jQuery 이벤트

재단법인 대덕인재개발원
Daedeok Human Resource Development Foundation

w3schools.com 참고

- [TUTORIALS] > [JQUERY] > [jQuery Tutorial] > [jQuery Events]
  - https://www.w3schools.com/jquery/jquery_events.asp

- [TUTORIALS] > [JQUERY] > [jQuery References] > [jQuery Events]
  - https://www.w3schools.com/jquery/jquery_ref_events.asp

이벤트

- jQuery의 이벤트에는 기존 자바스크립트의 이벤트가 모두 존재
- jQuery를 사용하면 기존 자바스크립트의 이벤트를 연결할 때보다 훨씬 간편하게 이벤트를 연결할 수 있음
- jQuery로 이벤트를 연결하는 가장 기본적인 방법은 on( ) 메서드 사용
- jQuery가 스스로 event 객체를 정형화하므로 jQuery의 이벤트 객체는 모든 브라우저가 같은 방법으로 사용하고 같은 속성을 갖음

```
$("p").click(function(e){
    var eventType = e.type;
})
```

event 객체

이벤트 객체

- 모든 이벤트의 정보를 event 객체로 제공
- 속성
  - type : 이벤트 종류 (예: click)
  - target : 이벤트가 발생한 객체(요소)
  - pageX, pageY : 문서의 좌상단을 기준으로 한 마우스 좌표
  - screenX, screenY : 화면상의 좌표
  - button : 눌러진 마우스 버튼
    (0 : 왼쪽 버튼, 1 : 휠(중간) 버튼, 2 : 오른쪽 버튼)
- 메소드*
  - preventDefault()
  - stopPropagation()

마우스 이벤트

메소드	설 명
click / dblclick	마우스 클릭 / 더블클릭 시 발생
mousedown	마우스 버튼을 누를 때 발생
mouseup	마우스 버튼을 땔 때 발생
mouseenter	마우스가 요소의 경계 외부에서 내부로 이동 시
mouseleave	마우스가 요소의 경계 내부에서 외부로 이동 시
hover	mouse enter와 leave의 조합* $(*selector*).hover(*inFunction*, *outFunction*)
mousemove	마우스가 움직일 때 발생
mouseout	마우스가 요소를 벗어날 때 발생
mouseover	마우스가 요소 안에 들어올 때 발생

기타 이벤트

메소드	설 명
keypress	키보드 키를 누르고 있을 때 발생
keydown	키보드 키를 누를 때 발생
keyup	키보드 키를 땔 때 발생
submit	입력 정보를 제출할 때 발생
change	입력 요소의 정보가 변경 되었을 때 발생
focus / blur	입력 요소가 포커스를 얻었을 때 / 잃었을 때
load	DOM이 로드 되었을 때 발생
resize	DOM 사이즈가 변경되었을 때 발생
ready	문서가 완전히 로드 되었을 때 $(document).ready(…)
on / off	*선택한 요소에 여러 이벤트를 넣을 때 / 지울 때

