벤트 대상 선택").on("change", function( ) { 자바스크립트 코드; });
```

#### ❷ focus 이벤트 강제 발생

```
$("#이벤트 대상 선택").change( );
```

## keydown()/keyup()/keypress() 이벤트 메서드

keydown()와 keypress() 이벤트 메서드는 선택한 요소에서 키보드 자판을 눌렀을 때 이벤트를 발생시키거나 해당 이벤트를 강제로 발생시킵니다. 두 이벤트의 차이점을 보면 keydown()은 모든 키(한글 키 제외)에 대해서 이벤트를 발생시키지만 keypress()는 기능키(F1~F12, Alt, Ctrl, Shift, ↑, Backspace, Caps Lock, 한/영, Tab 등)에 대해서는 이벤트를 발생시키지 않습니다. 또한 키보드 이벤트 핸들러에서 생성된 이벤트 객체의 속성을 이용하면 고유 키의 코드값을 구할 수 있습니다. 그리고 이를 사용하여 단축키 기능을 만들 수 있습니다.

keyup() 이벤트 메서드는 자판의 키를 눌렀다 키에서 손을 떼면 이벤트를 발생시키거나 keyup 이벤트를 강제로 발생시킵니다. 다음은 keydown(), keyup(), keypress() 이벤트 메서드의 기본형입니다.

## ① keydown 이벤트 등록

- ① `$("이벤트 대상 선택").keydown(function( ) { 자바스크립트 코드; });`
- ② `$("이벤트 대상 선택").on("keydown", function( ) { 자바스크립트 코드; });`

## ② keydown 이벤트 강제 발생

`$("이벤트 대상 선택").keydown( );`

## ③ keyup 등록

- ① `$("이벤트 대상 선택").keyup(function( ) { 자바스크립트 코드; });`
- ② `$("이벤트 대상 선택").on("keyup", function( ) { 자바스크립트 코드; });`

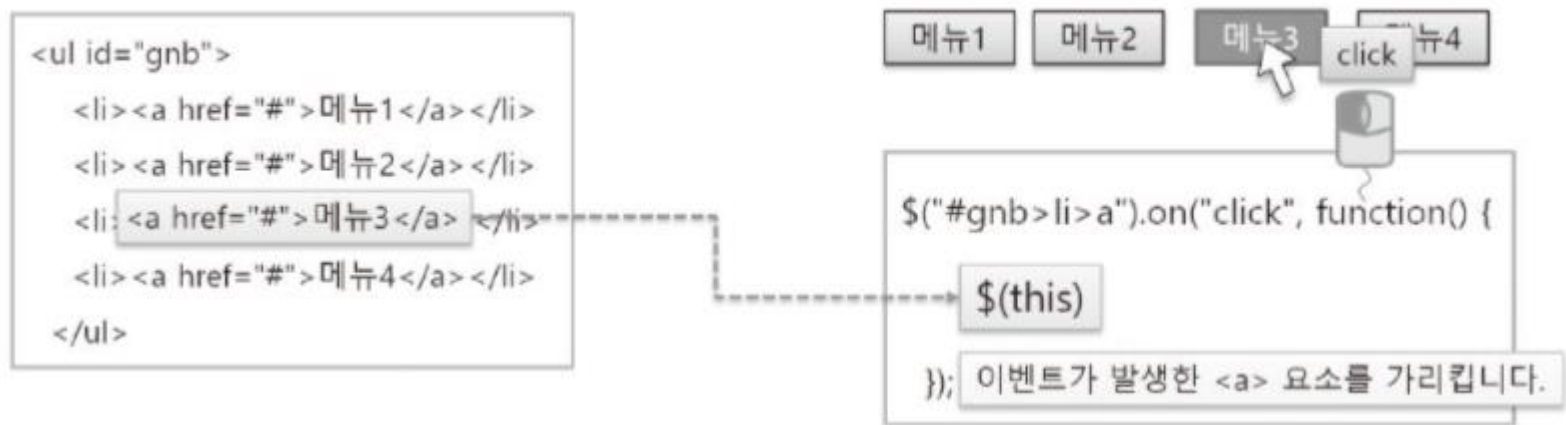
## ④ keyup 이벤트 강제 발생

`$("이벤트 대상 선택").keyup( );`

# \$(this) 선택자

이벤트 핸들러에서 \$(this)를 사용하면 이벤트가 발생한 요소를 선택하여 이벤트가 발생한 요소를 추적할 수 있습니다.

다음은 id 속성값이 "gnb"인 요소의 <a> 요소를 선택하여 click 이벤트를 등록하는 과정을 그림으로 나타낸 것입니다. 이때 '메뉴3'을 클릭하면 이벤트 핸들러에 사용된 \$(this)는 이벤트가 발생한 <a href="#">메뉴3</a>를 선택합니다.





# index( ) 인덱스 반환 메서드

index( ) 인덱스 반환 메서드는 이벤트를 등록한 요소 중 이벤트가 발생한 요소의 인덱스값을 반환합니다.

기본형

```
$("#이벤트 대상 선택").on("이벤트 종류", function( ) {  
    $("#이벤트 대상 선택").index(this);  
});
```

