

ch09. jQuery02

학습 목차

1. 이벤트 연결
2. 이벤트 제거
3. 이벤트 사용
4. 시각 효과
5. 애니메이션

학습 목표

- jQuery 이벤트 연결 방법을 이해한다
- jQuery 이벤트를 동적 효과에 활용할 수 있다
- jQuery 이벤트 타입을 이해한다
- jQuery 효과를 활용할 수 있다
- jQuery 애니메이션을 이해하고 활용할 수 있다

1. 이벤트 연결

- on() 메소드 이용
- 두 가지 형태
 1. `$(selector).on(events, [selector] [data], handler)`
 - events: 이벤트 이름
 - selector : 이벤트를 연결할 후손 선택자
 - data : 이벤트 핸들러에 전달할 데이터, 이벤트 핸들러 매개변수 `event.data`에 담겨 전달
 - handler: 이벤트 핸들러, 첫번째 인자로 이벤트 객체가 묵시적으로 전달
 2. `$(selector).on(object)`
 - object: 이벤트 이름을 속성으로, 이벤트 리스너를 속성의 값으로 갖는 객체

```
$(selector).on({  
  eventName_0: eventHandler_0,  
  eventName_1: eventHandler_1,  
  eventName_2: eventHandler_2  
});
```

1. 이벤트 연결

- on() 메소드 이용 - ❶ 방법으로 이벤트 연결

```
<head>
<title>jquery 이벤트 - 이벤트 연결 메소드</title>
<script src="https://code.jquery.com/jquery-3.7.0.js"> </script>
<script>
$(function () {
    $('#box').css({
        width: 100,
        height: 100,
        background: 'orange'
    }).on('click',function() { //이벤트 연결
        $(this).css('background', 'red');
    }).on('mouseenter',function () {
        $(this).css('background', 'blue');
    }).on('mouseleave', function () {
        $(this).css('background', 'orange');
    });
});
</script>
</head>
<body> <div id="box"> </div> </body>
```

1. 이벤트 연결

- on() 메소드 이용 - ❷ 방법으로 이벤트 연결

```
<head>
<script src="https://code.jquery.com/jquery-3.7.0.js"> </script>
<script>
$(function () {
    $('#box').css({
        width: 100,
        height: 100,
        background: 'orange'
    }).on({
        click: function () {
            $(this).css('background', 'red');
        },
        mouseenter: function () {
            $(this).css('background', 'blue');
        },
        mouseleave: function () {
            $(this).css('background', 'orange');
        }
    });
});
</script>
</head>
<body>
<div id="box"> </div>
</body>
```

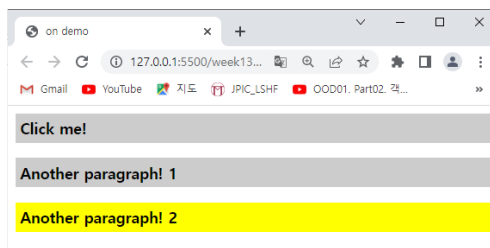
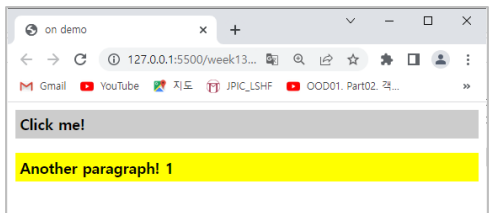
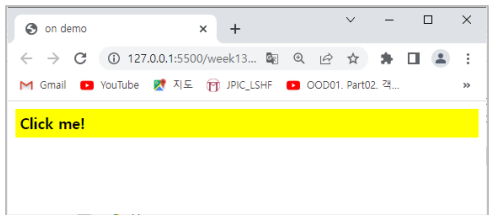
1. 이벤트 연결

- 단축 이벤트 연결 방식(deprecated)
 - 이벤트 이름을 메소드 이름으로 사용

```
<script>  
  //HTML 문서 객체의 로딩이 완료된 후 이벤트 발생  
  $(document).ready(function(){  
    //document : 이벤트 대상  
    //ready : 이벤트 종류 -> 메소드 이름 사용  
    //function() : 이벤트 핸들러  
  });  
</script>
```

