



Day14; 220922

📅 날짜	
📁 유형	
🏷️ 태그	

GitHub - u8yes/Web

You can't perform that action at this time. You signed in with another tab or window. You signed out in another tab or window. Reload to refresh

<https://github.com/u8yes/Web>

u8yes/Web

1 Contributor 0 Issues 1 Star 0 Forks

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/7222a5c9-cbc3-4648-85b1-bb10b2e09f27/img.zip>

다른 곳이 체크되면서 other는 체크가 안되게 disabled 만드는 것.

.요소:checked
.요소:disabled
요소.enabled

Text file encoding

☒ Default (MS949)

☐ Other:

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/f9743fba-7100-4301-8d10-3639a792cd84/webapp.zip>

01_jQueryBasic.html

```
<!-- jQuery 기본
1. jQuery 개요
- 모든 브라우저에서 동작하는 클라이언트 자바스크립트 라이브러리.
- 다음의 기능을 위해 제작
  . DOM과 관련된 처리를 쉽게 구현
  . 일관된 이벤트 연결을 쉽게 구현
  . 시각적 효과를 쉽게 구현
  . ajax 애플리케이션을 쉽게 구현

- https://jquery.com
- jQuery 사용 방법
  1) 직접 다운로드 사용하는 방법.
  2) CDN 호스트 사용하는 방법.
    . Content Delivery Network
    . 사용자에게 간편하게 콘텐츠를 제공하는 방식을 의미.
    . 구글, 마이크로소프트, jQuery측에서 사용자가 jQuery를 사용하기 편하게 콘텐츠를 제공.
-->

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>jQuery 태그 구조</title>

<style type="text/css">
  .high-light-0{ background:yellow; }
  .high-light-1{ background:orange; }
  .high-light-2{ background:gold; }
  .high-light-3{ background:green; }
  .high-light-4{ background:red; }

</style>

<!-- 방법1. 직접 다운로드, src=""는 외부로 import해준다.
기능을 사용하려면 반드시 js 밑으로 실행할 함수, 예제들을 넣어야 한다. -->
<script src="./js/jquery-3.6.1.min.js"></script>

<!-- 방법2. CDN 방식, 네트워크로 직접 접근해서 다운로드 받음.(인터넷 연결 필요)
<script src="https://code.jquery.com/jquery-3.6.1.min.js"></script>
-->
<script type="text/javascript">
$(document).ready(function(){
/* 1. 라이브러리 포함 방법 : 2가지 방식
- 방법1, 방법2 참조
*/

/* 2. jQuery를 사용한 웹 페이지 형식
- $(document).ready(function(){});
  . 문서가 준비 완료되면 매개변수로 전달된 함수를 실행하라는 의미.
  . $(function(){}); // 간단한 형식 정의.

  alert('First Ready');
*/

/* 3. jQuery 기본 선택자
var headers = document.getElementsByTagName('h1');
```

```

headers[0].innerHTML = '더조은';
headers[1].innerHTML = '컴퓨터 학원';

1) 전체 선택자(*)
$('*').css('color', 'red');

2) tag 선택자
- 한 개 태그 선택
  $('p').css('color', 'blue');
- 두 개 이상의 태그 선택
  $('h1, p').css('color', 'blue');

3) id 선택자
$('#target').css('color', 'blue');

4) class 선택자
$('.item').css('color', 'green');
$('h1.select').css('color', 'red');
$('.item.select').css('background', 'blue');
*/

/* 4. jQuery 자식(손) 선택자와 후손 선택자
- 자식(손) 선택자
  $('body > div').css('color', 'blue');
- 후손 선택자
  $('body div').css('color', 'blue');
*/

/* 5. jQuery 속성 선택자
- 기본 선택자 뒤에 붙여 사용.
- 입력 양식과 관련된 태그를 선택할 때 많이 사용.

  $('input[type=text]').val('Hello jQuery..!');
*/

/* 6. jQuery 필터 선택자
1) 입력 양식 필터 선택자
//   .input:button
//   .input:checkbox
//   .input:radio
//   .input:file
//   .input:image
//   .input:password
//   .input:text
//   .input:submit
//   .input:reset

//   .요소:checked
//   .요소:disabled
//   .요소:enabled
//   .요소:focus
//   .요소:selected

// 변수를 넣어주면 주소값이 있어야 할바엔 그냥 function을 안에다 넣어줌.
// 두번째 숫자는 2000 밀리세컨드 = 2초 지나면 한번 함수 호출
// $()는 제이쿼리 함수, .val()는 메서드로 호출하면서 기능 사용
setTimeout(function(){
  var value = $('select > option:selected').val();
  alert(value);
}, 2000);
*/

/*
// 2) 위치 필터 선택자
//   . 선택자를 사용해 문서 객체를 선택한 후에 다음과 같은 위치를 지정한다.
//   >> 요소:odd -> 홀수 번째에 위치한 문서 객체 선택.

