

# ch08. jQuery01

---

# 학습 목차

---

1. jQuery 개요
2. jQuery 이용방법
3. jQuery 명령 형식
4. jQuery 선택자
5. jQuery 배열 관리
6. 문서 객체 선택과 탐색
7. 문서 객체 조작
8. 문서 객체 삽입

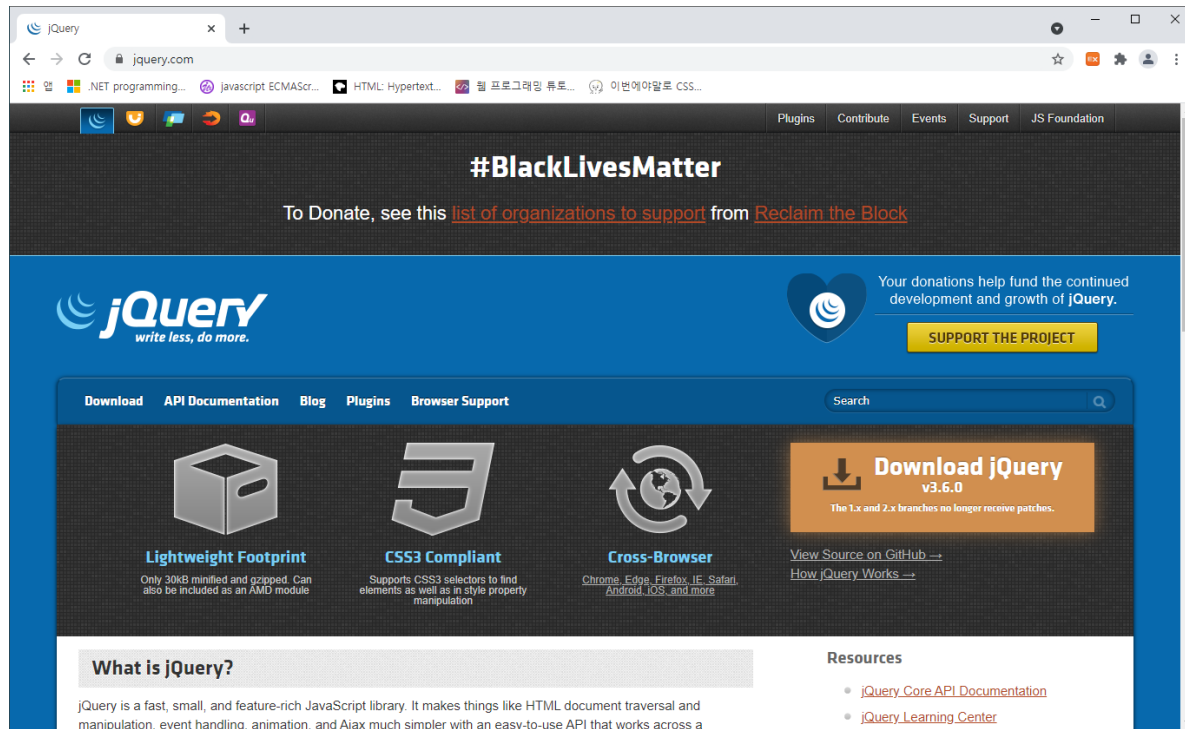
# 학습 목표

---

- jQuery 이용 방법을 이해할 수 있습니다
- jQuery 선택자를 이해하고 활용할 수 있습니다.
- jQuery 명령 형식을 이해하고 활용할 수 있습니다
- jQuery 배열을 이해하고 활용할 수 있습니다.
- jQuery 부모 노드 검색을 할 수 있다
- jQuery 자식 노드 검색을 할 수 있다
- jQuery 형제 노드를 검색할 수 있다
- jQuery 문서 객체 속성과 스타일을 조작할 수 있다
- jQuery 문서 객체 생성, 삽입, 복제, 제거 작업을 할 수 있다.

# 1. jQuery 개요

- 2006년 존 레식(John Resig)이 개발
- **모든 브라우저에서 동작하는** 자바스크립트 라이브러리
- 무료로 사용할 수 있는 오픈소스라이브러리
- 공식 웹 사이트 주소는 <http://jquery.com/>



# 1. jQuery 개요

---

- jQuery 코드와 자바스크립트 코드의 비교
  - 프로그래밍 시간 단축
  - 코드량 단축

```
// jQuery 코드
<h1 id="title"> 자바스크립트 프로그래밍</h1>
...

$('#title').html("jQuery 프로그래밍");
$('#title').css (backgroundColor, "green");
```

```
// 자바스크립트 코드
<h1 id="title"> 자바스크립트 프로그래밍</h1>
...

// <h1> 요소의 내용 변경
var tag1=document.getElementById ("title");
tag1.innerHTML="jQuery 프로그래밍";
tag1.style.backgroundColor="green";
```

# 1. jQuery 개요

---

- jQuery 주요 작업
  - DOM과 관련된 처리
    - HTML 요소들의 내용이나 속성 변경
    - HTML 요소들의 CSS3 스타일 속성 변경
    - HTML DOM 트리의 변경(즉 요소 노드 추가, 삭제 처리)
  - 이벤트 연결
    - HTML 요소들에 대한 다양한 효과 지정
    - 시각적 효과
  - Ajax 애플리케이션 개발
    - 서버와의 비동기 통신 지원
  - 웹 브라우저에 종속되지 않는 코드 작성

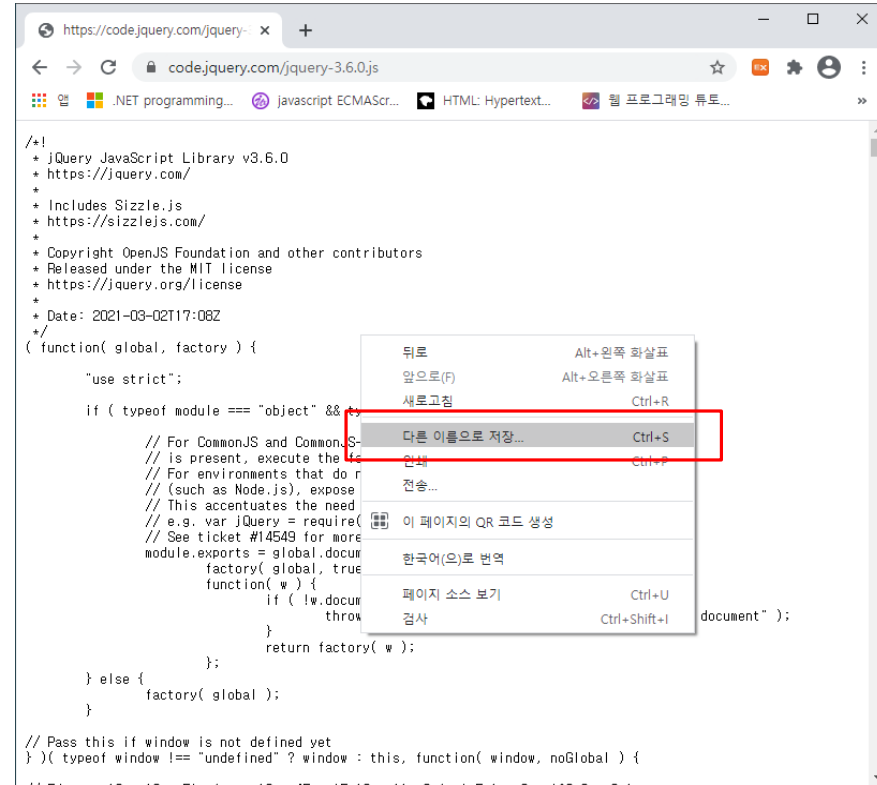
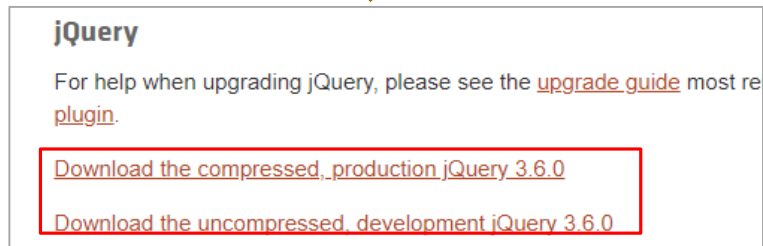
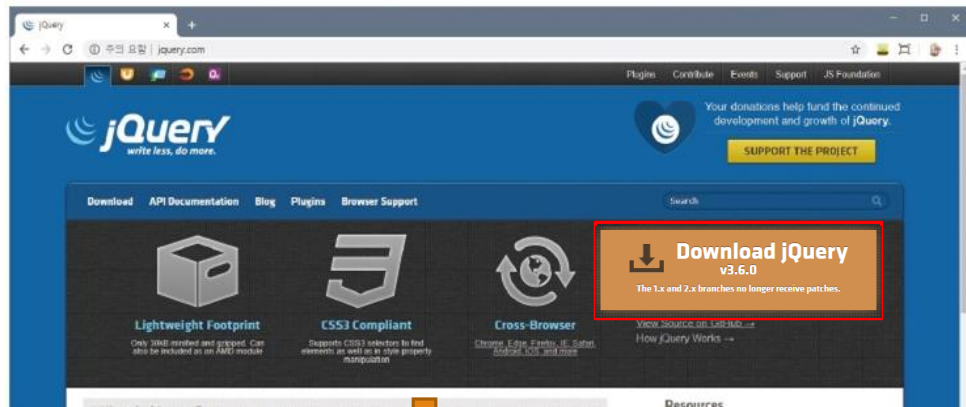
## 2. jQuery 이용방법

---

- jQuery는 자바스크립트로 작성된 라이브러리 파일(파일 확장자 : '.js' )
- jQuery 라이브러리 파일은 jQuery 홈페이지에서 다운받을 수 있음
  - 최소 버전, 일반 버전, 슬림 버전, 모바일 버전 등 제공
- jQuery 이용방법 : 외부 자바스크립트 파일 연결 방법과 동일
  - 1) 직접 jQuery 파일을 다운받아 PC에 설치 후 연결함
  - 2) jQuery 라이브러리 서비스 제공. CDN(Content Delivery Network) 호스트 서비스 이용 연결