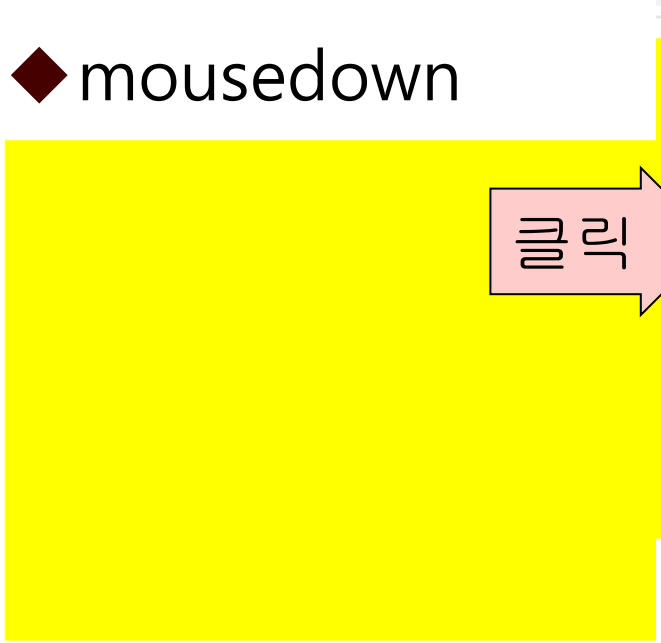
on 이벤트

◆선택한 요소에 하나 이상의 이벤트를 넣을 때 사용

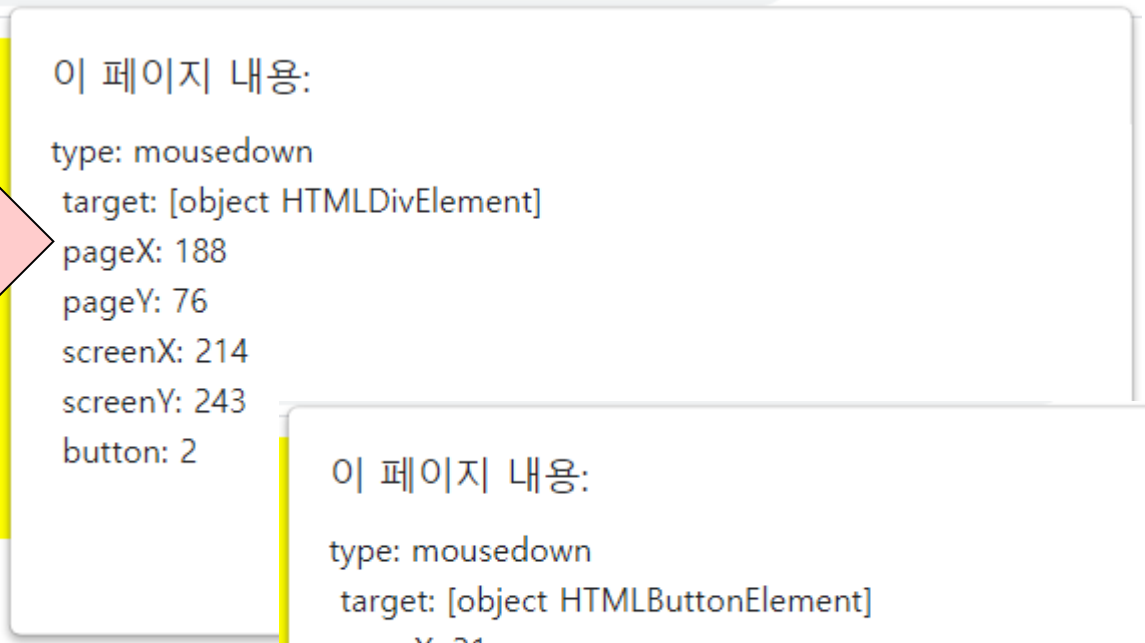
```
$("p").on({
  mouseenter : function(){
    $(this).css("background-color", "gray");
  }
  , mouseleave : function(){
    $(this).css("background-color", "blue");
  }
  , click : function(){
    $(this).css("background-color", "yellow");
  }
});
```