```

```

//      >> 요소:even -> 짝수 번째에 위치한 문서 객체 선택.
//      >> 요소:first -> 첫번째에 위치한 문서 객체 선택.
//      >> 요소:last -> 마지막에 위치한 문서 객체 선택.

$('tr:odd').css('background', 'silver'); // index 0부터 카운팅, 홀수로 적용
$('tr:even').css('background', '#9F9F9F');
$('tr:first').css('background', '#000000').css('color', 'FFFFFF');
$('tr:last').css('background', 'yellow').css('color', 'red');
*/

/*
// 3) 함수 필터 선택자
// >> 요소:contains(문자열)
//      ->특정 문자열을 포함하는 문서 객체를 선택.
// >> 요소:eq(n)
//      ->n번째에 위치하는 문서 객체를 선택.
// >> 요소:gt(n)
//      ->n번째 초과하는 위치에 있는 문서 객체를 선택.
// >> 요소:has(h1)
//      ->h1 태그를 가지고 있는 문서 객체를 선택.
// >> 요소:lt(n)
//      ->n번째 미만에 위치하는 문서 객체를 선택.
// >> 요소:not(선택자)
//      ->선택자와 일치하지 않는 문서 객체를 선택.
// >> 요소:nth-child(3n+1) (단, n = 0, 1, ...)
//      -> 3n+1번째에 위치하는 문서 객체를 선택(인덱스 1부터 카운팅).

$('tr:eq(0)').css('background', 'black').css('color', 'white');
$('tr:nth-child(3n+1)').css('background', 'green');
$('tr:nth-child(3n+2)').css('background', 'red');
$('tr:nth-child(3n)').css('background', 'blue');
*/

/* 7. jQuery를 사용한 배열 관리(1)
// - each() 메서드 사용
//      . 매개 변수로 입력한 함수 사용.
//      . for in 반복문처럼 객체나 배열의 요소 검사하는 메서드

// - each() 메서드의 형태
//      1. $.each(object, function(index, item){});
// 자바스크립트는 배열이 [], 자바의 {}와 같다. {}는 하나의 객체
// each는 배열을 담는 기능, index는 0,1,2,3... 객체를 세는 숫자

var array = [{name:'Naver', link:'https://www.naver.com'},
             {name:'Daum', link:'https://www.daum.net'},
             {name:'Nate', link:'https://www.nate.com'}];

$.each(array, function(index, item){
    var output = '';

    output += '<a href="'+item.link+'" >';
    output += '<h1>' + item.name + '</h1>';
    output += '</a>';

    document.body.innerHTML += output;
});// innerHTML 내용 보여줘라

*/

/* 8. jQuery를 사용한 배열 관리(2)
// - 각 객체에 서로 다른 class 속성을 설정하고 싶을 때 each() 메서드 사용.
// 예제 1.

```

```

// .class="high-light" 만들어줌
$('h1').addClass('high-light');

// 예제2
// - each() 메서드의 형태
// 2. $(selector).each(function(index, item){});

$('h1').each(function(index, item){
    $(item).addClass('high-light-'+index);
    //$(this).addClass('high-light-'+index);
});

// 예제3 : addClass() 메서드의 매개변수에 함수 전달도 가능.

$('h1').addClass(function(index){
    return 'high-light-'+index;
});

*/

/* 9. jQuery를 사용한 객체 확장
// - 자바스크립트에서의 객체 속성 추가

var object = {};

object.name = '홍길동';
object.gender = '남자';
object.part = '영업';

// - jQuery에서의 객체 속성 추가
//   . extend() 메서드

var object = {name:'홍길동'};

$.extend(object, {
    gender:'남자',
    part:'영업'
});

var output = '';
$.each(object, function(index, item){
    output += index + ' : ' + item+'\n';
});

alert(output);

// */

/* 10. jQuery 라이브러리 충돌 방지
// - jQuery와 prototype 자바스크립트 라이브러리도 동일하게 $를
//   식별자로 사용.
// - 따라서 두 라이브러리를 함께 사용시 충돌 발생.

// 더 이상 식별자 $를 사용할 수 없다.
$.noConflict();

var J = jQuery;