## 2. jQuery 이용방법 - jQuery 다운로드

- <https://jquery.com>에 접속 jQuery 파일 다운로드
- 파일을 다운로드하여 외부 자바스크립트 파일로 추가



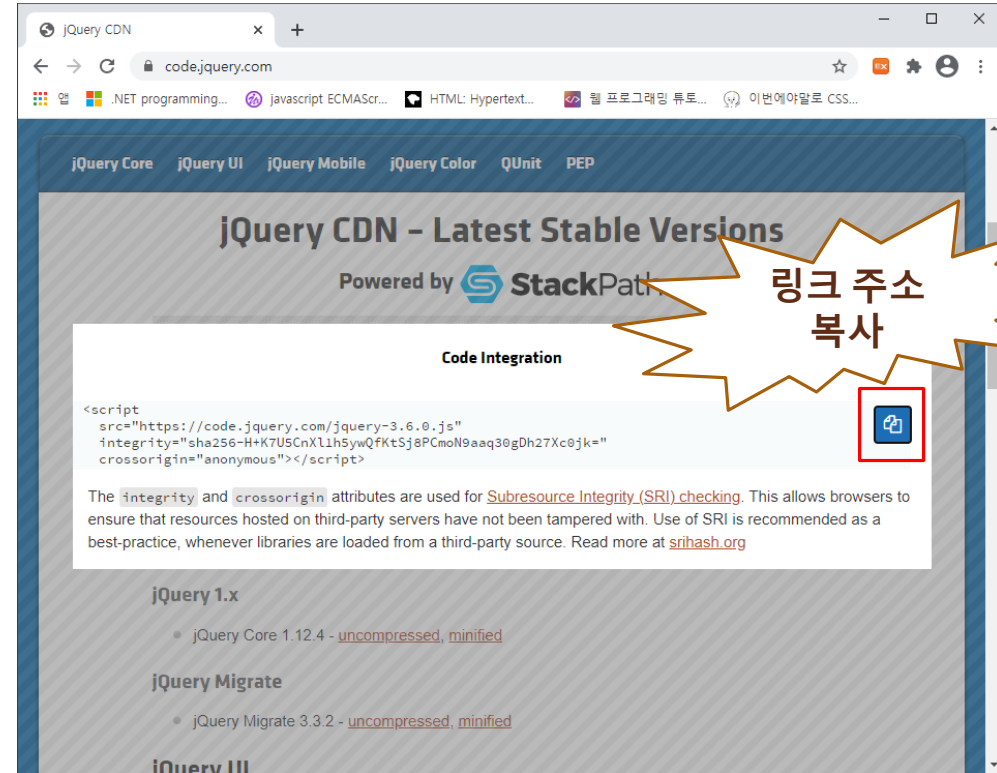
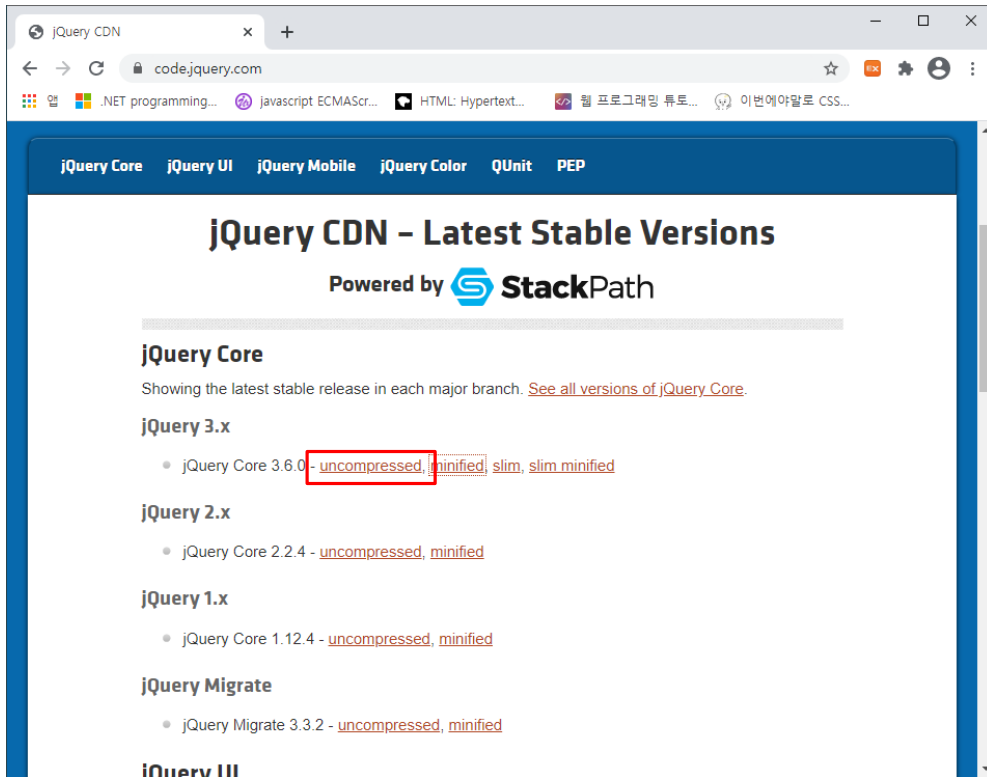
### ■ jQuery 파일명

- 1) \*.js : uncompressed(압축되지 않은) 버전
- 2) \*.min.js : Minified(축소된) 버전, 파일의 용량을 최소화한 지핑(zipping)파일



## 2. jQuery 이용방법 - CDN 호스트 사용

- CDN 호스트 사용 :
  - <https://code.jquery.com/>에 접속 원하는 파일 (uncompressed 버전 사용) 링크 주소 복사
- 실제 웹 서비스를 운영할 때는 CDN 호스트 서비스를 이용해야 함



## 2. jQuery 이용방법

- 다운로드한 jQuery 파일 사용

```
<head>
  <script src="../jquery-3.7.0.min.js"></script>
  <script>
    $(document).ready(function () {
      $('h1').css('color', 'red');
    });
  </script>
</head>
```

```
<body>
  <h1 class="item">jQuery 테스트 </h1>
  <h1 id="target">jQuery 테스트</h1>
  <label> 이름 : </label> <input type="text" />
</body>
```

- CDN 호스트 사용

```
<head>
  <script src="https://code.jquery.com/jquery-3.7.0.js"></script>
  <script>
    $(document).ready(function () {
      $('h1').css('color', 'red');
    });
  </script>
</head>
```

최신버전을 사용하려면

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

# 3. jQuery 명령 형식

---

- 일반적인 jQuery의 객체 사용 구조
  - jQuery를 사용한 모든 웹 페이지 시작 코드

```
$(document).ready(function () {  
  
    //jQuery 코드 작성  
  
});
```

- 웹 문서가 준비되면 매개변수로 전달된 콜백 함수 실행
- window.onload = function(){}과 비슷하지만 여러 개의 콜백 함수 등록 가능
- 간단한 형태

```
$(function () {  
    //jQuery 코드 작성  
});
```

# 3. jQuery 명령 형식

- jQuery 메소드의 기본 형태 - 문서 객체를 다룰 때 사용하는 형태

```
$(선택자).작업함수(); 또는 jQuery(선택자).작업함수(); // jQuery 식별자를 $로 대체
```

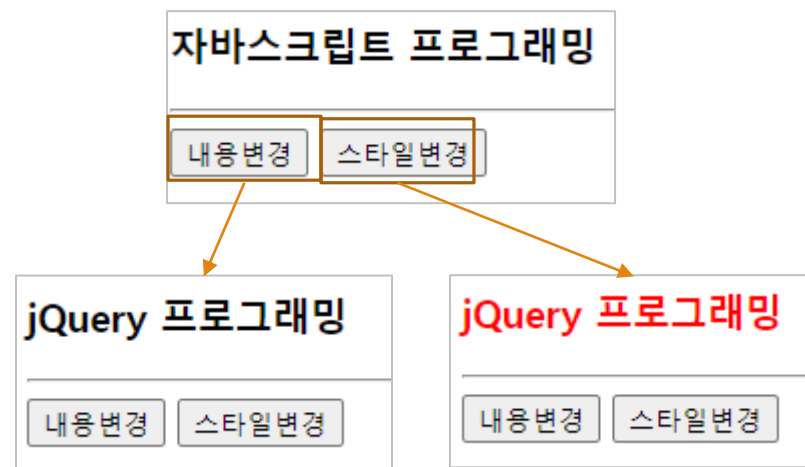
```
예) $('#title1').html('jQuery 프로그래밍');
```

- jQuery 라이브러리는 \$() 메소드 활용
  - \$() 메소드의 매개 변수에는 문서 객체, CSS 형식, HTML 형식의 문자열 삽입  
\$(매개변수).메소드(매개변수, 매개변수..)
  - \$() 메소드 리턴 값 - jQuery 객체
    - `$('h1').css('color', 'red');` => `let $h=$('h1');` `$h.css('color', 'red');`
  - jQuery 객체 생성
    - 일반 문서 객체로 jQuery 객체 생성
    - CSS 선택자로 jQuery 객체 생성
    - HTML 문자열로 jQuery 객체 생성

# 3. jQuery 명령 형식

```
<head>
  <title>jQuery Basic</title>
  <script src="https://code.jquery.com/jquery-3.7.0.js"> </script>
  <script>
    $(function () { // DOM이 완전히 로드하면 매개변수로 전달된 콜백 함수 실행
      $('#btn1').click(function () {
        $('#t1').html('jQuery 프로그래밍');
      });

      $('#btn2').click(function () {
        $('#t1').css('color', 'red');
      });
    });
  </script>
</head>
<body>
  <h3 id="t1"> 자바스크립트 프로그래밍 </h3> <hr>
  <button id="btn1"> 내용변경 </button>
  <button id="btn2"> 스타일변경 </button>
</body>
```



# 4. jQuery 선택자

- 기본 선택자
  - 전체 선택자 : \*
  - 클래스 선택자 : `$('.item').css('color', 'red');`
  - 태그 선택자 : `$('h1').css('color', 'red');`
  - 아이디 선택자 : `$('#target').css('color', 'red');`
  - 여러 태그 선택자 : `$('h1, p').css('color', 'red');`
- 자손 선택자 :
  - `$('div>ul').css('color', 'blue');`
- 후손 선택자 :
  - `$('div ul').css('color', 'orange');`
- 속성 선택자 :
  - `$('input[type="text"]').val('이름을 입력하세요');`

```
<head>
  <script src="https://code.jquery.com/jquery-3.7.0.js"></script>
  <script>
    $(function() {
      $('div.menu').css({'border':'1px solid blue','width':'200px'});
      $('.menu ul li').css('list-style', 'none');
      $('span').css('color', 'blue');
      $('input[type=text]').val('이름을 입력하세요');
      $('#target').html('jQuery Selector');
    });
  </script>
</head>
<body>
  <h1 id="target">jQuery 테스트</h1>
  <div class="menu">
    <ul>
      <li>파이썬</li>
      <li>창의코딩웹</li>
      <li>자바프로그래밍</li>
      <li>C프로그래밍</li>
      <li>웹프로그래밍</li>
    </ul>
  </div>
  <br>
  <span> 이름 : </span> <input type="text" />
</body>
```

# 4. jQuery 선택자

---

- 필터 선택자
  - 선택자 중에 : (콜론) 기호를 포함하는 선택자
  - 기본 선택자에 비해 활용도가 낮음 – 필요 시 찾아서 사용
- 입력 양식 필터 선택자 : 입력 양식 문서 객체를 간단하게 선택할 수 있게 함
  - 종류
    - 요소 :button - type이 button인 모든 요소 선택
    - 요소 :checkbox - type이 checkbox 인 모든 요소 선택
    - 요소 :password - type이 password인 모든 요소 선택
    - 요소 :radio - type이 radio인 모든 요소 선택
    - 요소 :focus – 포커스 된 요소 선택
    - 요소 :checked – 체크되었거나 선택된 모든 요소 선택
    - 요소 :input – 모든 input, textarea, select and button 요소 선택
    - 요소:disabled – 비 활성화된 입력 양식을 선택
    - 요소:enabled - 활성화된 입력 양식을 선택
    - 요소:selected - option 객체 중 선택된 태그를 선택