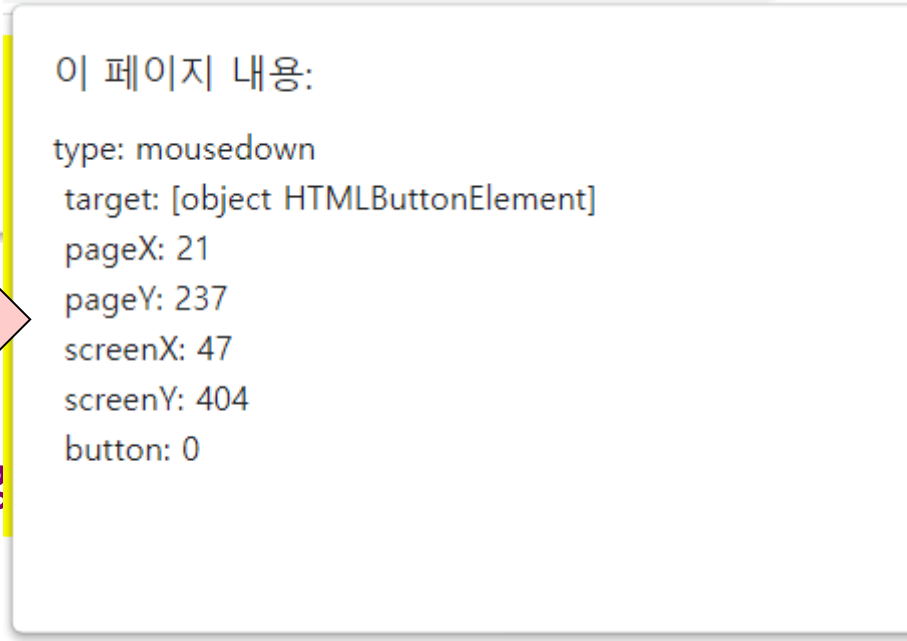
마우스 이벤트 예제1

◆mousedown



클릭





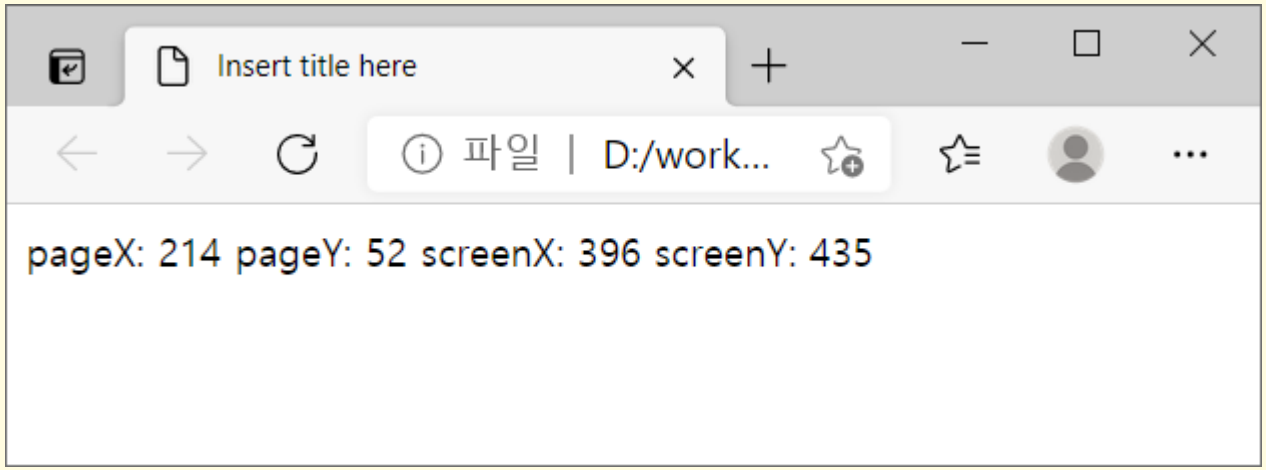
버튼 클릭

```
<body>
  <div style="width:300px;heig
  color:yellow;"></div>
  <br>
  <button type="button">버튼</button>
</body>
```

마우스 이벤트 예제2

◆mousemove

```
<script>
//마우스를 움직이면 좌표값을 얻어서 div에 출력
// div에 출력 - $("~").html(~)
</script>

<body>
    <div id="log" style="height: 50px;"></div>
</body>
```



마우스 이벤트 예제3

1) 마우스를 이미지 위에 올리면 다른 이미지로 변경 ➔ mouseover
2) 마우스가 이미지 위에서 나가면 원래 이미지로 돌아옴 ➔ mouseout
3) 이미지 더블 클릭 시 이미지 사라짐 ➔ dblclick
4) 이미지가 모두 사라지면 이미지 하단에 [보이기] button이 나타남

힌트

- $("img"). mouseover(function(e){
  //e.target.src 이용
  })
- $(~).css("display", "~")
- $(~).each(function(){}) 또는 $.each(~, function(){})

이벤트 객체 메소드 1

◆ *event*.preventDefault()

- 요소의 기본 작업이 발생하지 않도록 함.
- 예) 제출 버튼이 양식을 제출하지 못하도록
  링크가 URL 을 따르지 못하도록

```
$("a").click(function(event){
      event.preventDefault();
});
```

또는

```
$("a").click(function(event){
      return false;
});
```

a 태그를
클릭해도
링크 페이지로
이동하지 않는다!