```

```

        J(document).ready(function(){
            J('h1').addClass('high-light');
        });
    // */

});
</script>

</head>
<body>

<!-- 3-1/3-2. 전체/태그 선택자
<h1>Lorem ipsum</h1>
<p>Lorem ipsum dolor sit amet.</p>
<h1>Lorem ipsum</h1>
<p>Lorem ipsum dolor sit amet.</p>
<hr>
-->

<!-- 3-3. 아이디 선택자
<h1>Header-0</h1>
<h1 id="target">Header-1</h1>
<h1>Header-2</h1>
<hr>
-->

<!-- 3-4. 클래스 선택자
<h1 class="item">Header-0</h1>
<h1 class="item select">Header-1</h1>
<h1 class="">Header-2</h1>
<h2 class="select">Header-3</h2>
<hr>
-->

<!-- 4. 자식(손) 선택자와 후손 선택자
<div>
    <ul>
        <li>Apple</li>
        <li>Bag</li>
        <li>Cat</li>
        <li>Dog</li>
    </ul>
</div>
<hr>
-->

<!-- 5. 속성 선택자
<input type="text"/><br>
<input type="password"/><br>
<input type="radio"/><br>
<input type="checkbox"/><br>
<input type="file"/><br>
<hr>
-->

<!-- 6-1. 입력 양식 필터 선택자
<select>
    <option>Apple</option>
    <option>Bag</option>
    <option>Cat</option>
    <option>Dog</option>
</select>
<hr>
-->

<!-- 6-2/6-3. 필터선택자
<table border="1">

```

```

<tr><th>이름</th><th>혈액형</th><th>지역</th></tr>
<tr><td>홍길동</td><td>AB형</td><td>서울시 강남구</td></tr>
<tr><td>홍길서</td><td>A형</td><td>서울시 마포구</td></tr>
<tr><td>홍길남</td><td>B형</td><td>서울시 서초구</td></tr>
<tr><td>홍길북</td><td>O형</td><td>서울시 송파구</td></tr>
<tr><td>김길동</td><td>O형</td><td>서울시 영등포구</td></tr>
<tr><td>이길동</td><td>A형</td><td>서울시 강서구</td></tr>
<tr><td>박길동</td><td>B형</td><td>서울시 용산구</td></tr>
<tr><td>홍길동</td><td>AB형</td><td>서울시 강남구</td></tr>
<tr><td>홍길서</td><td>A형</td><td>서울시 마포구</td></tr>
<tr><td>홍길남</td><td>B형</td><td>서울시 서초구</td></tr>
<tr><td>홍길북</td><td>O형</td><td>서울시 송파구</td></tr>
<tr><td>김길동</td><td>O형</td><td>서울시 영등포구</td></tr>
<tr><td>이길동</td><td>A형</td><td>서울시 강서구</td></tr>
<tr><td>박길동</td><td>B형</td><td>서울시 용산구</td></tr>
</table>
<hr>
-->

<!-- 8. jQuery를 사용한 배열 관리(2)
<h1>item - 0</h1>
<h1>item - 1</h1>
<h1>item - 2</h1>
<h1>item - 3</h1>
<h1>item - 4</h1>
-->

</body>
</html>

```

02_DOM_Search.html

```

<!--
문서 객체 탐색 : 선택자와 관련된 추가 메서드를 살펴보자.
-->
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>문서 객체 탐색</title>

<script src="./js/jquery-3.6.1.min.js"></script>
<script type="text/javascript">

/* 1. 기본 필터링 메서드
// - filter() 메서드 사용 두 가지 형태
// 1) $(selector).filter(selector);
// 2) $(selector).filter(function(index){});

// 예제: 홀수번째 속성 변경.
$(function(){
    $('h3:even').css({
        backgroundColor: 'black',
        color: 'white'
    });
});

// filter() 메서드 적용 형태1.
$(function(){