1. 이벤트 연결

- click 이벤트

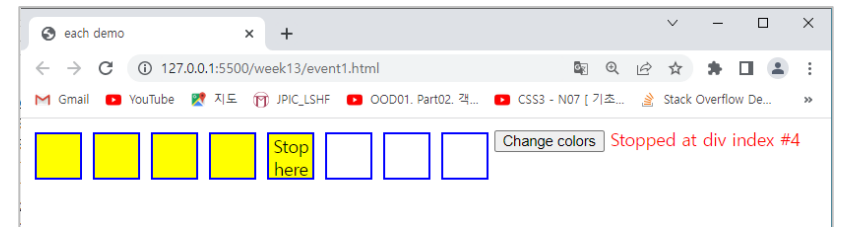
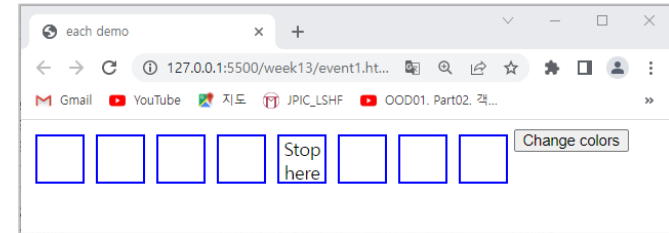


```
<head>
  <meta charset="utf-8">
  <style>
    p { background: yellow; font-weight: bold; cursor: pointer; padding: 5px; }
    p.over { background: #ccc; }
    span { color: red; }
  </style>
  <script src="https://code.jquery.com/jquery-3.7.0.js"> </script>
</head>
<body>
  <p>Click me!</p>
  <span> </span>
  <script>
    let count = 0;
    $("body").on("click", "p", function () {
      $(this).addClass('over');
      $(this).after("<p>Another paragraph! " + (++count) + "</p>");
    });
  </script>
</body>
```


1. 이벤트 연결

- click 이벤트

```
<head>
  <meta charset="utf-8">
  <style>
    div { width: 40px; height: 40px; margin: 5px; float: left; border: 2px blue solid; text-align: center; }
    span { color: red; }
  </style>
  <script src="https://code.jquery.com/jquery-3.7.0.js"> </script>
</head>
<body>
  <button>Change colors</button>
  <span></span>
  <div></div> <div></div> <div></div> <div></div>
  <div id="stop">Stop here</div>
  <div></div> <div></div> <div></div>
  <script>
    $("button").on("click", function () {
      $("div").each(function(index, element) {
        $(element).css("backgroundColor", "yellow");
        if ($(this).is("#stop")) {
          $("span").text("Stopped at div index #" + index);
          return false; //each() 종료
        }
      });
    });
  </script>
</body>
```



1. 이벤트 연결

- 단축 이벤트 연결 방식 – hover 이벤트
 - 두 개의 이벤트를 동시에 연결
 - hover(mouseenter, mouseleave)

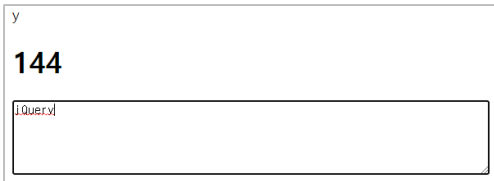
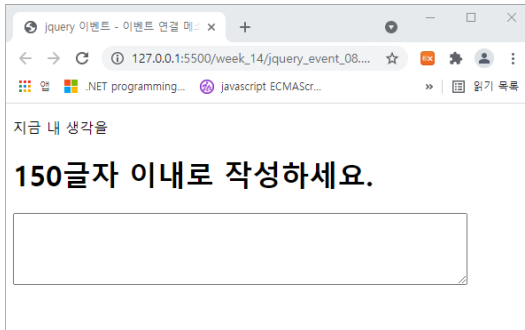


```
<script>
$( "li" ).on({
  mouseenter :function() {
    $( this ).append( $( "<span> ***</span>" ) );
  },
  mouseleave : function() {
    $( this ).find( "span" ).last().remove();
  }
});
</script>
```

```
<head>
  <title>jquery 이벤트 - 이벤트 연결 메소드</title>
<style>
  ul { margin-left: 20px; color: blue; }
  span { color: red; }
<script src="https://code.jquery.com/jquery-3.7.0.js"> </script>
<script>
  $(function () {
    $("li").hover(
      function () { //mouseenter
        $(this).append($( "<span> ***</span>" ));
      }, function () { //mouseleave
        $(this).find("span").last().remove();
      }
    );
  });
</script>
</head>
<body>
  <ul>
    <li>Milk</li> <li>Bread</li> <li>Chips</li> <li>Socks</li>
  </ul>
</body>
```

1. 이벤트 연결

- 키보드 이벤트
 - keydown: 키보드 키를 누를 때 발생함
 - keypress: 글자가 입력될 때 발생함
 - keyup: 키보드 키를 떼 때 발생