preventDefault 예제

◆ 이벤트 기본 처리 금지

```
<script>
//a 태그 클릭 시 prompt로 나이를 입력 받아, 15미만이면 페이지 이동 금지
~
</script>
</head>
<body>
  <p><a id="link" href="http://www.naver.com">네이버</a>로 이동
  </p>
</body>
```

이벤트 객체 메소드 2

◆ *event*.stopPropagation()

- 부모 요소에 대한 이벤트 버블링을 중지하여 부모 이벤트 처리기가 실행되지 않음.

p클릭➔div클릭 수행
span클릭➔p, div클릭 수행X

```
$(document).ready(function(){
   $("span").click(function(e){
      alert("span");
      e.stopPropagation();//return false;도 가능
   });
   $("p,div").click(function(){ alert(this); });
});
… <div style="border:1px solid;background-color:green;">
      div
      <p style="background-color:pink">
         p
         <br>
         <span style="background-color:white">span</span>
      </p>
   </div>…
```

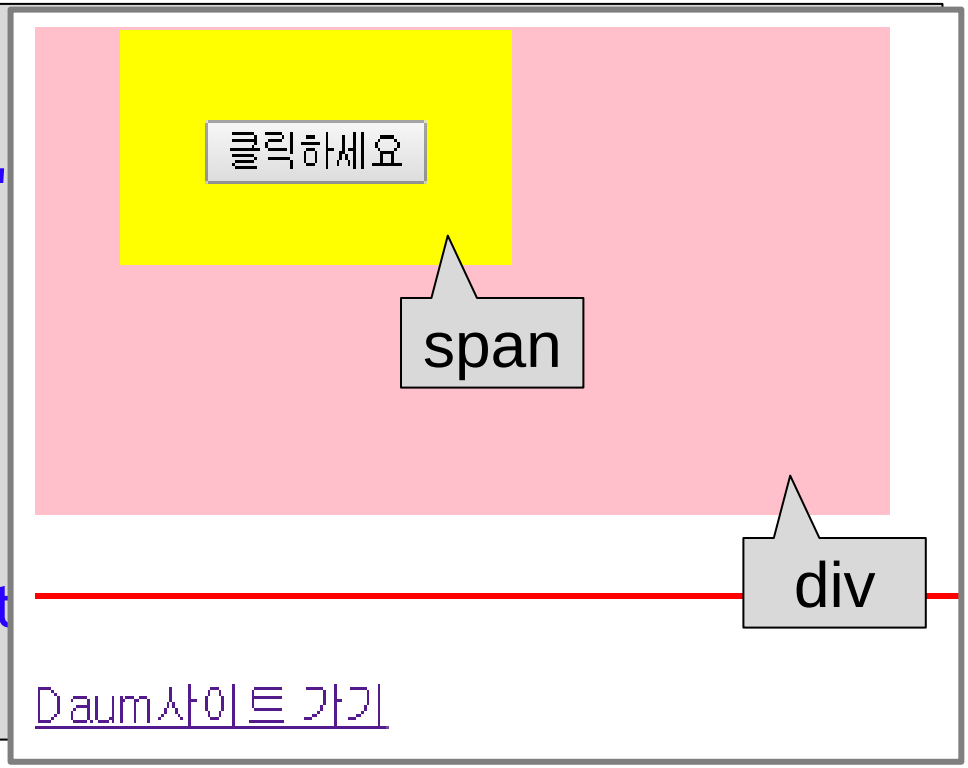
stopPropagation 예제1

◆ 이벤트 버블링 막기

```
<script>
//span클릭:배경색 빨강, h1클릭:배경색 녹색으로 변경
//span 클릭해도 h1에 이벤트가 전달되지 않아야 함
~
</script>
</head>
<body>
  <h1>
    큰 제목 안에 <span>작은 영역</span>이 있다.
  </h1>
</body>
```

stopPropagation 예제2

```
<div>
  <span>
    <button type="button"
  </span>
</div>
<br>
<hr color="red">
<br>
<a href="http://www.daum.net
Daum 으로 이동</a>
```



1) button 클릭 시, 버튼의 테두리 설정

2) span 클릭 시, span의 테두리 설정.

3) div 클릭 시, div의 테두리 설정. div안에 "Hello~" 글자 추가.