```

```

        $('h3').filter(':even').css({
            backgroundColor:'black',
            color:'white'
        });
    });

    // filter() 메서드 적용 형태2. // index는 var 변수
    $(document).ready(function(){
        $('h3').filter(function(index){
            return index % 3 == 0; // return true 반환되면 선택하겠다.
        }).css({
            background:'black',
            color:'white'
        });
    });

    });
    // */

/* 2. 문서 객체 탐색 종료
// 예제1.
// 체이닝(연속적으로 선택)을 사용해 한 줄로 서로 다른 문서 객체에 스타일 적용 가능

$(function(){
    $('h3').css('background', 'orange').filter(':even').css('color', 'red');
});

// 예제2.
$(function(){
    $('h3').css('background', 'orange');
    $('h3:even').css('color', 'white');
    $('h3:odd').css('color', 'red');
});

// 예제3.
// 체이닝을 이용한 예제2의 표현
// end() : 문서 객체 선택을 한 단계 뒤로 돌림.
$(function(){
    // $('h3').css('background', 'orange').filter(':even').css('color',
    //     'white').filter(':odd').css('color', 'red');
    $('h3').css('background', 'orange').filter(':even').css('color',
        'white').end().filter(':odd').css('color', 'red');
});
// */
/*
3. 특정 위치의 문서 객체 선택
// - eq(n) : n번째 위치에 존재하는 문서 객체를 선택.
// - first() : 첫번째 위치하는 문서 객체를 선택.
// - last() : 마지막에 위치하는 문서 객체를 선택.

$(function(){
    $('h3').eq(3).css('background', 'orange');
    $('h3').eq(-2).css('background', 'red'); // 뒤에서 2번째를 말한다.
    $('h3').first().css('background', 'green');
    $('h3').last().css('background', 'blue');
});

// */
/* 4. 문서 객체 추가 선택
// - add() : 기존 문서 객체를 포함하여 추가로 문서 객체를 선택.

$(function(){
    $('h1').css('background', 'gray').add('h2').css('color', 'red'); // h2에서 add처리를 해도 h1도 포함된다.

```



```

});

// */
/* 5. 문서 객체의 특징 판별
// - is() : 문서 객체의 특징 판별.
// */
$(function(){
    $('h1').each(function(){
        if($(this).is('.select')){
            $(this).css('color', 'red');
        }
    });
});

</script>

</head>
<body>
<!--
    <h3>Header - 0</h3>
    <h3>Header - 1</h3>
    <h3>Header - 2</h3>
    <h3>Header - 3</h3>
    <h3>Header - 4</h3>
    <h3>Header - 5</h3>
-->

<!-- 4. 문서 객체 추가 선택 예시
    <h1>Header - 0</h1>
    <h2>Header - 1</h2>
    <h1>Header - 2</h1>
    <h2>Header - 3</h2>
    <h1>Header - 4</h1>
-->

<!-- 5. 문서 객체의 특징 판별 -->
    <h1 class="select">Header - 0</h1>
    <h1>Header - 1</h1>
    <h1 class="select">Header - 2</h1>

</body>
</html>

```

자바스크립트는 var 타입이 여러가지 있기 때문에 동적 프로그래밍 언어.

자바는 int는 int, String은 String... 등등등 정적 프로그래밍 언어다.

03_DOM_Operation.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>문서 객체 조작</title>

<style type="text/css">
    .first {color:red;}
    .second {color:pink;}
    .third {color:orange;}
</style>

<script src="./js/jquery-3.6.1.min.js"></script>
<script type="text/javascript">
/* 1. 문서 객체의 클래스 속성 추가
// - addClass() : 문서 객체의 클래스 속성을 추가.

    // 예제1. 매개변수에 문자열 입력.
    $(document).ready(function(){
        $('h1').addClass('item');
    });

    // 예제2. 매개변수에 함수 입력.
    $(document).ready(function(){
        $('h1').addClass(function(index){
            return 'item_'+index;
        });
    });
// */

/* 2. 문서 객체의 클래스 속성 제거
// - removeClass(): 문서 객체의 클래스 속성을 제거.
// - 여러 개의 클래스 속성 중 선택적으로 제거.