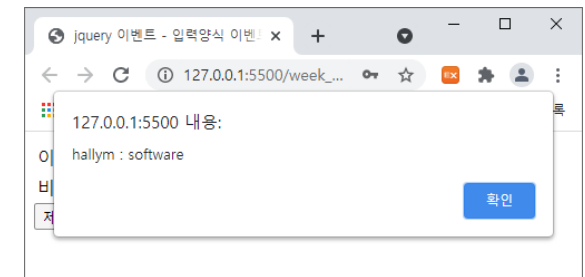
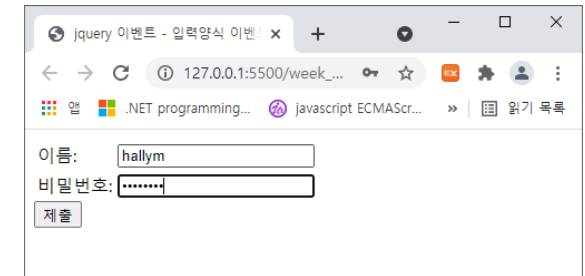
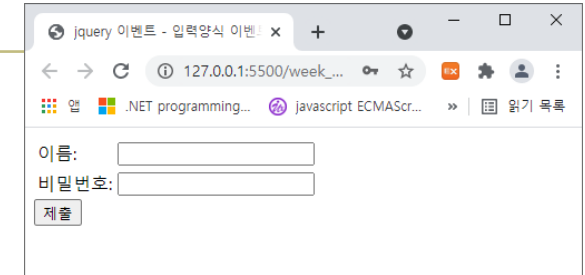


```
<head>
  <title>jquery 이벤트 - 키보드 이벤트</title>
  <script src="https://code.jquery.com/jquery-3.7.0.js"> </script>
  <script>
    // 이벤트를 연결합니다.
    $(function () {
      $('textarea').on('keyup', function(e) {
        // 남은 글자 수를 구합니다.
        let inputLength = $(this).val().length;
        let remain = 150 - inputLength;
        // 문서 객체에 입력합니다.
        $('h1').html(remain);
        $('p').text(e.key);
      })
    });
  </script>
</head>
<body>
  <div>
    <p>지금 내 생각을</p>
    <h1>150글자 이내로 작성하세요.</h1>
    <textarea cols="70" rows="5"> </textarea>
  </div>
</body>
```

1. 이벤트 연결

- submit 이벤트

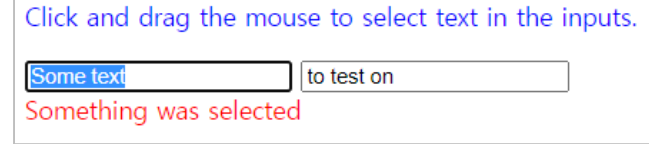
```
<head>
  <script src="https://code.jquery.com/jquery-3.7.0.js"> </script>
  <script>
    $(function(){
      $('#my-form').on('submit', function (event) {
        let name = $('#name').val(); // 입력 양식의 value를 가져옵니다
        let password = $('#password').val();
        alert(name + ' : ' + password); // 경고창으로 출력합니다.
        event.preventDefault(); // 기본 이벤트를 제거합니다.
      });
    });
  </script>
</head>
<body>
  <form id="my-form">
    <table>
      <tr><td>이름: </td><td><input type="text" name="name" id="name" /> </td></tr>
      <tr><td>비밀번호: </td><td><input type="password" name="password" id="password" /> </td></tr>
    </table>
    <input type="submit" value="제출" />
  </form>
</body>
```



1. 이벤트 연결

- select 이벤트

```
<head>
  <meta charset="utf-8">
  <title>on demo</title>
  <style>
    p { color: blue; }
    div { color: red; }
  </style>
  <script src="https://code.jquery.com/jquery-3.7.0.js"> </script>
</head>
<body>
  <p>Click and drag the mouse to select text in the inputs.</p>
  <input type="text" value="Some text">
  <input type="text" value="to test on">
  <div></div>
  <script>
    $(":input").on("select", function () {
      $("div").text("Something was selected").show();
    });
  </script>
</body>
```



2. 이벤트 제거

- 이벤트 연결 제거
 - off() 메소드
 - \$(selector).off()
 - 해당 문서 객체와 관련된 모든 이벤트 제거
 - \$(selector).off(event)
 - 해당 문서 객체의 특정 이벤트와 관련된 모든 리스너 제거
 - \$(selector).off(event, handler)
 - 해당 문서 객체의 특정 이벤트와 관련된 특정 리스너 제거
 - \$(selector).off(event [,selector] [, handler])
 - event: 이벤트 이름
 - selector : 이벤트를 해제할 후손 선택자
 - handler: 특정 이벤트와 관련된 핸들러 제거
 - one() 메소드
 - 이벤트를 한번만 처리하고 해제시키는 메소드