4) a 클릭 시, 프롬프트로 태어난 년도 4자리를 입력 받아서 20세 이상인 경우만 링크 이동하기

이벤트 연결 메소드 1

- 이벤트(Event) : 웹 페이지에서 발생하는 사용자의 동작들
- 이벤트 핸들러(Event Handler) : 이벤트를 처리 하는 것
- 이벤트 바인딩(Event Binding) : 특정 요소에 이벤트 핸들러를 연결 하는 것

이벤트 연결	이벤트 제거	
$(*selector*).bind(...)	$(*selector*).unbind(...)	v3.0에서 사용 안함
↓	↓	
$(*selector*).live(...)	$(*selector*).die(...)	v1.9에서 삭제됨
↓	↓	
$(*selector*).delegate(...)	$(*selector*).undelegate(...)	v3.0에서 사용 안함
↓	↓	
$(*selector*).on(...)	**$(*selector*).off(...)**	현재 버전

이벤트 연결 메소드 2

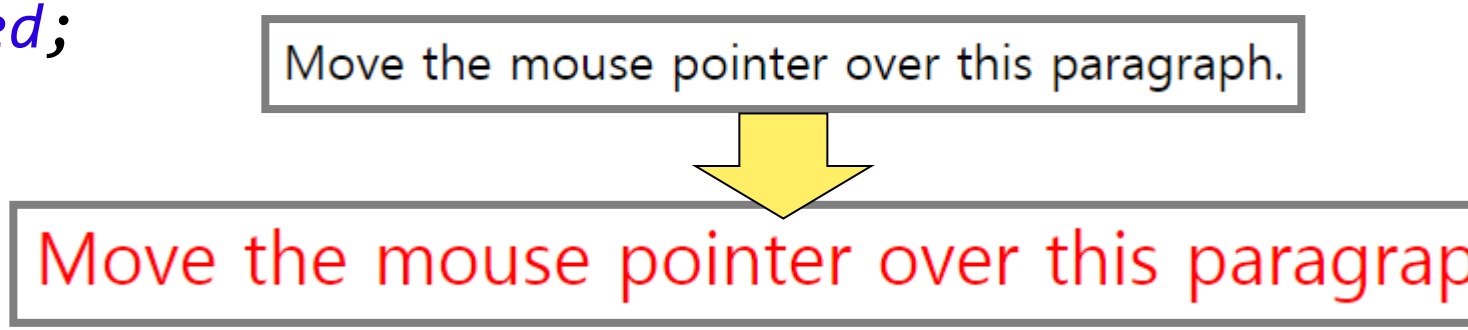
- $(*selector*).on(***event***, *childSelector, data,* ***function***, *map*)
  - 선택된 요소와 하위 요소에 한 개 이상의 이벤트를 바인딩
  - 현재 로드 된 요소 및 스크립트로 생성될 새로운 요소에도 적용됨

- $(*selector*).one(***event***, *data,* ***function***)
  - 선택된 요소와 하위 요소에 한 개 이상의 이벤트를 바인딩. 이벤트가 한 번만 수행되고 제거됨

- $(*selector*).off(***event***, *selector, function(eventObj), map*)
  - 선택된 요소에 on메소드로 연결된 이벤트를 제거

on 사용법 1(1/3)

```
<script>
//p 요소에 마우스를 올리면 intro 라는 스타일이
적용되고 마우스 벗어나면 적용됐던 스타일이 사라진다.
…

<style>
.intro {
  font-size: 1.5em;
  color: red;
}
</style>
…
<body>
  <p>Move the mouse pointer over this paragraph.</p>
</body>
```



on 사용법 1(2/3)

```
<script>
//p 요소에 마우스를 올리면 intro 라는 스타일이
적용되고 마우스 벗어나면 적용됐던 스타일이 사라진다.
$(document).ready(function(){
    $("p").on("mouseover", function(){
        this.className = "intro";
    });
    $("p").on("mouseout", function(){
        this.className = "";
    });
});
</script>
<style>
    .intro { font-size: 1.5em;  color: red; }
</style>
```

on 사용법 1(3/3)