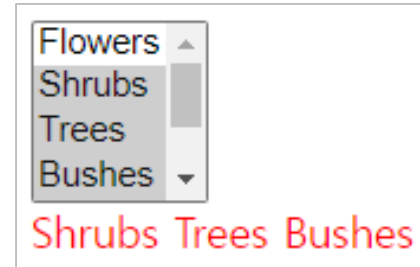
# 4. jQuery 선택자

```
<head> <!-- 필터 선택자 사용 예 -->
  <meta charset="UTF-8">
  <title>jQuery 입력 양식 필터 선택자</title>
  <script src="https://code.jquery.com/jquery-3.7.0.js"> </script>
  <script>
    $(function () {
      $("p:hidden").css({'display': 'block', 'color': 'green', 'fontSize': '20px'});
      $("input:text").css("border", "1px dashed #f00");
      $("input:checked").attr("checked", false); //checked인 <input>요소를 선택하여 속성을 false로 변경
      $("option:selected").attr("selected", false);
    });
  </script>
</head>
<body>
  <p>내용1</p> <p style="display:none;">내용2</p>
  <p><label for="user_name">이름</label> <input type="text" name="user_name" id="user_name" /></p>
  <p id="block1">
    <input type="checkbox" name="accept" value="독서" checked="checked" />독서
    <input type="checkbox" name="accept" value="등산" />등산</p>
  <p><select name="rel_site" id="resl_site">
    <option value="네이버">네이버</option>
    <option value="네이트" selected="selected">네이트</option></select></p>
</body>
```



# 4. jQuery 선택자

```
<head> <!-- 필터 선택자 사용 예 -->
<meta charset="UTF-8">
<title>jQuery 입력 양식 필터 선택자</title>
<script src="https://code.jquery.com/jquery-3.7.0.js"> </script>
<script>
  $(function () {
    $("select").on("change", function () {
      let str = "";
      $("select option:selected").each(function () {
        str += $(this).text() + " ";
      });
      $("div").text(str);
    })
  })
</script>
</head>
<body>
  <select name="garden" multiple="multiple">
    <option>Flowers</option> <option>Shrubs</option>
    <option>Trees</option> <option selected>Bushes</option>
    <option>Grass</option> <option>Dirt</option>
  </select>
</body>
```



# 4. jQuery 선택자

- 위치 필터 선택자

붉은색 텍스트로 표시된 선택자는 3.4버전에서 deprecated

- 요소:odd() - 홀수 번째 위치한 문서 객체 선택 -> odd()
- 요소:even() - 짝수 번째 위치한 문서 객체 선택 -> even()
- 요소:first() - 첫 번째 위치한 문서 객체 선택 -> first()
- 요소:last() - 마지막에 위치한 문서 객체 선택 -> last()

- 함수 필터 선택자

- 요소:contains(문자열) - 특정 문자열을 포함하는 모든 문서 객체 선택
- 요소:eq(n) - n 번째 위치하는 문서 객체 선택(0이 첫번째 요소) -> eq()
- 요소:gt(n) - n번째 초과에 위치하는 모든 문서 객체 선택 -> gt()
- 요소:has(태그) - 특정 태그가 있는 모든 문서 객체를 선택
  - 예: \$('div:has(p)') => <p> 태그를 갖는 모든 <div> 선택
- 요소:lt(n) - n 번째 미만에 위치하는 모든 문서 객체 선택 -> lt()
- 요소:not(선택자) - 선택자와 일치하지 않는 모든 문서 객체 선택
  - 예: \$('input:not(:empty)) => 비워 있지 않은 모든 입력 객체 선택
- 요소:nth-child(3n+1) - 3n+1번째에 위치하는 모든 문서 객체를 선택(1, 4, 7, ... 번째)

# 4. jQuery 선택자

```
<head>
<title>jQuery 위치 필터 선택자</title>
<style>
  table{ border-collapse: collapse; }
  tr{ border-bottom: 1px solid black; opacity: 0.7; }
</style>
<script src="https://code.jquery.com/jquery-3.7.0.js"></script>
<script>
```

소속	학년	주소
빅데이터	1	서울시 강남구
스마트IoT	2	강원도 춘천시
컨텐츠IoT	3	강원도 원주시
빅데이터	3	서울시 송파구
스마트IoT	2	서울시 노량진구
컨텐츠IoT	1	서울시 강서구

```
$(function () {
  $('tr:odd').css('background', 'red');
  $('tr:even').css('background', 'yellow');
  $('tr:first').css('background', 'green').css('color','white'); //체이닝 적용
});
</script>
</head>
```

```
<script>
$(function () {
  $('tr').odd().css('background', 'red');
  $('tr').even().css('background', 'yellow');
  $('tr').first().css('background', 'green').css('color','white');
});
</script>
```

```
<body>
<table>
  <tr><th>소속</th><th>학년</th><th>주소</th></tr>
  <tr><td>빅데이터</td><td>1</td><td>서울시 강남구</td></tr>
  <tr><td>스마트IoT</td><td>2</td><td>강원도 춘천시</td></tr>
  <tr><td>컨텐츠IoT</td><td>3</td><td>강원도 원주시</td></tr>
  <tr><td>빅데이터</td><td>3</td><td>서울시 송파구</td></tr>
  <tr><td>스마트IoT</td><td>2</td><td>서울시 노량진구</td></tr>
  <tr><td>컨텐츠IoT</td><td>1</td><td>서울시 강서구</td></tr>
</table>
</body>
```

## 4. jQuery 선택자

소속	학년	주소
빅데이터	1	서울시 강남구
스마트IoT	2	강원도 춘천시
컨텐츠IoT	3	강원도 원주시
빅데이터	3	서울시 송파구
스마트IoT	2	서울시 노랑진구
컨텐츠IoT	1	서울시 강서구

```
<script>
$(function () {
    $('tr:eq(0)').css('background', 'green').css('color', 'white');
    $('td:nth-child(3n+1)').css('background', 'red');
    $('td:nth-child(3n+2)').css('background', 'yellow');
    $('td:nth-child(3n+3)').css('background', 'brown');
});
</script>
```

`$('tr').eq(0).css('background', 'green').css('color', 'white');`

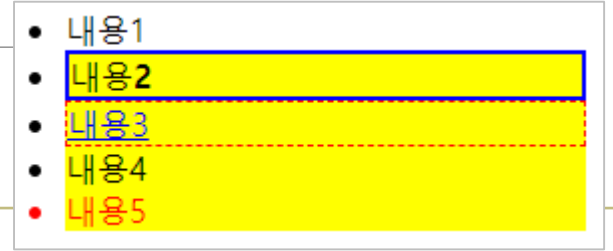
# 4. jQuery 선택자

```
<head>
  <meta charset="UTF-8">
  <title>jQuery 함수 필터 선택자</title>
  <style>
    #wrap{ width : 300px; }
  </style>
  <script src="https://code.jquery.com/jquery-3.7.0.js"> </script>
  <script>
    $(function () {
      //선택한 li 요소 중 "내용3"을 포함하는 요소만 선택
      $("#menu li:contains('내용3')).css("border", "1px dashed red");

      //선택한 li 요소 중 <strong>을 포함하는 요소만 선택
      $("#menu li:has('strong')).css("border", "2px solid blue");

      //아이디 값 menu의 하위 li 요소에서 첫번째 요소만 제외
      $("#menu li").not(":first").css("background-color", "yellow");

      //아이디 값 menu의 하위 li 요소에서 인덱스가 4인 요소에서 <strong> 요소만 선택
      $("#menu li:eq(4)").css("color", "red"); // $("#menu").eq(4).css("color", "red");
    });
  </script>
</head>
```



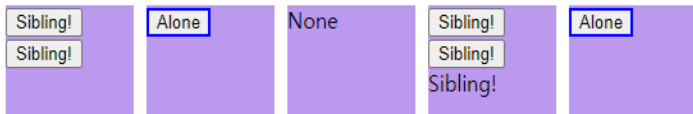
```
<body>
  <div id="wrap">
    <ul id="menu">
      <li>내용1</li>
      <li>
        <span>내용<strong>2</strong></span>
      </li>
      <li><a href="#">내용3</a></li>
      <li>내용4</li>
      <li>내용5</li>
    </ul>
  </div>
</body>
```

# 4. jQuery 선택자

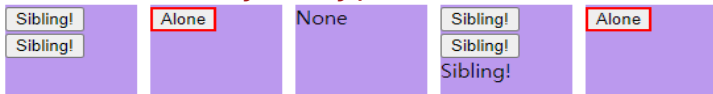
- 자식 필터 선택자

- 요소 :first-child
- 요소 :first-of-type
- 요소 :last-child
- 요소 :last-of-type
- 요소 :nth-child()
- 요소 :nth-last-child()
- 요소 :nth-last-of-type()
- 요소 :nth-of-type()
- 요소 :only-child
- 요소 :only-of-type()

div button:only-child



div button:only-of-type



```
<head>
  <style>
    div { width: 100px; height: 80px; margin: 5px; float: left; background: #b9e; }
  </style>
  <script src="https://code.jquery.com/jquery-3.7.0.js"> </script>
</head>
<body>
  <div>
    <button>Sibling!</button> <button>Sibling!</button>
  </div>
  <div>
    <button>Sibling!</button>
  </div>
  <div>None </div>
  <div>
    <button>Sibling!</button> <button>Sibling!</button> <span>Sibling!</span>
  </div>
  <div>
    <button>Sibling!</button>
  </div>
  <script>
    // $("div button:only-child").text("Alone").css("border", "2px blue solid");
    $( "div button:only-of-type" ).text( "Alone" ).css( "border", "2px red solid" );
  </script>
</body>
```

# 5. jQuery 배열 관리

---

- jQuery로 배열을 관리할 때는 `each( )` 메소드를 사용함
- jQuery 배열은 선택자로 여러 개의 문서 객체를 선택할 때 생성
- 두 가지 형태
  - `$.each(object, function(index, item) { } )`
    - 자바스크립트 배열의 각 원소를 접근하거나
    - 객체의 각 속성을 접근할 때 사용함
    - 매개변수
      - `object`: 배열 객체
      - `function`: 원소를 처리하는 함수
  - `$(selector).each(function(index, item) { } )`
    - `selector`로 선택된 모든 문서 객체의 각각을 접근할 때 사용

# 5. jQuery 배열 관리

- \$(selector).map(function(index, element))
  - 선택된 요소에서 함수를 사용하여 매치되는 각각의 요소들을 새로운 제이 쿼리 객체(배열)로 반환
  - get() 메소드와 함께 사용

```
<style>
  p { color: red; }
</style>
<script src="https://code.jquery.com/jquery-3.7.0.js"></script>
<body>
<p><b>Values: </b></p>
<form>
  <input type="text" name="name" value="John">
  <input type="text" name="password" value="password">
  <input type="text" name="url" value="https://johnresig.com/">
</form>
<script>
$( "p" )
  .append( $( "input" ).map(function() { return $( this ).val(); })
  .get()
  .join( ", " ) );
</script>
</body>
```

Values: John, password, https://johnresig.com/



# 5. jQuery 배열 관리

- get()
  - 선택한 요소를 배열(Array)로 반환
  - \$(선택자).get()
    - 모든 요소 검색
  - \$(선택자).get(index)
    - index에 해당하는 요소 검색

```
<body>
  Reversed - <span> </span>
  <br>
  noeName - <span> </span>
  <div>One</div>
  <div>Two</div>
  <div>Three</div>
</body>
```

```
Reversed - Three-Two-One
noeName - DIV
One
Two
Three
```

```
<head>
  <style>
    span{
      color :brown; font-weight: bold;
    }
  </style>
  <script src="https://code.jquery.com/jquery-3.7.0.js"> </script>
  <script>
    function display(divs) {
      let a = [];
      divs.forEach(element =>{
        a.push(element.innerHTML);
      });
      $("span").eq(0).text(a.join("-"));
    }

    $(function () {
      display($(".div").get().reverse());
      $("span").eq(1).text($(".div").get(1).nodeName));
    });
  </script>
</head>
```