    $(function(){
        $('h1').removeClass('item');
    });
// */

/* 3. 문서 객체의 속성 검사
// - attr(): 속성과 관련된 모든 기능 수행.
//          첫번째 문서 객체의 속성 출력.

    $(function(){
        var src = $('img').attr('src');

        alert(src);
    });

// */
/* 4. 문서 객체의 속성 추가
// - attr() 메서드 사용.
// - 3가지 형태로 사용.

// 1) $(selector).attr(name, value);
    $(function(){
        $('img').attr('width', '200');
    });

// 2) $(selector).attr(name, function(index){});
    $(function(){
```

```

        $('img').attr('width', function(index){
            return (index + 1) * 100;
        });
    });

// 3) $(selector).attr(object);
$(function(){
    $('img').attr({
        width:function(index){
            return (index+1)*100;
        }, height:100
    });
});
// */

/* 5. 문서 객체의 속성 제거
// - removeAttr(name):문서 객체의 속성을 제거해 주는 메서드.

$(function(){
    $('h1').removeAttr('data-index');
});

// */

/* 6. 문서 객체의 스타일 검사
// - css(): 스타일과 관련된 모든 기능을 수행.

$(function(){
    var color = $('h1').css('color');
    alert(color);
});
// */

/* 7. 문서 객체의 스타일 추가
// - 사용형태 3가지

// 1) $(selector).css(name, value);

$(function(){
    $('h1').css('color', 'red');
});

// 2) $(selector).css(name, function(index) { });
// . 선택자로 선택한 문서 객체에 개별적으로 스타일 속성을 적용하고 싶을 때 사용하는 방법.

$(function(){
    var color = ['red', 'green', 'blue'];

    $('h1').css('color', function(index){
        return color[index];
    });
});

// 3) $(selector).css(object);

$(function(){
    var color = ['red', 'green', 'blue'];

    $('h1').css({
        color:function(index){ return color[index]; },
        background:'black'
    });
});
// */

/* 8. 문서 객체의 내부 검사
// - 기존 자바스크립트에서 문서 객체의 innerHTML 속성과 관련된 jQuery 메서드.
// - 1) html(): 문서 객체 내부의 글자와 관련된 모든 기능을 수행.

```

```

$(function(){
    var html = $('h1').html();

    alert(html);
});

//- 2) text():
//   . 문서 객체 내부의 글자와 관련된 모든 기능을 수행.
//   . 선택자로 선택한 모든 문서 객체의 글자를 가져와 출력.

$(function(){
    var text = $('h1').text();

    alert(text);
});
// */

/* 9. 문서 객체의 내부 추가
// 1) $(selector).html(value);
//     -> HTML 태그를 인식해 문자열 삽입.

$(function(){
    $('div').html('<h1>${}.html(</h1>');
});

//   $(selector).text(value);
//     -> HTML 태그를 인식하지 않음.

$(function(){
    $('div').text('<h1>${}.text(</h1>');
});

// 2) $(selector).html(function(index, html){});
//     $(selector).text(function(index, html){});

$(function(){
    $('div').html(function(index){
        return '<h1>Header - '+index+'</h1>';
    });
});

$(function(){
    $('h1').html(function(index, html){
        return '* '+html+' *';
    });
});
// */

/* 10. 문서 객체의 내부 제거
// 1) remove(): 특정 문서 객체를 제거.
// 2) empty() : 특정 문서 객체의 후손을 모두 제거.

$(function(){
    $('h1').first().remove();
});

$(function(){
    $('div').empty();
});
// */

/* 11. 문서 객체의 생성(1)
// - 기존 자바스크립트로 구현 방법.
// <h2>Hello DOM</h2>
//   1. 문서 객체 생성
//

```

```

$(function(){
    var header = document.createElement('h1');
    var textNode = document.createTextNode("Hello DOM");

    // 2. 노드 (요소 / 텍스트)를 연결.
    header.appendChild(textNode);

    // 3. 바디 문서 객체에 header 문서 객체를 추가.
    document.body.appendChild(header);
});

// 텍스트 노드를 갖는 문서 객체를 생성하는 방법.
// $(): 문서 객체 생성.
// $() 함수의 매개 변수에 HTML 태그를 문자열로 넣기만 하면 문서 객체가 생성됨.