2. 이벤트 제거

- 이벤트 연결 제거
 - \$(selector).off()
 - one() 메소드

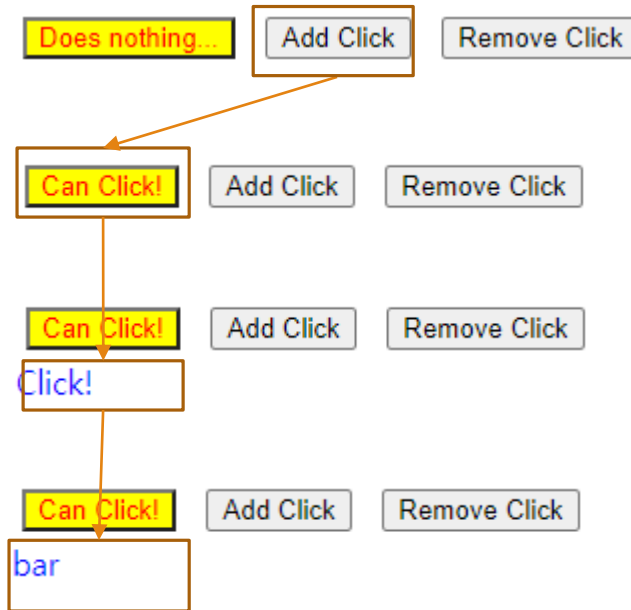
```
<head>
<title>jquery 이벤트 - 이벤트 제거</title>
<script src="https://code.jquery.com/jquery-3.7.0.js"></script>
<script>
$(function () {
    $('h1 a').on('mouseover', function () {
        $(this).css('background-color', 'yellow');

        $('button').on('click', function () { //버튼을 클릭하면 <a>에 연결된 이벤트 제거
            $('h1 a').css('background', 'none');
            $('h1 a').off(); //$('h1 a').off('mouseover');
        });
    });
});
</script>
</head>
<body>
<h1><a href="#">내용</a></h1>
<button>이벤트 삭제</button>
</body>
```

```
//또는 one()를 사용하여 이벤트 제거
$('h1 a').one('mouseover', function () {
    $(this).css('background-color', 'yellow');
});
```

2. 이벤트 제거

- 이벤트 연결 제거 – \$(selector).off()



```
<style>
  button, div{ margin: 5px; }
  button#theone { color: red; background: yellow; }
</style>
<script src="https://code.jquery.com/jquery-3.7.0.js"></script>
<script>
  $(function () {
    function myHandler(event){ $(this).html(event.data.foo); }
    function flash() {
      if ($("#div").css("display") === 'none') { $("#div").show().css("color", "blue"); }
      else { $("#div").hide(); }
    }
    $("#bind").click(function () {
      $("body")
        .on("click", "#theone", flash).find("#theone").text("Can Click!");
    });
    $("#unbind").click(function () {
      $("body")
        .off("click", "#theone", flash).find("#theone").text("Does nothing...");
    });
    $("#div").on("click", {foo:"bar"}, myHandler);
  });
</script>
</head>
<body>
  <button id="theone">Does nothing...</button>
  <button id="bind">Add Click</button>
  <button id="unbind">Remove Click</button>
  <div style="display: none;">Click!</div>
</body>
```


3. 이벤트 사용

- 이벤트 강제 발생
 - trigger() 메소드
 - 이벤트를 강제로 발생시키는 메소드
 - 사용방법
 - \$(selector). trigger(eventName)
 - 특정 이벤트를 발생시킴
 - \$(selector). trigger(eventName, data)
 - 연결된 이벤트 리스너에 data를 전달함
 - 일반적으로 data에는 배열을 넣어 줌