# 5. jQuery 배열 관리

```
<head>
  <meta charset="UTF-8">
  <title>jQuery 배열 관리</title>
  <style>
    .high-light0{ background: yellow; }
    .high-light1{ background: blue; }
    .high-light2{ background: greenyellow; }
  </style>
  <script src="https://code.jquery.com/jquery-3.7.0.js"> </script>
  <script>
    $(function () {
      $('h1').each(function(index){
        $(this).addClass('high-light'+index);
      });
    });
  </script>
</head>
<body>
  <h1>contents01 </h1>
  <h1>contents02 </h1>
  <h1>contents03 </h1>
</body>
```

contents01

contents02

contents03

```
<script>
  $(function () {
    const array = [...$('h1')];
    array.forEach((value, index) => value.className = `high-light${index}`);
  });
</script>
```

# 5. jQuery 배열 관리

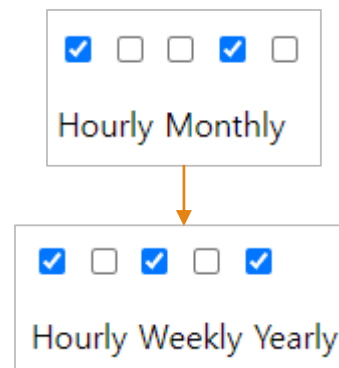
```
<head>
  <meta charset="UTF-8">
  <title>jQuery 배열 관리</title>
  <style>
    a { text-decoration: none; }
  </style>
  <script src="https://code.jquery.com/jquery-3.7.0.js"></script>
  <script>
    $(function () {
      const array = [
        { name: 'hallym', link: 'https://www.hallym.ac.kr' },
        { name: 'smart', link: 'https://smartlead.hallym.ac.kr' },
      ]
      $.each(array, function (index, item) {
        let result = "";
        result += `<a href = "${item.link}" >`;
        result += ` <h1> ${item.name} </h1>`;
        result += `</a>`;
        document.body.innerHTML += result;
      });
    });
  </script>
</head>
<body>
</body>
```

```
<script>
  $(function () {
    const array = [
      { name: 'hallym', link: 'https://www.hallym.ac.kr' },
      { name: 'smart', link: 'https://smartlead.hallym.ac.kr' }];

    array.forEach((item) => {
      let result = "";
      result += `<a href = "${item.link}" >`;
      result += ` <h1> ${item.name} </h1>`;
      result += `</a>`;
      document.body.innerHTML += result;
    })
  });
</script>
```

# 5. jQuery 배열 관리

```
<head>
  <meta charset="UTF-8">
  <title>jQuery 배열 관리</title>
  <style>
    a { text-decoration: none; }
  </style>
  <script src="https://code.jquery.com/jquery-3.7.0.js"></script>
  <script>
    const countChecked = function() {
      let str="";
      $("input:checked").each(function(){ str += $(this).val() + ' '; });
      $('div').text(str);
    };
    $(function () {
      countChecked();
      $('input[type=checkbox]').on('click', countChecked);
    });
  </script>
</head>
</body>
```



```
<body>
  <form>
    <p>
      <input type="checkbox" name="newsletter" value="Hourly" checked>
      <input type="checkbox" name="newsletter" value="Daily">
      <input type="checkbox" name="newsletter" value="Weekly">
      <input type="checkbox" name="newsletter" value="Monthly" checked>
      <input type="checkbox" name="newsletter" value="Yearly">
    </p>
  </form>
  <div></div>
</body>
```

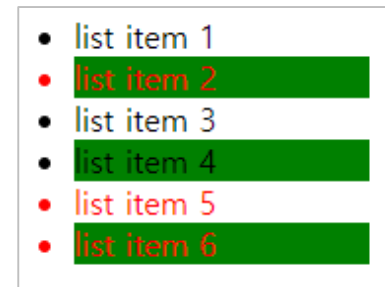
# 6. 문서 객체 선택과 탐색

- 자바스크립트만으로 문서 객체 모델을 다루려면 복잡
- jQuery를 이용하면 쉽게 다룰 수 있다.

- 검색한 노드 다루기
  - filter() : 검색한 문서 객체 중에서 특정 객체만 찾기
    - \$(선택자01).filter(선택자02);
    - \$(선택자).filter(function(index, element));
      - 리턴 하는 함수 값에 따라 문서 객체를 선택

- 검색한 문서 객체 개수
  - length 프로퍼티
    - 예) \$('div').length //div 개수

```
<head>
  <style>
    ul { width: 150px; }
  </style>
  <script src="https://code.jquery.com/jquery-3.7.0.js"> </script>
  <script>
    $(document).ready(function () {
      $('li').filter(':nth-child(2n)').css('background', 'green');
      $('li').filter(function (index) {
        return index % 3 === 1 || $(this).attr('id') === 'sixth';
      })
        .css("color", "red");
    });
  </script>
</head>
<body>
  <ul>
    <li>list item 1 </li>
    <li>list item 2 </li>
    <li>list item 3 </li>
    <li>list item 4 </li>
    <li>list item 5 </li>
    <li id="sixth">list item 6 </li>
  </ul>
</body>
```



# 6. 문서 객체 선택과 탐색

- 검색한 노드 다루기

메소드	설명
end()	문서 객체 선택을 한 단계 뒤로 돌림
eq(index)	검색한 문서 객체의 index에 해당하는 객체 선택, 양수, 0, 음수 모두 입력 가능
first()	첫 번째에 위치하는 문서 객체 선택
last()	마지막에 위치하는 문서 객체 선택
odd()	홀수 번째 문서 객체 선택
even()	짝수 번째 문서 객체 선택
slice(start [,end])	일치하는 문서 객체의 일부분만 선택
add(선택자)	문서 객체를 추가적으로 선택
is(선택자), is(function)	문서 객체의 특징을 판별
find(선택자)	검색한 문서 객체의 자식(후손 포함) 객체 중 특정 객체 찾기
get(index)	jQuery 객체와 일치하는 DOM 객체 검색
parent(), parents(선택자), parents()	특정 객체의 상위 부모 객체와 모든 조상 객체 찾기
children(선택자), children()	특정 객체의 자식만 찾기
prev(), prevAll(선택자), prevAll()	이전 형제 객체 선택
next(), nextAll(선택자), nextAll()	다음 형제 객체 선택

# 6. 문서 객체 선택과 탐색

- 체이닝을 이용한 필터 적용 – end(), filter()

```
<head>
  <meta charset="UTF-8">
  <title>체이닝을 이용한 필터 적용</title>
  <script src="https://code.jquery.com/jquery-3.7.0.js"></script>
  <script>
    $(function () {
      $('h1').css('background', 'orange');
      $('h1:even').css('color', 'white');
      $('h1:odd').css('color', 'red');
    });
  </script>
</head>
```

선택자 1

선택자 2

선택자 3

선택자 4

선택자 5

```
<body>
  <h1>선택자 1</h1>
  <h1>선택자 2</h1>
  <h1>선택자 3</h1>
  <h1>선택자 4</h1>
  <h1>선택자 5</h1>
</body>
```

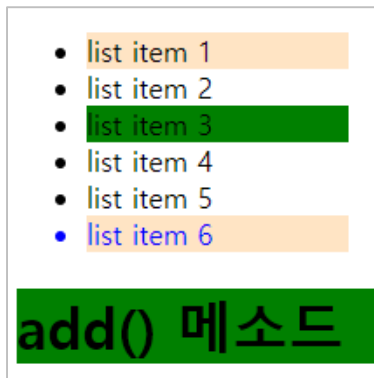
//체이닝을 이용한 필터 적용

```
$('h1').css('background', 'orange').filter(':even').css('color', 'white').end().filter(':odd').css('color', 'red')
```

## 6. 문서 객체 선택과 탐색

- last(), eq(), add() 사용 예

```
<body>
  <ul>
    <li>list item 1 </li>
    <li>list item 2 </li>
    <li>list item 3 </li>
    <li>list item 4 </li>
    <li>list item 5 </li>
    <li id="sixth">list item 6 </li>
  </ul>
  <h1>add() 메소드 </h1>
</body>
```



```
<head>
  <style>
    ul { width: 150px; }
    .highlight{ background-color: bisque; }
  </style>
  <script src="https://code.jquery.com/jquery-3.7.0.js"> </script>
  <script>
    $(function () {
      $('ul li').last().addClass('highlight');

      // 1번째 위치에 존재하는 <li> 태그 선택
      $('ul li').eq(0).addClass('highlight');

      // 맨 마지막 위치에 존재하는 <h1> 태그 선택
      // 인자가 음수인 경우는, 뒤에서부터 카운트함
      $('ul li').eq(-1).css('color', 'blue');

      // h1 요소와 ul li:nth-child(3) 요소를 선택하여 스타일 적용
      $('h1').add('ul li:nth-child(3)').css('background-color', 'green');
    });
  </script>
</head>
```



# 6. 문서 객체 선택과 탐색

- is(function) 사용 예

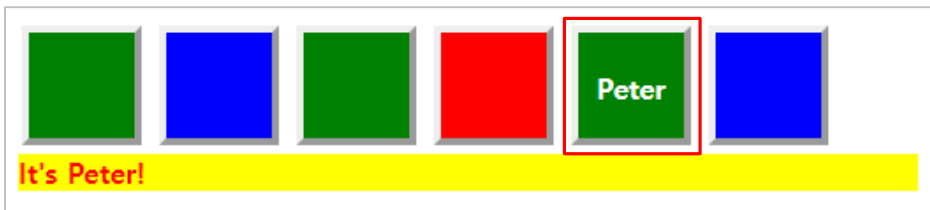
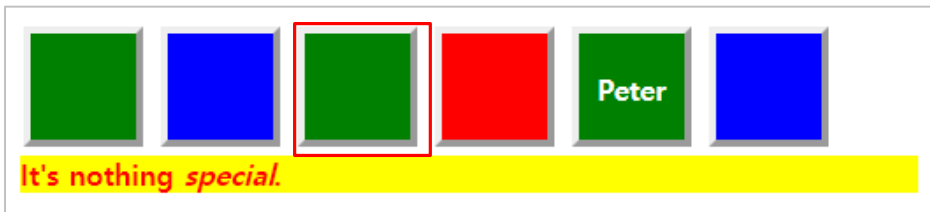
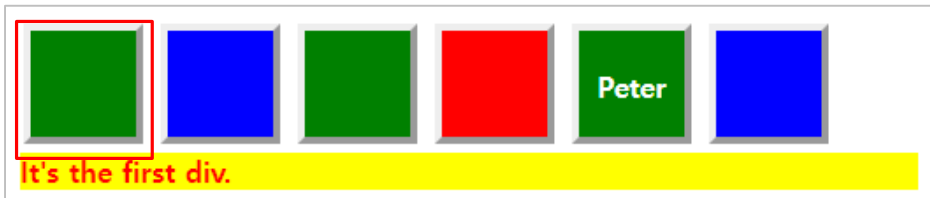
```
<head>
<script src="https://code.jquery.com/jquery-3.7.0.js"></script>
<script>
$(function () {
  $("li").click(function () {
    let li = $(this);
    isWithTwo = li.is(function () {
      return $('strong', this).length === 2;
      //return $(this).find('strong').length === 2;
    });
    if (isWithTwo) {
      li.css("background-color", "green");
    } else {
      li.css("background-color", "red");
    }
  });
});
</script>
</head>
```

```
<body>
<ul>
  <li><strong>list</strong> item 1 - one strong tag</li>
  <li><strong>list</strong> item <strong>2</strong> -
    two <span>strong tags</span></li>
  <li>list item 3</li>
  <li>list item 4</li>
  <li>list item 5</li>
</ul>
</body>
```