// 생성한 문서 객체를 body 태그에 추가하는 방법1.
$(document).ready(function(){});

$(function(){
    $('<h1></h1>').html('Hello jQuery...!').appendTo('body');
});

// $() 메서드의 내부에 HTML 태그를 바로 넣어 body 태그에 추가하는 방법2

$(function(){
    $('<h1>Hello jQuery...!</h1>').appendTo('body');
});
*/

/* 12. 문서 객체의 생성(2)
// 텍스트 노드를 갖지 않는 문서 객체를 생성하는 방법.

$(function(){
    $('<img />').attr('src', 'img/Tulips.jpg').appendTo('body');
});

$(function(){
    $('<img />').attr({
        src:'img/Tulips.jpg',
        width:'350',
        height:'250'
    }).appendTo('body');
});
//*/

/* 13. 문서 객체의 삽입(1) */
// $(A).appendTo(B) : A를 B의 뒷 부분(자식 요소)에 추가한다.
// $(A).prependTo(B) : A를 B의 앞 부분(자식 요소)에 추가한다.
// $(A).insertAfter(B) : A를 B의 뒤(형제요소)에 추가한다.
// $(A).insertBefore(B) : A를 B의 앞(형제요소)에 추가한다.

/* 14. 문서 객체의 삽입(2)
// $(A).append(B) : B를 A의 뒷 부분(자식 요소)에 추가한다.
// $(A).prepend(B) : B를 A의 앞 부분에 추가한다.
// $(A).after(B) : B를 A의 뒤에 추가한다.
// $(A).before(B) : B를 A의 앞에 추가한다.

// 모든 메서드 사용 방법 동일
// 1. $(selector).append(content, content, ..... , content)

$(function(){
    var h1 = '<h1>Header 1</h1>';
    var h2 = '<h2>Header 2</h2>';

```

```

        $('body').append(h1, h2, h1, h2);
    });

    //- 2. $(selector).append(function(index, value){});

    $(function(){
        var content = ['홍길동', '김길동', '박길동'];

        $.each(content, function(index, value){
            $('#name_region').append($('<li>' + value + '</li>'));
        });
    });

    //- 3.
    $(function(){
        // $('header').after($('<div>이길동</div>'));
        $('<div>이길동</div>').insertAfter($('header'));
    });
    // */

    /* 15. 문서 객체의 이동
    //- 기존에 존재하는 문서 객체를 선택하고 문서 객체 삽입 메서드를 사용하면,
    // 문서 객체를 쉽게 다른 곳으로 이동시킬 수 있다.

    $(function(){
        $('img').css('width', '250');

        setInterval(function(){
            $('img').first().appendTo('body');
        }, 2000);
    }); // appendTo()는 body 뒤에 붙여라
    // */

    /* 16. 문서 객체의 복제
    //- 기존 문서 객체를 선택하고 문서 객체 삽입 메서드를 사용하면, 문서 객체가 이동.
    // append는 div 뒤에 h1을 자식으로 넣어라

    $(function(){
        $('div').append($('h1'));
    });

    //=====
    //- clone() : 문서 객체 복제.
    //- $(selector).clone()
    // $(selector).clone(Boolean dataAndEvents) : true -> 이벤트까지 복제하는 deep copy
    // $(selector).clone(Boolean dataAndEvents, Boolean deepDataAndEvents) : child까지의 복제 여부.

    $(function(){
        $('div').append($('h1').clone());
    });
    // */

</script>
</head>

<body>

<!-- 1. 문서 객체의 클래스 속성 추가
    <h1>Header-0</h1>
    <h1>Header-1</h1>
    <h1>Header-2</h1>
-->

<!-- 2. 문서 객체의 클래스 속성 제거
    <h1 class="item">Header-0</h1>
    <h1 class="item select">Header-1</h1>
    <h1 class="item">Header-2</h1>
-->

```

```

-->

<!-- 3/4. 문서 객체의 속성 검사/추가
  
  
  