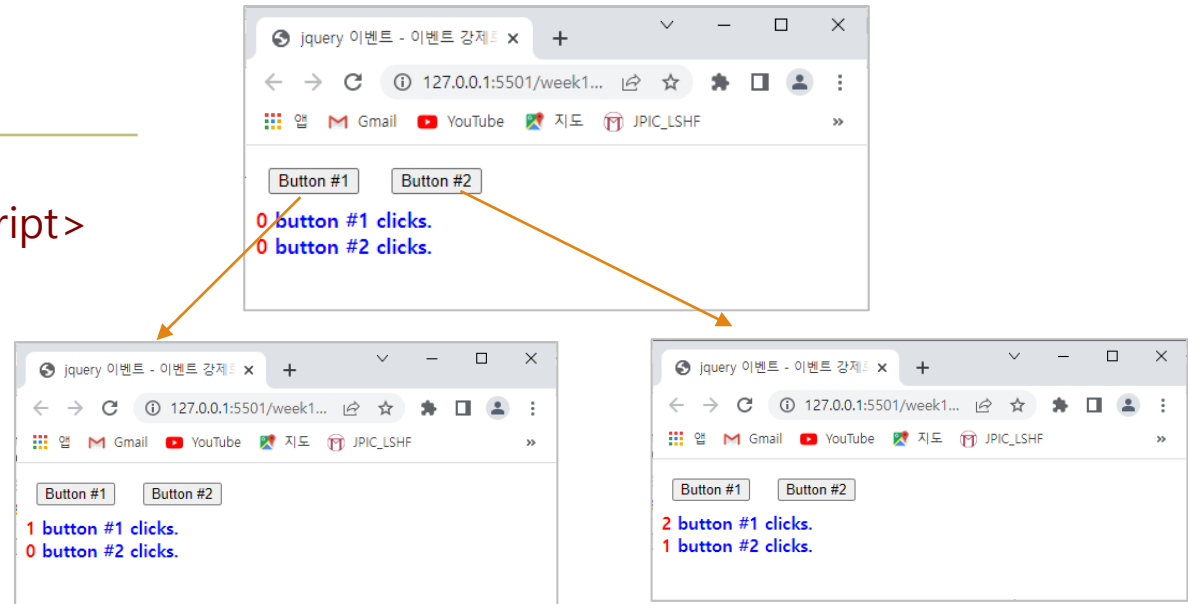
3. 이벤트 사용

- 이벤트 강제 발생 – trigger()

```
<title>jquery 이벤트 - 이벤트 강제로 발생 시키기</title>
<script src="https://code.jquery.com/jquery-3.7.0.js"></script>
<script>
const update = (j) => {
  let n = Number(j.text());
  j.text(n + 1);
};

$(function () {
  $("button").first().click(function () {
    update($("span").first());
  });

  $("button").last().click(function () {
    $("button").first().trigger("click");
    update($("span").last());
  });
});
</script>
```



```
<body>
  <button>Button #1</button>
  <button>Button #2</button>
  <div><span>0</span> button #1 clicks.</div>
  <div><span>0</span> button #2 clicks.</div>
</body>
```

3. 이벤트 사용

- 사용자 정의 이벤트

- 방법1

```
// 1)이벤트 객체 생성
let event = jQuery.Event(eventName);

// 2)이벤트 데이터 생성
event.data1 = value;
event.data2 = value;
.....

// 3) 이벤트 발생
$선택자.trigger(event);
```

- 방법2

```
// 1)이벤트 객체 생성
let event = jQuery.Event(eventName);

// 2) 이벤트 데이터 생성 & 발생
$선택자.trigger(event, [data1=value, ....]);
```

3. 이벤트 사용

- 사용자 정의 이벤트
 - 방법1

```
<body>
  <div>
    <button id="add">추가</button>
  </div>
  <ul class="menu">
    <li>menu1</li>
    <li>menu2</li>
    <li>menu3</li>
    <li>menu4</li>
  </ul>
  <p>추가 정보가 출력되는 영역</p>
  <div id="output"></div>
</body>
```

```
<script>
$(function () {
  let $output = $("#output");
  let $menu = $("ul.menu");
  let count = $menu.children().length; //카운트 변수 추가

  $("#add").click(function () { // #add 버튼에 클릭 이벤트 등록
    count++;
    let $newItem = $("<li>menu" + count + "</li>"); // 신규 메뉴 항목 생성
    $menu.append($newItem); // 메뉴 항목 추가

    //이벤트 객체 생성 - 방법1
    let uevent = jQuery.Event("addItem");
    //이벤트 데이터 생성
    uevent.itemName = $newItem.text();
    uevent.itemLength = $menu.children().length;
    // 이벤트 발생
    $menu.trigger(uevent);
  })

  // addItem 이벤트 리스너 등록
  $menu.on("addItem", function (e) {
    $output.prepend(`<p>추가 아이템 = ${e.itemName}, 아이템 개수 = ${e.itemLength}</p>`);
  })
});
</script>
```

3. 이벤트 사용

- 사용자 정의 이벤트
 - 방법2

추가

- menu1
- menu2
- menu3
- menu4

추가 정보가 출력되는 영역

추가

- menu1
- menu2
- menu3
- menu4
- menu5
- menu6
- menu7

추가 정보가 출력되는 영역

추가 아이템 = menu7, 아이템 개수 = 7
추가 아이템 = menu6, 아이템 개수 = 6
추가 아이템 = menu5, 아이템 개수 = 5

```
<body>
  <div>
    <button id="add">추가</button>
  </div>
  <ul class="menu">
    <li>menu1</li>
    <li>menu2</li>
    <li>menu3</li>
    <li>menu4</li>
  </ul>
  <p>추가 정보가 출력되는 영역</p>
  <div id="output"></div>
</body>
```

```
//이벤트 객체 생성-방법2
let event = jQuery.Event("addItem");
let data = [$newItem.text(), $menu.children().length]; // 이벤트 데이터 생성
$menu.trigger(event, data); // 이벤트 발생
})

// addItem 이벤트 리스너 등록
$menu.on("addItem", function (e, itemName, itemLength) {
  $output.prepend("<p>추가 아이템 = " + itemName + ", 아이템 개수 = " + itemLength + "</p>");
})
```