- list item 1 - one strong tag
- list item 2 - two strong tags
- list item 3
- list item 4
- list item 5

# 6. 문서 객체 선택과 탐색

- is(선택자) 사용 예

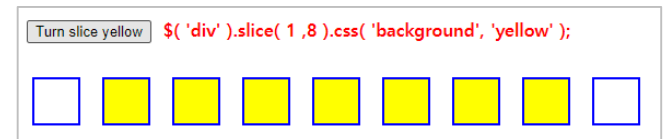
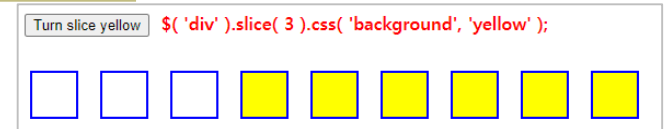


```
<script src="https://code.jquery.com/jquery-3.7.0.js"> </script>
<script>
  $(function () {
    $("div").on("click", function () {
      if ($(this).is(":first-child")) {
        $("p").text("It's the first div.");
      } else if ($(this).is(".blue, .red")) {
        $("p").text("It's a blue or red div.");
      } else if ($(this).is(":contains('Peter')")) {
        $("p").text("It's Peter!");
      } else {
        $("p").html("It's nothing <em>special</em>.");
      }
    });
  });
</script>
</head>
<body>
  <div> </div>
  <div class="blue"> </div>
  <div> </div>
  <div class="red"> </div>
  <div> <br> <span>Peter</span> </div>
  <div class="blue"> </div>
  <p> &nbsp;</p>
</body>
```

# 6. 문서 객체 선택과 탐색

- slice(start [,end]) 사용 예

```
<script>
function colorEm() {
  let $div = $("div");
  let start = Math.floor(Math.random() * $div.length);
  let end = Math.floor(Math.random() * ($div.length - start)) + start + 1;
  if (end === $div.length) {
    end = undefined;
  }
  $div.css("background", "");
  if (end) {
    $div.slice(start, end).css("background", "yellow");
  } else {
    $div.slice(start).css("background", "yellow");
  }
  $("span").text(`$( 'div' ).slice( ${start} ${ end ? ',' + end : ' ' } ).css( 'background', 'yellow' );`);
}
$(document).ready(function () {
  $("button").click(colorEm);
});
</script>
```



```
<body>
  <p> <button>Turn slice yellow</button>
    <span>Click the button!</span>
  </p>
  <div> </div>
  <div> </div>
  <div> </div>
  <div> </div>
  <div> </div>
  <div> </div>
  <div> </div>
  <div> </div>
  <div> </div>
  </body>
```

# 6. 문서 객체 선택과 탐색

- 자식 노드 찾기 -children(), find(), filter(), length 프로퍼티

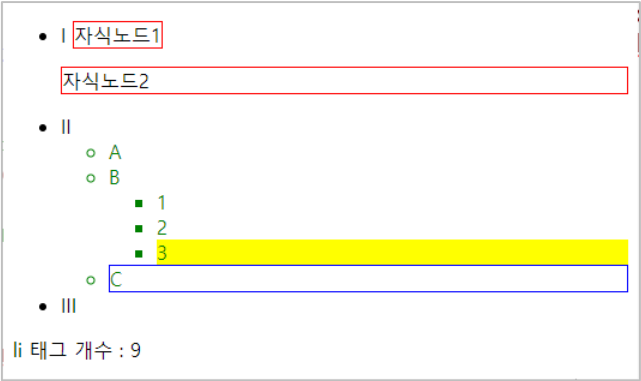
```
<body>
  <ul class="level-1">
    <li class="item-i">I
      <span>자식노드1 </span>
      <p>자식노드2 </p>
    </li>
    <li class="item-ii">II
      <ul class="level-2">
        <li class="item-a">A
        <li class="item-b">B
          <ul class="level3">
            <li class="item-1">1 </li>
            <li class="item-2">2 </li>
            <li class="item-3">3 </li>
          </ul>
        </li>
        <li class="item-c">C </li>
      </ul>
    </li>
    <li class="item-iii">III </li>
  </ul>
</div> </div>
</body>
```

```
<head>
  <script src="https://code.jquery.com/jquery-3.7.0.js"> </script>
  <script>
    $(function () {
      // 'li.item-i' 노드의 모든 자식 노드 선택
      $('li.item-i').children().css('border', '1px solid red');

      // 'ul.level-2' 자식 노드 중 'item-c' 노드 선택
      $('ul.level-2').children('.item-c').css('border', '1px solid blue');

      // 'ul.level-2' 노드의 후손 노드에서 'li' 노드 선택
      $('ul.level-2').find('li').css('color', 'green');

      // 'li' 노드 중 'item-3' 노드만 선택
      $('li').filter('.item-3').css('background-color', 'yellow');
      $('div').text('li 태그 개수 : $('li').length);
    });
  </script>
</head>
```



모든 자식 노드 찾기	<code>\$.children()</code>
특정 자식 노드 찾기	<code>\$.children('선택자')</code>
첫번째 자식 노드 찾기	<code>\$.children.first()</code> <code>\$.children.eq(0)</code>
마지막 자식 노드 찾기	<code>\$.children.last()</code> <code>\$.children.eq(-1)</code>

# 6. 문서 객체 선택과 탐색

- 부모 노드 찾기 - parent(), parents()



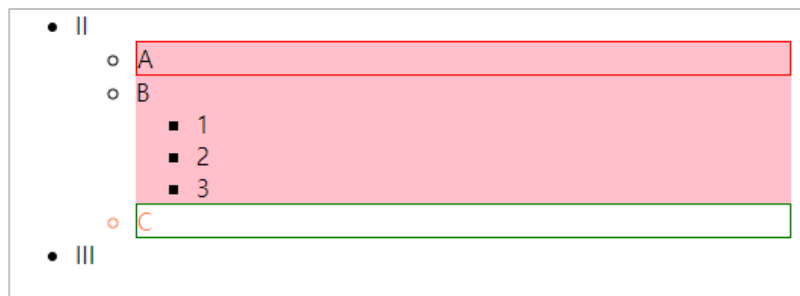
```
<html lang="en" style="background-color: yellow;">
<head>...</head>
<body style="background-color: yellow;">
  <ul class="level-1" style="background-color: yellow;">
    <li class="item-i" style="border: 1px solid red;">...</li>
    <li class="item-ii" style="background-color: yellow;">
      ::marker
      "II "
      <ul class="level-2" style="color: blue; background-color: yellow;">
        <li class="item-a">
          ::marker
          "A "
        </li>
        <li class="item-b">...</li>
        <li class="item-c">...</li>
      </ul>
    </li>
    <li class="item-iii">...</li>
  </ul>
```

```
<head>
  <script>
    $(document).ready(function () {
      $('.item-i>span').parent().css('border', '1px solid red');
      $('li.item-c').parents('ul.level-2').css('color', 'blue');
      $('li.item-c').parents().css('background-color', 'yellow');
    });
  </script>
</head>
<body>
  <ul class="level-1">
    <li class="item-i">I
      <span>자식노드1 </span>
      <p>자식노드2 </p>
    </li>
    <li class="item-ii">II
      <ul class="level-2">
        <li class="item-a">A
        <li class="item-b">B
          <ul class="level-3">
            <li class="item-1">1 </li> <li class="item-2">2 </li> <li class="item-3">3 </li>
          </ul>
        </li>
        <li class="item-c">C </li>
      </ul>
    </li>
    <li class="item-iii">III </li>
  </ul>
</body>
```

## 6. 문서 객체 선택과 탐색

- 형제 노드 찾기 – prev(), prevAll(), next(), nextAll()

```
<head>
<script src="https://code.jquery.com/jquery-3.6.0.js"></script>
<script>
$(document).ready(function () {
    $('.item-b').prev().css('border', '1px solid red');
    $('.item-c').prevAll().css('background-color', 'pink');
    $('.item-b').next().css('border', '1px solid green');
    $('.item-a').nextAll('li.item-c').css('color', 'coral');
    //$($('.item-a').nextAll()).css('color', 'violet');
});
</script>
</head>
```



```
<body>
<ul class="level-1">
  <li class="item-ii">II
    <ul class="level-2">
      <li class="item-a">A
      <li class="item-b">B
        <ul class="level-3">
          <li class="item-1">1 </li>
          <li class="item-2">2 </li>
          <li class="item-3">3 </li>
        </ul>
      </li>
      <li class="item-c">C </li>
    </ul>
  </li>
  <li class="item-iii">III </li>
</ul>
</body>
```

# 7. 문서 객체 조작

- jQuery 문서 객체 조작 관련 메소드

메소드	설명
addClass()	문서 객체의 클래스 속성을 추가
removeClass()	문서 객체의 클래스 속성을 제거
attr()	속성과 관련된 모든 기능 수행
removeAttr()	문서 객체의 속성 제거
css()	스타일과 관련된 모든 기능 수행
html()	문서 객체 내부의 글자와 관련된 모든 기능 수행(HTML태그 인식)
text()	문서 객체 내부의 글자와 관련된 모든 기능 수행(HTML태그 무시)
remove()	문서 객체 제거
empty()	문서 객체 내부를 비움
\$()	문서 객체 생성
clone()	문서 객체 복제
val()	폼 요소의 value값을 가져오거나 새 값 적용

# 7. 문서 객체 조작

---

- 클래스 속성 추가
    - `addClass(className)`
    - `addClass(function(index, currentClassName))`
- 

```
<head>
  <meta charset="utf-8">
  <title>addClass demo</title>
  <style>
    p { margin: 8px; font-size: 16px; }
    .selected { color: red; }
    .highlight { background: yellow; }
  </style>
  <script src="https://code.jquery.com/jquery-3.7.0.js"> </script>
  <script>
    $(function () {
      $("p").last().addClass(["selected", "highlight"]); //$( "p" ).last().addClass( "selected highlight" );
    });
  </script>
</head>
<body>
  <p>Hello</p> <p>and</p> <p>Goodbye</p>
</body>
```

---



# 7. 문서 객체 조작

- 클래스 속성 추가
  - `addClass(className)`
  - `addClass(function(index, currentClassName)`

```
<p>Hello</p>
<p class="red green">and</p>
<p>Goodbye</p>
<div>There is one green p</div>
```

```
<head>
  <meta charset="UTF-8">
  <title>addClass demo</title>
  <style>
    p { margin: 8px; font-size: 16px; }
    .red { color: red; }
    .red.green { background: gold; }
  </style>
  <script src="https://code.jquery.com/jquery-3.7.0.js"></script>
  <script>
    $(function () {
      $("p").addClass(function (index, currentClass) {
        let addedClass;

        if (currentClass === "red") {
          addedClass = "green";
          $("div").text("There is one green p");
        }
        return addedClass;
      });
    });
  </script>
</head>
<body>
  <p>Hello</p> <p class="red">and</p> <p>Goodbye</p> <div></div>
</body>
```