-->

<!-- 5. 문서 객체의 속성 제거, 임의로 작성자가 data-index라는 이름을 붙인 속성
  <h1 data-index="0">Header-0</h1>
  <h1 data-index="1">Header-1</h1>
  <h1 data-index="2">Header-2</h1>
-->

<!-- 6. 문서 객체의 스타일 검사
  <h1 class="first">Header-0</h1>
  <h1 class="second">Header-1</h1>
  <h1 class="third">Header-2</h1>
-->

<!-- 7/8. 문서 객체의 스타일 추가
  <h1>Header-0</h1>
  <h1>Header-1</h1>
  <h1>Header-2</h1>
-->

<!-- 9-1. 문서 객체의 내부 추가
  <div></div>
  <div></div>
  <div></div>
-->

<!-- 9-2. 문서 객체의 내부 추가
  <h1>Header-0</h1>
  <h1>Header-1</h1>
  <h1>Header-2</h1>
-->

<!-- 10. 문서 객체의 내부 제거
  <div>
    <h1>Header-0</h1>
    <h1>Header-1</h1>
  </div>
-->

<!-- 14. 문서 객체의 삽입(2)
  <div>
    <b>데이터</b>
    <ul id="name_region"></ul>
  </div>
-->

<!-- 14. 문서 객체의 삽입(3)
  <header>
    <h1>데이터</h1>
  </header>
-->

<!-- 15. 문서 객체의 이동
  
  
  
  
  
  
-->

```

```

<!-- 16. 문서 객체의 복제
    <h1>HEADER</h1>
    <hr />
    <div></div>
    <hr />

-->
</body>
</html>

```

04_Events.htm

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Events</title>

<style type="text/css">
/* 1-2) / 2. 이벤트 연결 */
    .reverse {background:Black; color:White;}

/* 4.매개변수 Context */
    * { margin:0px; padding:0px }

    div {
        margin:5px; padding:3px;
        border:3px solid black;
        border-radius:10px;
    }

/* 8. 마우스 이벤트 */
    .outer {
        width:200px; height:200px;
        background:Orange; padding:50px;
        margin:10px;
    }
    .inner {
        width:100%; height:100%;
        background:Pink;
    }

</style>

<script src="./js/jquery-3.6.1.min.js"></script>
<script type="text/javascript">

/* 1. 이벤트 연결1
// - on()
// - 사용 형태
//   1) $(selector).on(eventName, function(event){ });
//
$(function(){
    // 이벤트 연결
    $('h1').on('click', function(event){
        $(this).html(function(index, html){
            return html + '+';
        });
    });
});

```



```

}); // h1 = this 자신에다가 출력해줘

// 2) $(selector).on(object);

$(function(){
    $('h1').on({
        mouseenter: function(){$(this).addClass('reverse');},
        mouseleave: function(){$(this).removeClass('reverse');}
    });
});
*/

/* 2. 이벤트 연결2
// - hover():mouseenter 이벤트와 mouseleave 이벤트를 동시에 연결.
// - $(selector).hover(function(event){}, function(event){});

$(function(){
    $('h1').hover(function(){
        $(this).addClass('reverse');
    }, function(){
        $(this).removeClass('reverse');
    });
});

// */
/* 3. 이벤트 제거
// - off() ---> one()
$(function(){
    //이벤트 연결
    $('h1').click(function(){
        $(this).html('CLICK');

        // 이벤트 제거
        $(this).off();
    });
});
// */

/* 4. 매개변수 context
// - $(selector, context)
//   . context:selector가 적용하는 범위를 한정.
//   . 특정 부분의 선택자를 선택하고 싶을 때 사용하는 매개변수.
// HTML태그를 자바스크립트에서는 도큐먼트 객체라고 부른다.

$(function(){
    // 이벤트 연결
    $('div').click(function(){
        var header = $('h1', this).text();
        var paragraph = $('p', this).text();

        alert(header + ' : ' + paragraph);
    });
});
// */

/* 5. 이벤트 강제 발생
// - 사용 형태
// 1) $(selector).trigger(eventName);

$(function(){
    // 이벤트 연결
    $('h1').click(function(){
        $(this).html(function(index, html){
            return html + '*';
        });
    });
});