4. 시각 효과

- 시각 효과 메서드
 - 간단한 효과 메서드 제공

jQuery 라이브러리의 시각 효과 메서드

메서드	설명
show(속도)	문서 객체가 커지며 표시
hide(속도)	문서 객체가 작아지며 사라짐
toggle(속도)	show() 메서드와 hide() 메서드를 번갈아 실행
slideDown(속도)	문서 객체가 슬라이드 효과와 함께 표시
slideUp(속도)	문서 객체가 슬라이드 효과와 함께 사라짐
slideToggle(속도)	slideDown() 메서드와 slideUp() 메서드를 번갈아 실행
fadeIn(속도)	문서 객체가 선명해지며 표시
fadeOut(속도)	문서 객체가 흐려지며 사라짐
fadeToggle(속도)	fadeIn() 메서드와 fadeOut() 메서드를 번갈아 실행

- 속도를 매개변수로 입력
- 속도를 입력하지 않으면 기본 값인 500밀리초가 적용

속도 문자열

속도 문자열	설명
slow	600밀리초 속도
normal	500밀리초 속도
fast	400밀리초 속도

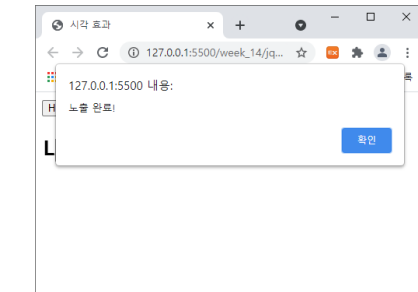
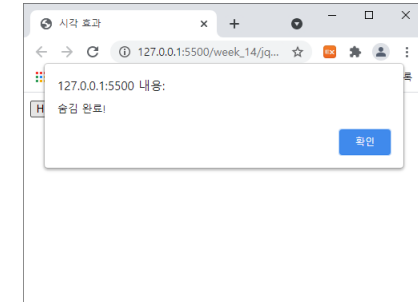
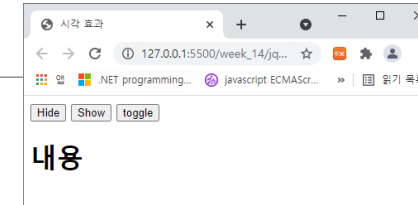
- \$(selector).메소드([duration] [, complete])
- \$(selector).메소드(duration [,easing] [, complete])
- \$(selector).메소드(options)
 - duration – 효과가 지속되는 시간
 - easing – 이징 함수 이름
 - complete – 효과 완료 후 호출 할 콜백 함수
 - options – duration, easing, complete 매개변수를 하나의 객체로 묶어 매개변수로 전달

4. 시각 효과

- 숨기기, 나타내기 – hide(), show(), toggle()

```
<head>
<title>시각 효과</title>
<script src="https://code.jquery.com/jquery-3.7.0.js"> </script>
<script>
$(function () {
    $("#btn1").on("click", function () { // "Hide" 버튼을 클릭했을 때
        // 1초 동안에 <h1>을 숨기고, 콜백 함수 실행
        $("#h1").hide(1000, function () { alert("숨김 완료!"); });
    });
    $("#btn2").on("click", function () { // "Show" 버튼을 클릭했을 때
        // 1초 동안에 <h1>을 노출시키고, 콜백 함수 실행
        $("#h1").show("slow", function () { alert("노출 완료!"); });
    });
    $("#btn3").on("click", function () { $("#h1").toggle("normal"); });
});
</script>
</head>
```

```
<body>
<button id="btn1">Hide</button>
<button id="btn2">Show</button>
<button id="btn3">toggle</button>
<h1>내용</h1>
</body>
```

