

이벤트

## ▶ jQuery 이벤트 객체

### ✓ event객체 속성 값

속성	내용
event.currentTarget	이벤트 발생한 bubbling범위 내 이벤트핸들러의 this와 같은 값을 가짐
event.data	이벤트 핸들러로부터 받아오는 값
event.delegateTarget	현재 호출된 이벤트핸들러에 연결된 객체 리턴
event.namespace	사용자지정 이벤트명 리턴
event.pageX	왼쪽기준 마우스위치 리턴
event.pageY	위쪽기준 마우스위치 리턴
event.relatedTarget	마우스가 요소에 들어오거나 나오는 요소리턴
event.result	이벤트핸들러가 반환한 값을 보관
event.target	이벤트가 발생한 요소 리턴
event.timestamp	이벤트가 발생한 시간, 밀리세컨초로 리턴
event.type	발생한 이벤트를 리턴
event.which	키/마우스를 눌렀을 때 그 키의 정수 값 리턴

## ▶ jQuery 이벤트 객체

### ✓ event객체 메소드

메소드	내용
event.isDefaultPrevented()	preventDefault()가 호출되었는지 확인
event.isImmediatePropagationStopped()	stopImmediatePropagation()이 호출되었는지 확인
event.isPropagationStopped()	stopPropagation()호출됐는지 확인
event.preventDefault()	기본이벤트 발생 방지
event.isImmediatePropagation()	실행된 한 개 이벤트 제외하고 나머지는 동일한 이벤트 중단
event.stopPropagation()	bubbling을 막는 메소드

## ▶ jQuery 이벤트 연결

### ✓ 이벤트 연결 메소드

요소 객체에 이벤트 발생 시 연결된 이벤트 핸들러 지정하는 메소드

메소드	내용
<code>\$( 's' ).bind( '이벤트명 [이벤트명]', function( ){ } )</code>	지정 이벤트에 이벤트 핸들러 지정 * 다수 이벤트 지정 시 띄어쓰기로 구분
<code>\$( 's' ).bind( '이벤트명 [이벤트 명]', { 객체 }, function() { } )</code>	이벤트처리 객체를 인자로 넣어서 처리 event객체 data에 들어감
<code>\$( 's' ).on( '이벤트명[,이벤트명]', function( ){ } )</code>	지정 이벤트에 이벤트 핸들러 지정
<code>\$( 's' ).on( { 객체 } )</code>	이벤트처리 객체를 인자로 넣어서 처리 event객체 data에 들어감
<code>\$( 's' ).on( '이벤트', '선택자', function() { } )</code>	선택된 요소 중 새로 동적으로 생성되는 선택자에 이벤트를 적용 동적적용

\* bind메소드는 3.0버전부터 사용하지 않으니 on메소드 사용 권고

## ▶ jQuery 이벤트 연결

### ✓ one()

이벤트를 연결하여 한 번만 실행하는 메소드

메소드	내용
<code>\$(s).one('이벤트명 [이벤트명]', '핸들러')</code>	핸들러에 있는 기능을 한 번만 실행 이벤트 명을 다수로 설정하면 각 이벤트 별 한 번씩만 실행

## ▶ jQuery 이벤트 삭제

### ✓ off()

on() 메소드로 지정된 이벤트 핸들러를 제거하는 메소드로  
1.7버전부터 unbind, die, undelegate 메소드를 대체

메소드	내용
<code>\$( 's' ).off( "이벤트명 [이벤트명]", [ "선택자" ] );</code>	선택자에 이벤트명이 있는 이벤트제거
<code>\$( 's' ).off( "이벤트명 [이벤트명]", [ "핸들러명" ] );</code>	핸들러명이 지정된 이벤트만 제거
<code>\$( 's' ).off( event );</code>	이벤트핸들러에 인자로 event를 받아 event객체를 기준으로 이벤트삭제

# ▶ jQuery 이벤트 연결

## ✓ 간단한 연결 이벤트

```
$( 's' ).method(function(event){});
```

메소드	내용	메소드	내용
blur	요소가 focus해제 시	scroll	스크롤을 움직일 때
focus	요소가 focus받을 때	click	클릭했을 때
focusin	요소, child가 focus받을 때	dblclick	더블 클릭 했을 때
focusout	요소,child가 focus해제 시	mousedown	마우스 왼쪽 버튼 누를 때
resize	윈도우 크기 변경 시	mouseup	마우스 왼쪽 버튼 땔 때
mouseover	마우스가 요소에 있을 때	mousemove	마우스가 요소, child에서 움직일 때
mouseout	마우스가 요소,child에서 나갈 때	mouseenter	마우스가 요소에 들어올 때
mouseleave	마우스가 요소에서 나갈 때	change	요소의 값이 변경 되었을 때
select	텍스트가 선택 됐을 때 (textarea,filed)	submit	form이 전송 되었을 때
keydown	키를 눌렀을 때	keypress	키를 눌렀을 때 (alt, ctrl, shift, esc인식X)

## ▶ jQuery 이벤트 연결

### ✓ hover()

mouseenter이벤트와 mouseleave이벤트를 동시에 연결  
만일 한 개의 이벤트 핸들러가 설정되면 두 이벤트 모두 작동  
이벤트 발생 객체는 함수에서 this로 호출 가능

```
$( 's' ).hover( 'function(){}', 'function(){}' );
```

\* 첫 번째 함수가 mouseenter에서 작동, 두 번째 함수가 mouseleave에서 작동



## ▶ jQuery 이벤트 연결

### ✓ trigger()

특정 이벤트나 기본 이벤트를 강제로 발생시키는 메소드로  
사용자 정의 이벤트 발생 시 사용  
이벤트 발생 시 인자 값 전달 가능

```
$( 's' ).trigger( '이벤트명', '전달 값', '전달 값', ... );
```

## ▶ jQuery 이벤트 전달

### ✓ 이벤트 전달 제거

요소에 기본적으로 설정되어 있는 이벤트 제거(ex. <a>, submit)  
이벤트가 상위 요소로 전달되는 것을 제거함

메소드	내용
<code>event.preventDefault();</code>	기본이벤트를 제거하는 메소드
<code>event.stopPropagation();</code>	이벤트 전달을 제거하는 메소드