```

```

        setInterval(function(){
            //$( 'h1' ).last().trigger('click');
            $( 'h1' ).last().click();
        }, 1000);
    });

// 2) $(selector).trigger(eventName, data);

$(function(){
    $( 'h1' ).click(function(event, data1, data2){
        alert(data1+ ' : '+data2);
    });

    // 강제로 이벤트 발생
    $( 'h1' ).eq(0).trigger('click', [273, 52]);
});

// */

/* 6. 기본 이벤트 제거
$(function(){
    $( 'a' ).click(function(event){
        var e = event || window.event;

        $(this).css('background-color', 'blue');

        e.cancelBubble = true;
        if(e.stopPropagation){
            e.stopPropagation(); // 이벤트 전달 기능 제거.
        }

        e.preventDefault(); // (a 태그의) 기본 이벤트 제거.

    });

    $( 'h1' ).click(function(){
        $(this).css('background-color', 'red');
    });
});
// */

/* 7. 이벤트 통합 메서드
// - on():이벤트를 연결.
// - off():이벤트를 제거.
$(function(){
    $( 'div' ).on('click', function(){
        $(this).clone().appendTo('body');
    });
});
// */

/* 8. 마우스 이벤트
// - click : 마우스 클릭할 때 이벤트 발생.
// - dbclick : 마우스 더블클릭할 때 이벤트 발생.
// - mousedown : 마우스 버튼을 누를 때 이벤트 발생.
// - mouseup : 마우스 버튼을 땔 때 이벤트 발생.
// - mouseenter : 마우스가 요소의 경계 외부에서 내부로 이동할 때 발생.
// - mouseleave : 마우스가 요소의 경계 내부에서 외부로 이동할 때 발생.
// - mousemove : 마우스를 움직일 때 발생.
// - mouseout : 마우스가 요소를 벗어날 때 발생.
// - mouseover : 마우스가 요소 안에 들어올 때 발생.

$(function(){
    $( '.outer' ).mouseover(function(){
        $( 'body' ).append('<h1>MOUSEOVER</h1>');
    }).mouseenter(function(){ // 문서 객체의 안에 있는지 외부에 있는지만 따짐.

```

```

        $('body').append('<h1>MOUSEENTER</h1>');
    }).mouseleave(function(){
        // $('body').append('<h1>MOUSELEAVE</h1>');
    });
});
// */

/* 9. 키보드 이벤트
// - keydown : 키보드가 눌려질 때 발생.
// - keypress : 글자가 입력될 때 발생(한글지원x).
// - keyup : 키보드가 떼어질 때 발생.

$(function(){
    $('textarea').keyup(function(event){
        // 남은 글자수를 구합니다.
        var inputLength = $(this).val().length;
        var remain = 150-inputLength;

        $('h1').html(remain);

        if(remain >= 0){
            $('h1').css('color', 'black');
        }else{
            $('h1').css('color', 'red');
        }
    });
});
// */

</script>
</head>

<body>

<!-- 1/2/3. 이벤트 연결/제거
<h1>Header-0</h1>
<h1>Header-1</h1>
<h1>Header-2</h1>
-->

<!-- 1-3)
<div id="wrap">
    <h1>Header</h1>
</div>
-->

<!-- 4. 매개변수 context
<div>
    <h1>Header1</h1>
    <p>Paragraph1</p>
</div>
<div>
    <h1>Header2</h1>
    <p>Paragraph2</p>
</div>
<div>
    <h1>Header3</h1>
    <p>Paragraph3</p>
</div>
-->

<!-- 5-1). 이벤트 강제 발생
<h1>Start: </h1>
<h1>Start: </h1>
-->

<!-- 5-2). 이벤트 강제 발생

```

```

        <h1>TRIGGER</h1>
    -->

<!-- 6. 기본 이벤트 제거
    <h1>
        <a href="https://www.naver.com">네이버</a>
    </h1>
-->

<!-- 7. 이벤트 통합 메서드
    <div>
        <h1>$(selector).on()</h1>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
    </div>
-->

<!-- 8. 마우스 이벤트
    <div class="outer">
        <div class="inner"></div>
    </div>
-->

<!-- 9. 키보드 이벤트 -->
    <div>
        <p>지금 내 생각을</p>
        <h1>150</h1>
        <textarea cols="70" rows="5"></textarea>
    </div>

</body>
</html>

```