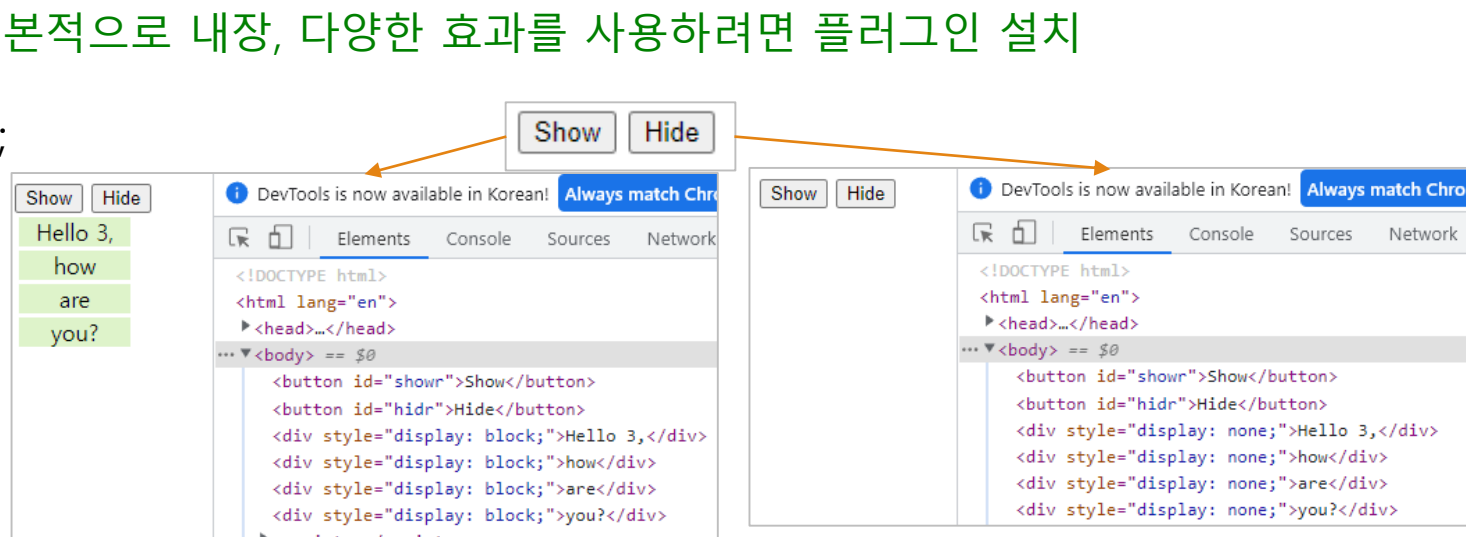


4. 시각 효과

- 숨기기, 나타내기 – hide(), show()

```
<body>
  <button id="showr">Show</button>
  <button id="hidr">Hide</button>
  <div>Hello 3,</div>
  <div>how</div>
  <div>are</div>
  <div>you?</div>
  <script>
    $("#showr").click(function () {
      $("#div").first().show({
        duration: "slow",
        easing: 'linear', // 'swing', 'linear' 함수가 기본적으로 내장, 다양한 효과를 사용하려면 플러그인 설치
        done: function showNext() {
          $(this).next("div").show("fast", showNext);
        }
      });
    });
    $("#hidr").click(function () {
      $("#div").hide(1000);
    });
  </script>
</body>
```

```
<head>
  <meta charset="utf-8">
  <title>show demo</title>
  <style>
    div {
      margin: 3px;
      width: 80px;
      display: none;
      text-align: center;
    }
  </style>
  <script src="https://code.jquery.com/jquery-3.6.0.js"> </script>
</head>
```



4. 시각 효과

- 투명도 적용, 사라지기, 나타나기 – fadeTo(), fadeIn(), fadeout(), fadeToggle()

```
<script src="https://code.jquery.com/jquery-3.6.0.js" > </script>
```

```
<script>
```

```
$(document).ready(function () {
```

```
  $("#btn1").on("click", function () {
```

```
    $("#h1").fadeTo("slow", Math.random()); //<h1>이 생성된 난수 값 만큼 노출, 투명도 적용
```

```
  });
```

```
  $("#btn2").on("click", function () {
```

```
    $("#h1").fadeIn("slow"); //<h1>이 서서히 표시
```

```
  });
```

```
  $("#btn3").on("click", function () {
```

```
    $("#h1").fadeOut("slow"); //<h1>이 투명해 지면서 사라짐
```

```
  });
```

```
  $("#btn4").on("click", function () {
```

```
    $("#h1").fadeToggle("slow"); //<h1>이 투명해 지면서 사라짐
```

```
  });
```

```
});
```

```
</script>
```

```
<body>
```

```
  <button id="btn1">투명도(Math.random())</button>
```

```
  <button id="btn2">fadeIn</button>
```

```
  <button id="btn3">fadeOut</button>
```

```
  <button id="btn4">fadeToggle</button>
```