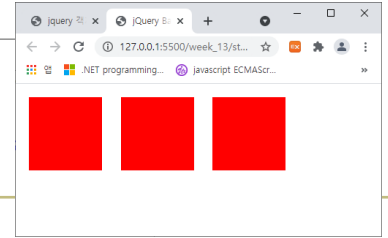
# 7. 문서 객체 조작

- 문서 스타일 추가

- `$(선택자).css(name, value);`
- `$(선택자).css(name, function(index, style){});`
- `$(선택자).css(object);`

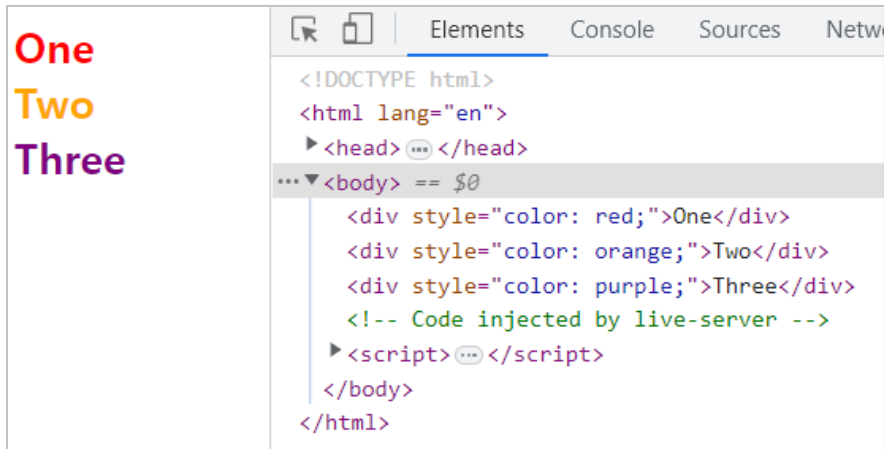
```
$('.box').css({  
  float: 'left',  
  margin: '10px',  
  width: '100px',  
  height: '100px',  
  backgroundColor: 'red',  
});
```

```
<head>  
  <title>jQuery Basic</title>  
  <script src="https://code.jquery.com/jquery-3.7.0.js"> </script>  
  <script>  
    $(function () {  
      // 스타일을 적용합니다.  
      $('.box').css('display', 'inline-block');  
      $('.box').css('margin', 10);  
      $('.box').css('width', 100);  
      $('.box').css('height', 100);  
      $('.box').css('backgroundColor', 'red');  
    });  
  </script>  
</head>  
<body>  
  <div class="box"> </div>  
  <div class="box"> </div>  
  <div class="box"> </div>  
</body>
```



# 7. 문서 객체 조작

- 문서 스타일 추가
  - `$(선택자).css(name, value);`
  - `$(선택자).css(name, function(index, style){});`
  - `$(선택자).css(object);`



```
<head>
  <style>
    div {
      font-size: 1.5em;
      font-weight: bold;
    }
  </style>
  <script src="https://code.jquery.com/jquery-3.7.0.js"> </script>
  <script>
    $(function () {
      const color = ['red', 'orange', 'purple'];
      $('div').css('color', function (index) {
        return color[index];
      });
    });
  </script>
</head>
<body>
  <div>One</div>
  <div>Two</div>
  <div>Three</div>
</body>
```

# 7. 문서 객체 조작

- 문서 스타일 추가

- `$(선택자).css(name, value);`
- `$(선택자).css(name, function(index, style){});`
- `$(선택자).css(object);`



```
<style>
  div { width: 20px; height: 15px; background-color: #f33; }
</style>
<script src="https://code.jquery.com/jquery-3.7.0.js"> </script>
<script>
  $(function () {
    $("div").on("click", function () {
      $(this).css({
        width: function (index, value) {
          return parseFloat(value) * 1.2;
        },
        height: function (index, value) {
          return parseFloat(value) * 1.2;
        }
      });
    });
  });
</script>
</head>
<body>
  <div>click</div> <hr> <div>click</div>
</body>
```

# 7. 문서 객체 조작

- 문서 객체 속성 조작

- `$(선택자).attr(name);`
  - 첫 번째 요소의 attribute name에 일치 값 반환
- `$(선택자).attr(name, function);`
- `$(선택자).attr(name, value);`

☒ Check me  
.attr( 'checked' ): checked  
.prop( 'checked' ): true

☐ Check me  
.attr( 'checked' ): checked  
.prop( 'checked' ): false

```
<head>
  <meta charset="UTF-8">
  <title>attr demo</title>
  <style>
    p { margin: 20px 0 0; }
    strong { color: blue; }
  </style>
  <script src="https://code.jquery.com/jquery-3.7.0.js"> </script>
  <script>
    $(function () {
      $("input")
        .change(function () {
          let $input = $(this);
          $('p').html(
            <strong> ${$input.attr("checked")} </strong> <br>
            .prop( 'checked' ): <strong> ${$input.prop("checked")} </strong> <br>`);
        })
    });
  </script>
</head>
<body>
  <input id="check1" type="checkbox" checked="checked">
  <label for="check1">Check me</label>
  <p> </p>
</body>
```

To retrieve and change DOM properties such as the checked, selected, or disabled state of form elements, use the `.prop()` method.

해당 속성이 없으면 undefined 반환



```
<strong> ${$input.attr("checked")} </strong> <br>
.prop( 'checked' ): <strong> ${$input.prop("checked")} </strong> <br>`);
```

# 7. 문서 객체 조작

- 문서 객체 속성 조작
  - `$(선택자).attr(name);`
  - `$(선택자).attr(name, function);`
  - `$(선택자).attr(name, value);`

```
<head>
  <meta charset="UTF-8">
  <title>attr demo</title>
  <style>
    div { color: blue; }
    span { color: red; }
    strong { font-weight: bolder; }
  </style>
  <script src="https://code.jquery.com/jquery-3.7.0.js"> </script>
  <script>
    $(document).ready(function () {
      $("div")
        .attr("id", function (index) { return "div-id" + index; })
        .each(function () {
          $("span", this).html("(id = '<strong>' + this.id + '</strong>')");
        });
    });
  </script>
</head>
<body>
  <div>Zero-th <span> </span> </div>
  <div>First <span> </span> </div>
  <div>Second <span> </span> </div>
</body>
```

Zero-th (id = 'div-id0')

First (id = 'div-id1')

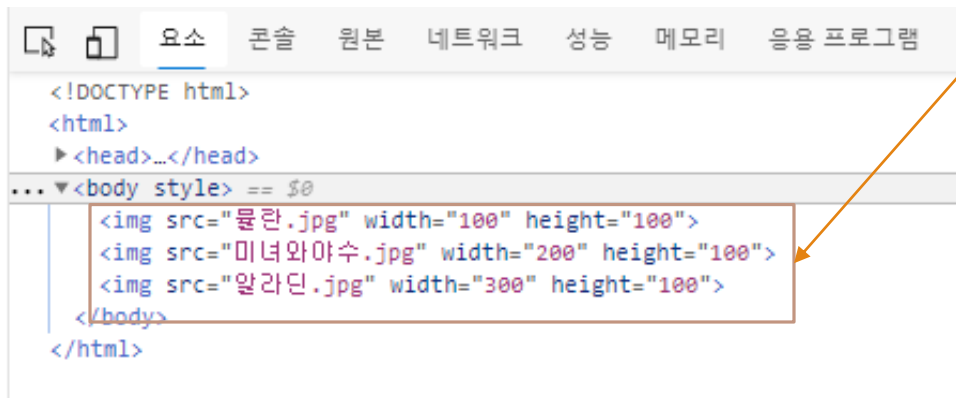
Second (id = 'div-id2')

# 7. 문서 객체 조작

- 문서 객체 속성 조작
  - `$(선택자).attr(name);`
  - `$(선택자).attr(name, function);`
  - `$(선택자).attr(name, value);`

```
$('#img').attr('width', '100')  
$('#img').attr('height', '100')
```

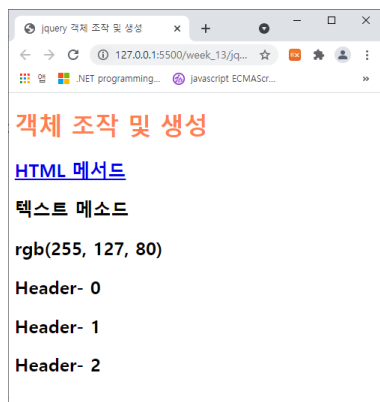
축약 => `$('#img').attr({width : 100, height:100})`



```
<!DOCTYPE html>  
<html>  
<head>  
  <script src="https://code.jquery.com/jquery-3.6.0.js"></script>  
  <script>  
    $(document).ready(function () {  
      $('#img').attr({  
        width: function (index) {  
          return (index + 1) * 100;  
        },  
        height: 100  
      });  
    });  
  </script>  
</head>  
<body>  
    
    
    
</body>  
</html>
```

# 7. 문서 객체 조작

- 문서 객체 내부 검사
  - html() 메소드 : 자바스크립트에서 innerHTML 속성과 관련(HTML요소)
    - \$(선택자).html(value);
    - \$(선택자).html(function(index, html){});
  - text() 메소드 : 자바스크립트에서 textContent 속성과 관련(텍스트 내용)
    - \$(선택자).text(value);
    - \$(선택자).text(function(index, text){});



```
<body>
  <h1>객체 조작 및 생성</h1>
  <h2><em>html()</em></h2>
  <h2>text()</h2>
  <h2></h2>
  <div></div>
  <div></div>
  <div></div>
</body>
```

```
<head>
  <meta charset="UTF-8">
  <title>jquery 객체 조작 및 생성</title>
  <script src="https://code.jquery.com/jquery-3.7.0.js"></script>
</script>
$(function () {
  $('h1').css('color','coral');
  $('h2:nth-child(2)').html('<a href="#">HTML 메서드</a>');
  $('h2:nth-child(3)').text('텍스트 메소드');
  $('h2:nth-child(4)').text( $("h1").css("color") );
  $('div').html(function(index){
    return '<h2>Header- ${index} </h2>';
  });
});
</script>
</head>
```

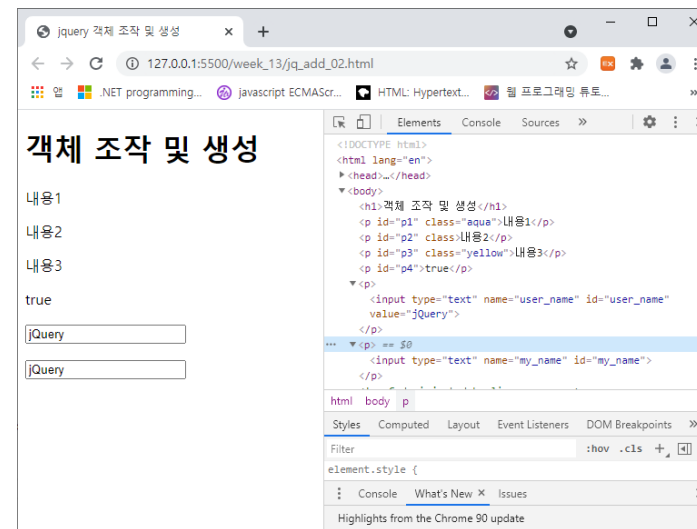


# 7. 문서 객체 조작

- removeClass(), val() 메소드
  - removeClass() 메소드의 매개변수에 아무것도 입력하지 않으면 문서 객체의 모든 클래스를 제거

```
<head>
<meta charset="UTF-8">
<title>jquery 객체 조작 및 생성</title>
<script src="https://code.jquery.com/jquery-3.7.0.js"></script>
<script>
$(document).ready(function () {
    $("#p1").addClass("aqua");
    $("#p2").removeClass("red");
    $("#p4").text($("#p3").hasClass("yellow"));
    $("#my_name").val($("#user_name").val());
});
</script>
</head>

<body>
<h1>객체 조작 및 생성</h1>
<p id="p1">내용1</p>
<p id="p2" class="red">내용2</p>
<p id="p3" class="yellow">내용3</p>
<p id="p4"></p>
<p><input type="text" name="user_name" id="user_name" value="jQuery"/></p>
<p><input type="text" name="my_name" id="my_name" /></p>
</body>
```