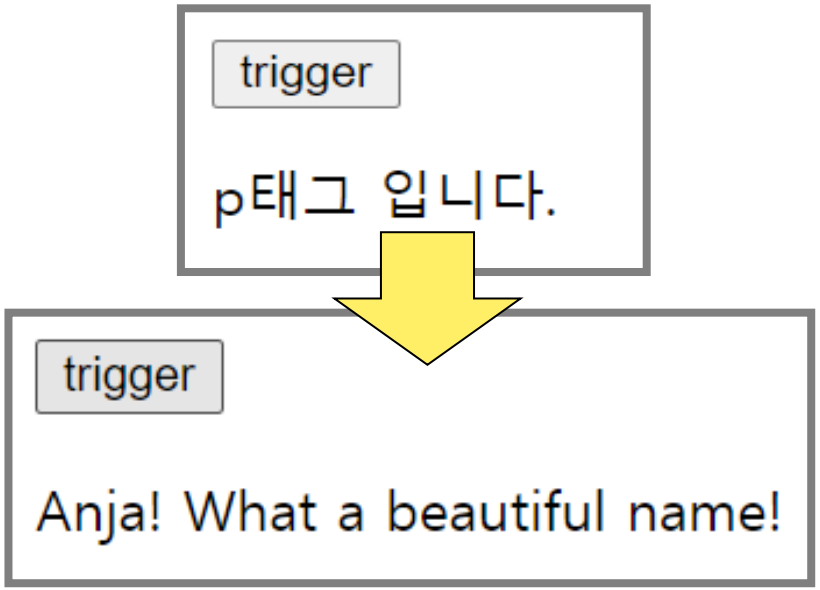
```
<script>
//p 요소에 마우스를 올리면 intro 라는 스타일이
적용되고 마우스 벗어나면 적용됐던 스타일이 사라진다.
$(document).ready(function(){
    $("p").on("mouseover mouseout", function(){
        $(this).toggleClass("intro");
    });
});
</script>
<style>
    .intro { font-size: 150%;  color: red; }
</style>
```

on 사용법 2

```
<script>
//id test인 p 요소에 마우스를 올렸을 때, 벗어났을 때, 클릭했을 때 body 배경색이 변함
$(document).ready(function(){
  $("#test").on({
    mouseover : function(){
      $("body").css("background-color", "lightblue");
    }
    ,mouseout : function(){
      $("body").css("background-color", "lightgray");
    }
    ,click : function(){
      $("body").css("background-color", "yellow");
    }
  });
});
</script>
<body>
  <p id="test">Click this paragraph.</p>
</body>
```

on 사용법 3 - 사용자 정의 이벤트

```
<script>
$(document).ready(function(){

  $("p").on("myOwnEvent", function(event, name){
    $(this).text(name + ", 멋진 이름이군!").show();
  });

  $("button").click(function(){
    $("p").trigger("myOwnEvent", "Ann");
  });

});
</script>
<body>
  <button>trigger</button>
  <p>p태그 입니다. </p>
</body>
```