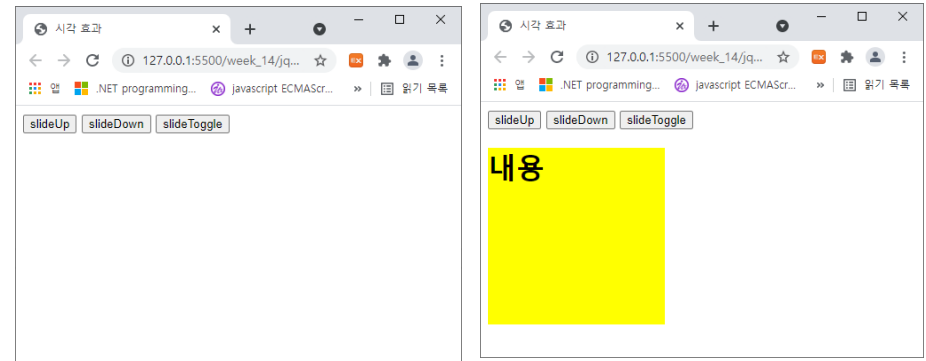
```
  <h1>내용</h1>
```

```
</body>
```

4. 시각 효과

- 접고 펼치기 – `slideUp()`, `slideDown()`, `slideToggle()`

```
<head>
  <title>시각 효과</title>
  <script src="https://code.jquery.com/jquery-3.6.0.js"> </script>
  <script>
    $(document).ready(function () {
      $("#btn1").on("click", function () {
        $("h1").slideUp(1000);
      });
      $("#btn2").on("click", function () {
        $("h1").slideDown("slow");
      });
      $("#btn3").on("click", function () {
        $("h1").slideToggle("fast");
      });
    });
  </script>
  <style type="text/css">
    h1 {
      width: 200px;      height: 200px;      background-color: #FF0;
    }
  </style>
</head>
```



```
<body>
  <button id="btn1">slideUp</button>
  <button id="btn2">slideDown</button>
  <button id="btn3">slideToggle</button>
  <h1>내용</h1>
</body>
```

5. 애니메이션

- 애니메이션
 - animate() 메서드
 - 세 가지 형태로 사용

❶ \$(selector).animate(속성 객체);

❷ \$(selector).animate(속성 객체, 시간);

❸ \$(selector).animate(속성 객체, 시간, 콜백 함수);

```
<head>
  <title>애니메이션</title>
  <script src="https://code.jquery.com/jquery-3.7.0.js"> </script>
  <script>
    $(document).ready(function () {
      // 스타일을 변경합니다.
      $('#box').css({
        width: 100,
        height: 100,
        background: 'red'
      }).animate({
        width: 300,
        opacity: 0.5,
      }, 500);
    });
  </script>
</head>
<body>
  <div id="box"> </div>
</body>
```

5. 애니메이션

```
<head>
<title>애니메이션</title>
<script src="https://code.jquery.com/jquery-3.7.0.js"></script>
<script>
$(function () {
    $("h1").animate({
        marginLeft: "250px"
    }, 5000, function () { alert("도착완료!"); });

    $("h2").animate({
        marginLeft: "250px",
        opacity: 0.3,
        width: "100px"
    }, 5000);

    $("h3").animate({ marginLeft: "250px" }, 3000)
        .animate({ marginLeft: "100px" }, 2000);
});
</script>
```

```
<style>
h1 {
    width: 50px;
    height: 30px;
    background-color: red;
    font-size: 100%;
}
h2 {
    width: 50px;
    height: 30px;
    background-color: orange;
    font-size: 100%;
}
h3 {
    width: 50px;
    height: 30px;
    background-color: aqua;
    font-size: 100%;
}
</style>
</head>
<body>
    <h1>내용</h1>
    <h2>내용</h2>
    <h3>내용</h3>
</body>
```

5. 애니메이션

- 애니메이션 효과 제어
 - ❶ delay() 메서드를 animate() 메서드 사이에 입력
 - 애니메이션을 특정시간만큼 정지

```
❶ $('h1').animate({ }).delay(1000).animate({ });
```

- ❷의 stop() 메서드
 - 현재 실행 중인 애니메이션 중지

```
❷ $('h1').stop();
```

5. 애니메이션

```
<head>
<title>애니메이션</title>
<script src="https://code.jquery.com/jquery-3.7.0.js"></script>
<script>
$(function () {
    $('.txt1').animate({ marginLeft: "300px" }, 1000);
    $('.txt2').delay(3000).animate({ marginLeft: "300px" }, 1000);
    $('.btn1').on('click', moveMent);

    function moveMent() {
        $('.txt3').animate({ marginLeft: "+=50px" }, 800);
        $('.txt4').animate({ marginLeft: "+=50px" }, 800);
        $('.txt4').stop();
        $('.txt5').animate({ marginLeft: "+=50px" }, 800);
    }
});
</script>
```

```
<style type="text/css">
p {
    width: 110px;
    text-align: center;
}
.txt1 { background-color: aqua; }
.txt2 { background-color: blue; }
.txt3 { background-color: crimson; }
.txt4 { background-color: darkmagenta; }
.txt5 { background-color: gold; }
</style>
</head>
<body>
<p class="txt1">effect1 </p>
<p class="txt2">effect2<br>delay(3000)</p>
<p><button class="btn1">50px전진</button></p>
<p class="txt3">effect3</p>
<p class="txt4">effect4<br>stop()/p>
<p class="txt5">effect5</p>
</body>
```

Q & A

- jQuery 이벤트와 효과에 대한 학습이 모두 끝났습니다.
- 모든 내용을 이해 하셨나요?
- 아직 이해가 안되는 내용이 있다면 다시 한번 복습하시기 바랍니다.
- 질문은 한림 SmartLEAD 쪽지 또는 e-mail 또는 전화상담을 이용하시기 바랍니다.
- 한 학기 수고하셨습니다.^^