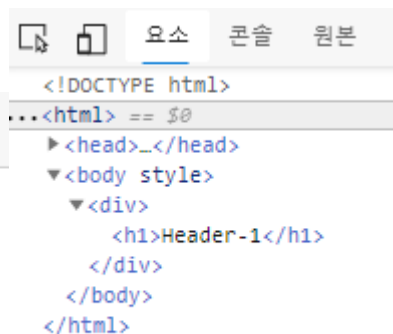


# 7. 문서 객체 조작

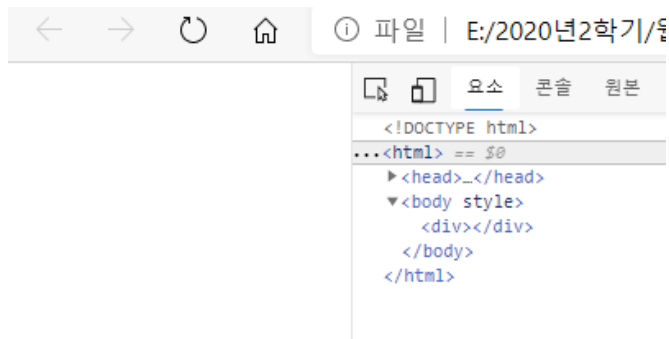
- 문서 객체 제거
  - empty() 메소드 – 특정 문서 객체의 자식 삭제, **매개변수 없음**

```
<!DOCTYPE html>
<html>
<head>
  <script src="https://code.jquery.com/jquery-3.6.0.js"></script>
  <script>
    $(document).ready(function () {
      $('h1').first().remove();
    });
  </script>
</head>
<body>
  <div>
    <h1>Header-0</h1>
    <h1>Header-1</h1>
  </div>
</body>
</html>
```

Header-1



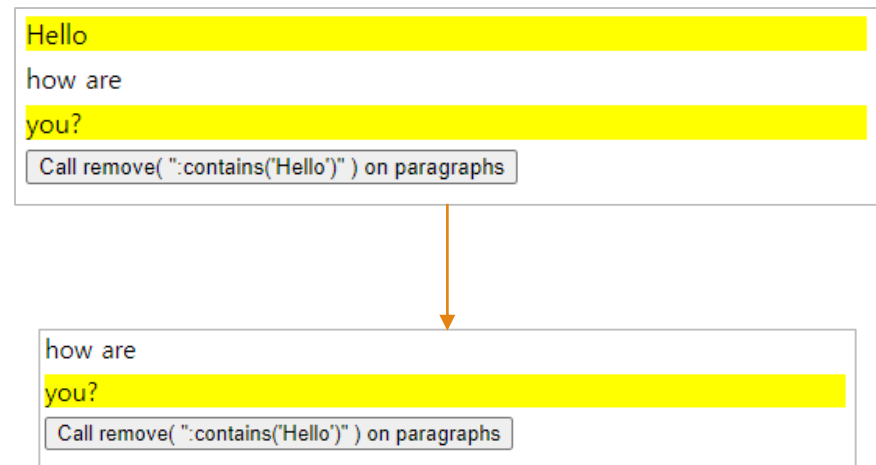
```
<!DOCTYPE html>
<html>
<head>
  <script src="https://code.jquery.com/jquery-3.6.0.js"></script>
  <script>
    $(document).ready(function () {
      $('div').empty();
    });
  </script>
</head>
<body>
  <div>
    <h1>Header-0</h1>
    <h1>Header-1</h1>
  </div>
</body>
</html>
```



# 7. 문서 객체 조작

- 문서 객체 제거
  - remove() 메소드 – 특정 문서 객체 삭제

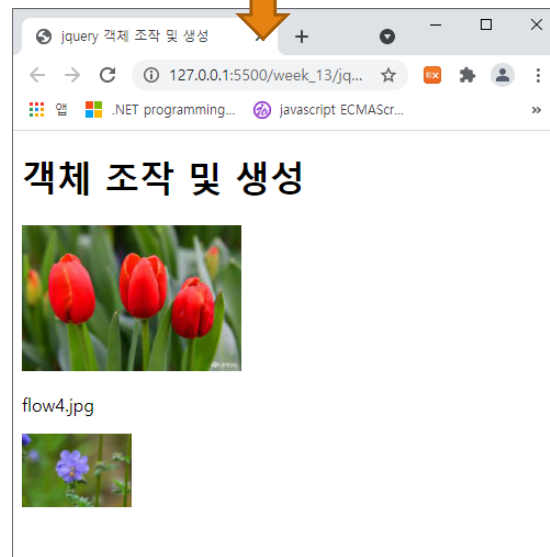
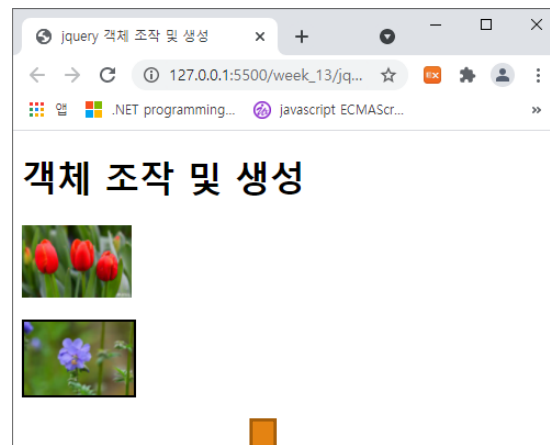
```
<head>
  <meta charset="utf-8">
  <title>remove demo</title>
  <style>
    p {
      background: yellow;
      margin: 6px 0;
    }
  </style>
  <script src="https://code.jquery.com/jquery-3.7.0.js"> </script>
</head>
<body>
  <p class="hello">Hello</p>
  how are
  <p>you?</p>
  <button>Call remove( ":contains('Hello') " ) on paragraphs</button>
  <script>
    $("button").click(function () {
      $("p").remove(":contains('Hello')");
    });
  </script>
</body>
```



# 7. 문서 객체 조작

- 문서 객체 속성 제거
  - removeAttr(attributeName) 메소드 – attributeName과 일치하는 속성 제거

```
<head>
  <meta charset="UTF-8">
  <title>jquery 객체 조작 및 생성</title>
  <script src="https://code.jquery.com/jquery-3.7.0.js"></script>
  <script>
    $(document).ready(function () {
      $("#txt").text($("#wrap img").attr("src")); //속성 검사
      $("#wrap img").attr("width", "200"); //속성 추가
      $("#box img").removeAttr("border"); //속성 제거
    });
  </script>
</head>
<body>
  <h1>객체 조작 및 생성</h1>
  <p id="wrap">
    
  </p>
  <p id="txt"></p>
  <p id="box">
    
  </p>
</body>
```



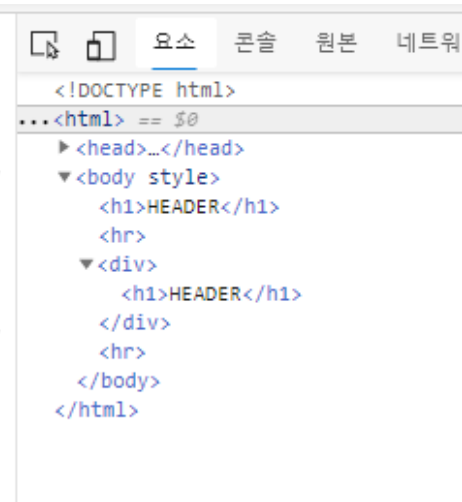
# 7. 문서 객체 조작

- clone() 메소드 : 문서 객체를 복제

```
<!DOCTYPE html>
<html>
<head>
  <script src="https://code.jquery.com/jquery-3.7.0.js"></script>
  <script>
    $(document).ready(function () {
      $('div').append($('h1').clone());
    });
  </script>
</head>
<body>
  <h1>HEADER</h1>
  <hr/>
  <div></div>
  <hr/>
</body>
</html>
```

HEADER

HEADER



# 7. 문서 객체 조작 - 요소 스타일 & 속성 값 가져오기

- 스타일 값 구하기
  - `$(selector).css("스타일 속성 이름")`
  - `$(selector).css(["스타일 속성 이름" , "스타일 속성 이름" ....])`
- 속성 값 구하기
  - `$(selector).attr("속성 이름")`
- 내용 문자열로 읽기
  - `$(selector).html()`
  - `$(selector).text()`

```
<ul class="menu">
  <li>web_menu1 </li>
  <li>web_menu2 </li>
  <li>web_menu3 </li>
  <li>web_menu4 </li>
  <li>web_menu5 </li>
</ul>
<div class="header">
  
</div>
```

```
font-size= 12px
width = 250px, height = 135px
src = http://placekitten.com/150/150

.menu.html() =
  <li>web_menu1</li>
  <li>web_menu2</li>
  <li>web_menu3</li>
  <li>web_menu4</li>
  <li>web_menu5</li>

.menu li = web_menu2
```

```
<script>
$(document).ready(function () {
  // 메뉴 아이템의 폰트 사이즈 정보 출력
  console.log(`font-size= ${$("ul.menu li").css("font-size")}`);

  // 메뉴의 너비와 높이 정보 구하기
  const info = $("ul.menu").css(["width", "height"]);

  // 너비와 높이 정보 출력 하기
  console.log(`width = ${info.width}, height = ${info.height}`);

  // img 태그의 src 속성 값 출력
  console.log(`src = ${$(".header img").attr("src")}`);

  //ul.menu 노드의 내부 구조를 문자열로 출력
  console.log(`.menu.html() = ${$(".menu").html()}`);

  //ul.menu노드의 두번째 자식노드 내용을 문자열로 출력
  console.log(`.menu li = ${$(".menu").children().eq(1).text()}`);

});
</script>
```

# 7. 문서 객체 조작 - 요소 크기 다루기

---

- 요소 크기 가져오기
  - 기본 크기
    - `$(selector).width()`
    - `$(selector).height()`
  - 기본 크기 + padding
    - `$(selector).innerWidth()`
    - `$(selector).innerHeight()`
  - 기본 크기 + padding + border
    - `$(selector).outerWidth()`
    - `$(selector).outerHeight()`
  - 기본 크기 + padding + border + margin
    - `$(selector).outerWidth(true)`
    - `$(selector).outerHeight(true)`
- 요소 크기 변경
  - 기본 크기
    - `$(selector).width(value)`
    - `$(selector).height(value)`
  - 기본 크기 + padding
    - `$(selector).innerWidth(value)`
    - `$(selector).innerHeight(value)`