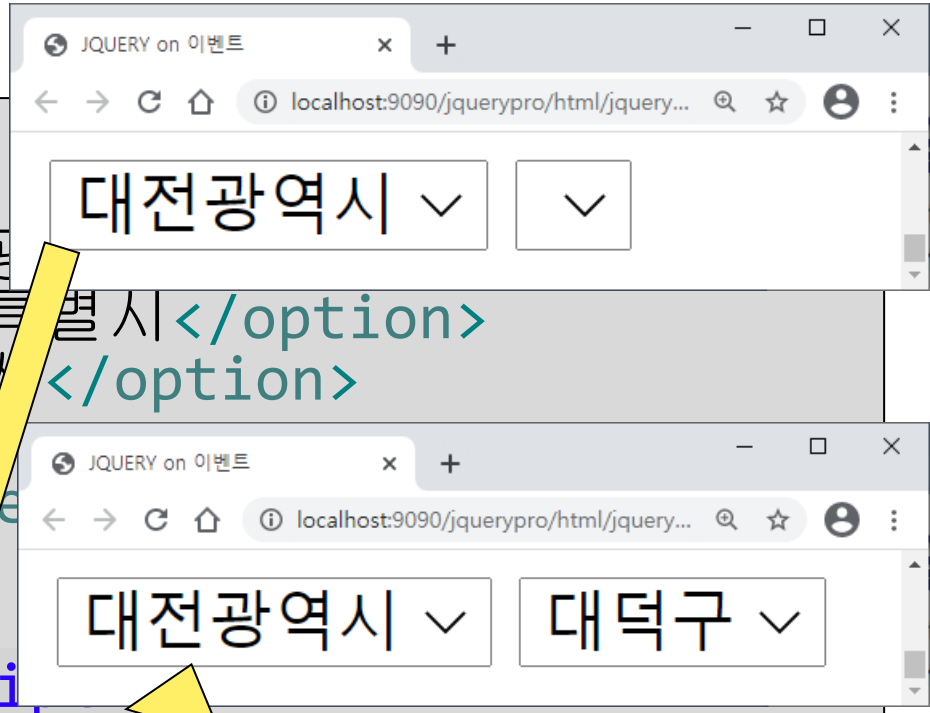
이벤트 연결 메소드 3

- $(*selector*).trigger(***event***, *eventObj, param1, param2,...*)
  - 이벤트의 기본 동작을 발생시킴
  - 선택된 모든 요소에 동작

- $(*selector*).triggerHandler(***event***, *param1, param2,...*)
  - 함수만 실행시키고, 이벤트의 기본 동작은 발생시키지 않음
  - 선택된 첫 번째 요소에만 동작
  - 버블링 되지 않음

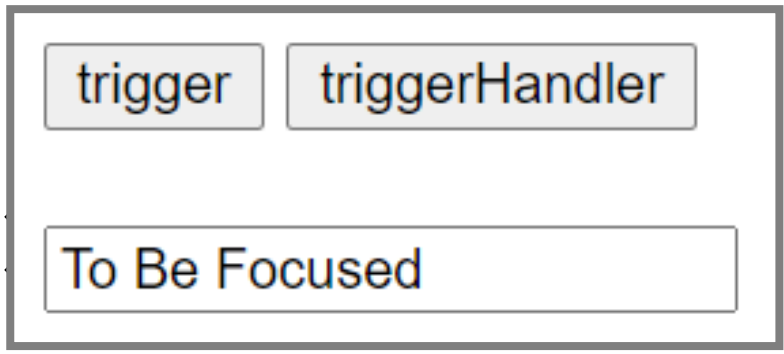
trigger 예제1

```
<body>
  <select id="sido">
    <option value="1">대전광역시</option>
    <option value="2">서울특별시</option>
    <option value="3">세종시</option>
  </select>
  <select id="gugun"></select>
</body>

<script type="text/javascript">
  $(document).ready(function(){
    $("#sido").change(function(){
      // 구군 셀렉 박스에 데이터 세팅하는 로직구현
    });

    $("#sido").trigger("change");
  });
</script>
```



trigger 예제2

```
<script>
$(document).ready(function(){
  $("#old").click(function(){
    $("input").trigger("focus")
  });
  $("#new").click(function(){
    $("input").triggerHandler("focus");
  });
  $("input").focus(function(){
    $("<span>포커스</span>").appendTo("body")
    .fadeOut(1000);
  });
});
</script>
```




```
<body>
  <button id="old">trigger</button>
  <button id="new">triggerHandler</button>
  <br/><br/>
  <input type="text" value="To Be Focused"/>
</body>
```

이벤트 등록 메소드 배경

◆ bind()는 선택된 모든 요소에 핸들러를 등록한다.

- 그러나 스크립트에 의해 새로 추가되는 요소에는 핸들러가 지정되지 않아 이벤트가 발생해도 핸들러가 호출되지 않음

◆ 미래에 추가될 요소들에 대해서도 핸들러를 미리 등록하는 메서드가 필요해짐 - delegate

- 특정 요소에 대해 핸들러를 지금 당장 등록하는 것이 아니라 문서가 변경될 때마다 핸들러가 등록되어야 하므로 document 수준에서 메소드를 호출한다. (document 는 이벤트 대상객체의 부모가 될 수 있다.)
- 이처럼 실행 중에 자동으로 등록되는 살아있는 핸들러를 라이브 핸들러라고 한다.

◆ version 3.0에서 등록 메소드 새로 도입 - on, off

- 선택자가 있으면 delegate와 같고 없으면 bind와 같음