# 7. 문서 객체 조작 - 요소 크기 다루기

- 요소 크기 다루기

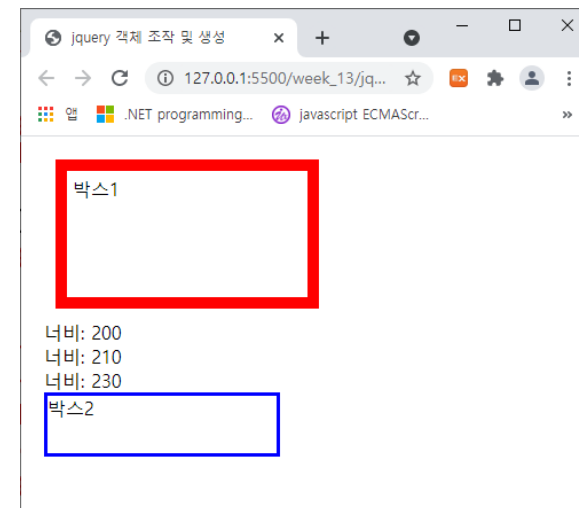
```
<head>
  <meta charset="UTF-8">
  <title>jquery 객체 크기 다루기</title>
  <script src="https://code.jquery.com/jquery-3.6.0.js"> </script>
  <script>
    $(function () {
      const w1 = $("#box1").width();
      $("#t1").text("너비: " + w1);

      const w2 = $("#box1").innerWidth();
      $("#t2").text("너비: " + w2);

      const w3 = $("#box1").outerWidth();
      $("#t3").text("너비: " + w3);

      $("#box2").width("200").height("50")
        .css("border", "3px solid blue");
    });
  </script>
```

```
<style type="text/css">
  * {
    margin: 0;
    padding: 0;
  }
  body {
    margin: 20px;
  }
  #box1 {
    width: 200px;
    height: 100px;
    padding: 5px;
    border: 10px solid red;
    margin: 10px;
  }
</style>
</head>
<body>
  <p id="box1">박스1 </p>
  <p id="t1"> </p>
  <p id="t2"> </p>
  <p id="t3"> </p>
  <p id="box2">박스2 </p>
</body>
```





## 8. 문서 객체 삽입

- jQuery 문서 객체 삽입 메소드

메소드	설명
<code>\$(A).appendTo(B)</code>	A를 B의 내부의 뒤 부분에 추가함(A가 B의 자식이 됨)
<code>\$(A).append(B)</code>	B를 A의 내부의 뒤 부분에 추가함 (B가 A의 자식이 됨)
<code>\$(A).prepend(B)</code>	B를 A의 내부의 앞 부분에 추가함 (B가 A의 자식이 됨)
<code>\$(A).prependTo(B)</code>	A를 B의 내부의 앞 부분에 추가함 (A가 B의 자식이 됨)
<code>\$(A).after(B)</code>	B를 A의 뒤에 추가함 (B가 A의 형제가 됨)
<code>\$(A).before(B)</code>	B를 A의 앞에 추가함 (B가 A의 형제가 됨)
<code>\$(A).insertAfter(B)</code>	A를 B 뒤에 추가함 (A가 B의 형제가 됨)
<code>\$(A).insertBefore(B)</code>	A를 B의 앞에 추가함 (A가 B의 형제가 됨)
<code>\$(새요소).replaceAll(요소선택)</code> <code>\$(요소선택).replaceWith(새요소)</code>	선택한 요소를 새 요소로 교체

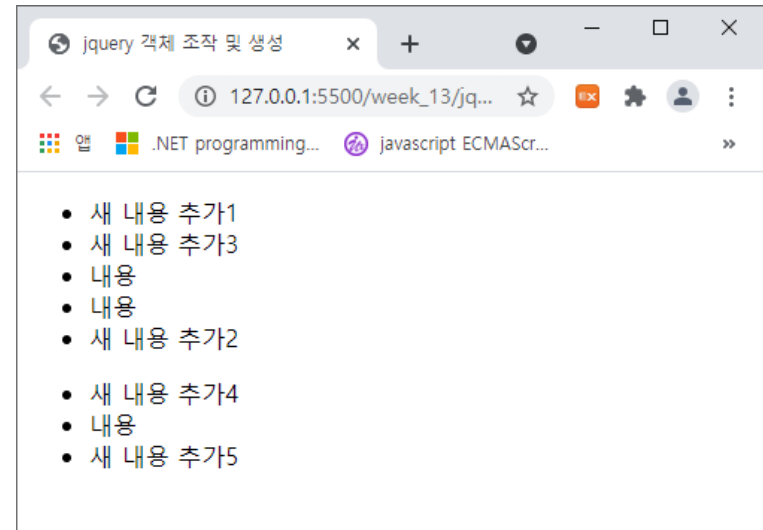
# 8. 문서 객체 삽입

- 문서 객체 삽입

```
<head>
  <meta charset="UTF-8">
  <title>jquery 객체 조작 및 생성</title>
  <script src="https://code.jquery.com/jquery-3.7.0.js"> </script>
  <script>
    $(document).ready(function () {
      $("#myList").before("<li>새 내용 추가1</li>");
      $("#myList").after("<li>새 내용 추가2</li>");

      $("<li>새 내용 추가3</li>").insertBefore("#myList");
      const copy_obj = $("#myList").clone();
      $(copy_obj).insertAfter("#myList");

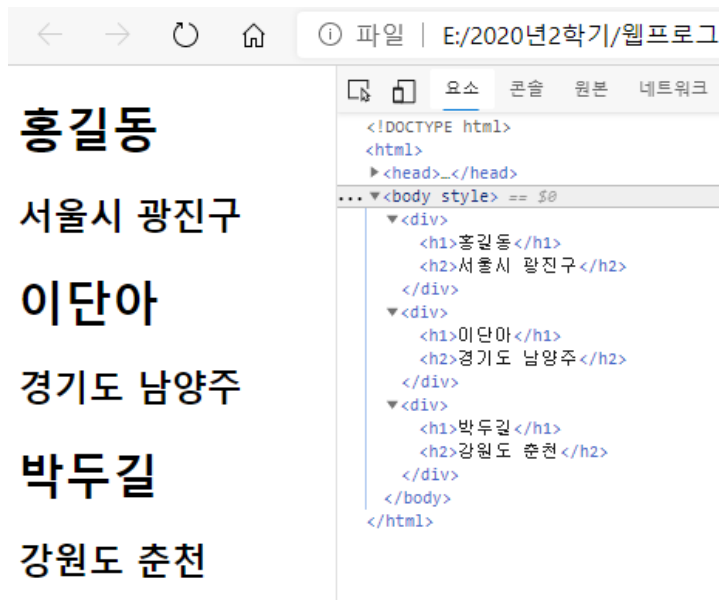
      $("#listZone").append("<li>새 내용 추가5</li>");
      $("<li>새 내용 추가4</li>").prependTo("#listZone");
    });
  </script>
</head>
```



```
<body>
  <ul>
    <li id="myList">내용</li>
  </ul>
  <ul id="listZone">
    <li>내용</li>
  </ul>
</body>
```

# 8. 문서 객체 삽입

- append() 메소드 응용



```
<!DOCTYPE html>
<html>
<head>
  <script src="https://code.jquery.com/jquery-3.6.0.js"> </script>
  <script>
    $(document).ready(function () {
      var content = [
        { name: '홍길동', region: '서울시 광진구' },
        { name: '이단아', region: '경기도 남양주' },
        { name: '박두길', region: '강원도 춘천' }
      ];

      $('div').append(function (index) {
        // 변수를 선언합니다.
        var item = content[index];
        var output = '';
        output += '<h1>' + item.name + '</h1>';
        output += '<h2>' + item.region + '</h2>';

        return output;
      });
    });
  </script>
</head>
<body>
  <div></div>
  <div></div>
  <div></div>
</body>
</html>
```

# 8. 문서 객체 삽입

- appendTo() 메소드를 이용하여 2초 간격으로 이미지 변경



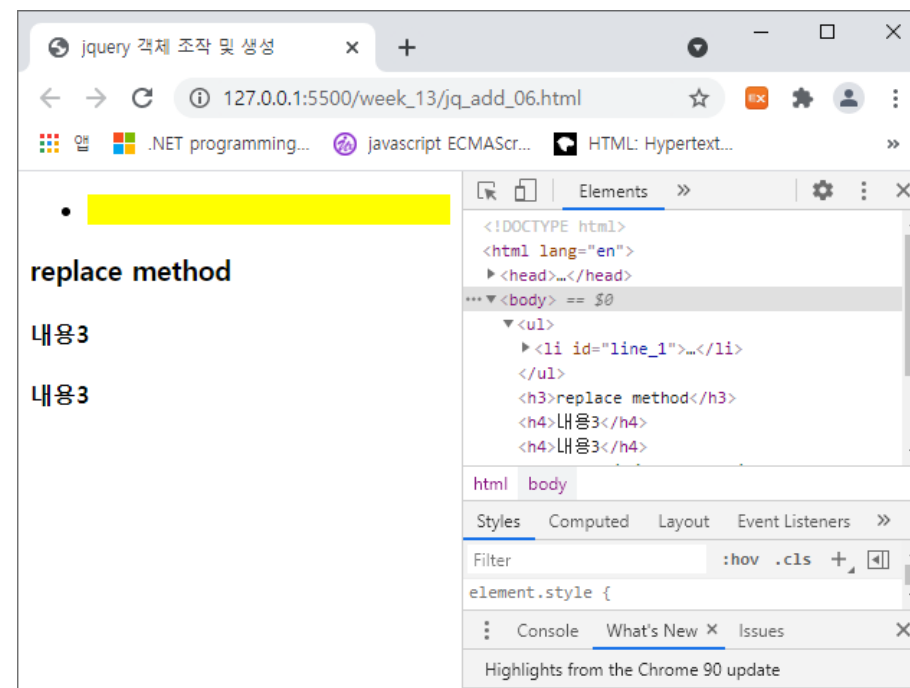
```
<!DOCTYPE html>
<html>
<head>
  <script src="https://code.jquery.com/jquery-3.6.0.js"></script>
  <script>
    $(document).ready(function () {
      // .image의 크기를 조정합니다.
      $('img').css('width', 150);

      // 함수를 2초마다 실행합니다.
      setInterval(function () {
        // 첫 번째 이미지를 마지막으로 보냅니다.
        $('img').first().appendTo('body');
      }, 2000);
    });
  </script>
</head>
<body>
  
  
  
  
  
</body>
</html>
```

# 8. 문서 객체 삽입

## • 문서 객체 교체

```
<head>
<meta charset="UTF-8">
<title>jquery 객체 조작 및 생성</title>
<script src="https://code.jquery.com/jquery-3.7.0.js"> </script>
<script>
$(document).ready(function () {
    $("#line_1").empty();
    $("#line_2").remove();
    $("h2").replaceWith("<h3>replace method</h3>");
    $("<h4>내용3</h4>").replaceAll("p");
});
</script>
<style type="text/css">
#line_1 {
    background-color: yellow;
}
#line_2 {
    background-color: orange;
}
</style>
</head>
```



```
<body>
<ul>
    <li id="line_1">내용1 </li>
    <li id="line_2">내용2 </li>
</ul>
<h2>교체 메서드</h2>
<p>내용1 </p>
<p>내용2 </p>
</body>
```

# Q & A

---

- jQuery 기본 개념과 문서객체에 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.
- 다음 시간에는 jQuery 이벤트와 효과에 대하여 알아보도록 하겠습니다
- 수고하셨습니다.^^