

CH 09

제이쿼리기초 다지기

1. 제이쿼리 개요
2. 제이쿼리 함수
3. 제이쿼리 선택자 요약

>> 학습목표 <<

- ❖ 제이쿼리 특성과 연동 방식을 이해한다.
- ❖ 제이쿼리 함수 명세 방법을 살펴본다.
- ❖ 다양한 제이쿼리 선택자의 종류와 사용 방법을 알아본다.

1.1 제이쿼리

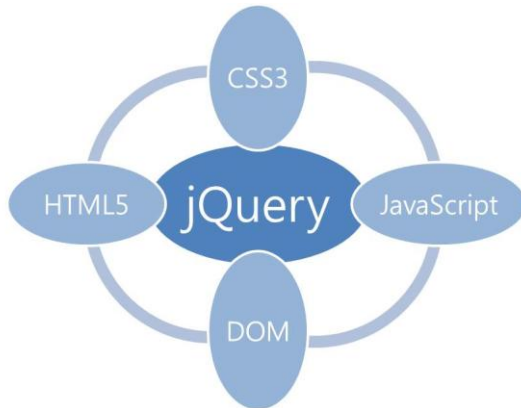
● 제이쿼리(jQuery)

- HTML5 웹 문서 안의 스크립트 언어를 단순화하도록 설계된 자바스크립트 라이브러리
- 존 레식(John Resig)이 2006년 자바스크립트를 쉽게 이용할 목적으로 제안

● 특성

- 가장 인기 있는 자바 스크립트 라이브러리
- 크로스 브라우징(cross browsing) 지원
- 생산성 향상 지원

● 제이쿼리 관련 기술



1.2 자바스크립트

● 자바스크립트 명세 방법

- 자바스크립트 코드는 <script> 태그 안에 명세
- <head> 태그 또는 <body> 태그 안에 포함시키거나 '<script src="sample.js">'을 통해 별도의 자바스크립트 파일에 명세할 수도 있음
- 자바스크립트 코드는 명세된 순서대로 실행

```
<html>
  <head>
    ... 생략 ...
    <script src="sample.js">
    <script>
      자바스크립트 코드2
    </script>
  </head>
  <body>
    ... 생략 ...
    <script>
      자바스크립트 코드3
    </script>
    ... 생략 ...
  </body>
</html>
```

```
// sample.js
```

```
자바스크립트 코드1
```

자바스크립트 실행하기

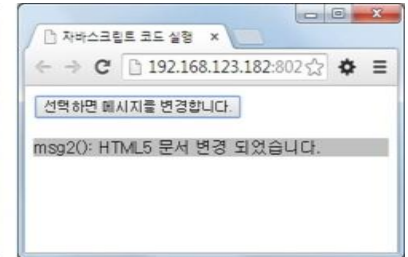
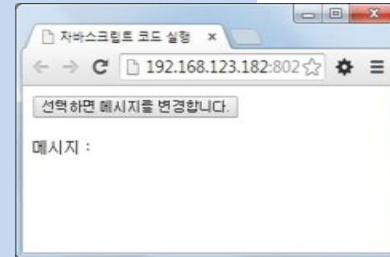
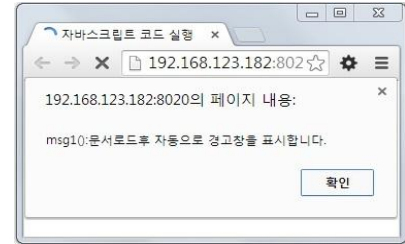
● 자바스크립트 활용 예

[예제9-1] 자바스크립트 실행하기

/ch09/javascript.html

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8"/>
  <title>자바스크립트 코드 실행</title>
  <script>
    function init() {
      msg1();
    }
    function msg1() {
      alert('msg1():문서로드후 자동으로 경고창을 표시합니다.');
```



1.3 제이쿼리 연동

● 제이쿼리 연동 방식

- HTML5 문서 안에 사용할 제이쿼리 라이브러리 파일을 포함하도록 선언
- 제이쿼리 라이브러리 파일의 URL 경로를 <script> 태그로 간단히 삽입
- 제이쿼리 모바일 방식과 동일
- CDN(Content Delivery Network) 연결 방식
 - 제이쿼리 사이트에서 실시간으로 다운로드
 - 'jquery-1.6.4.min.js' 파일은 앞으로 사용할 제이쿼리 라이브러리를 포함
 - 항상 인터넷과 연결되어 있어야 한다는 제약, 최신 버전의 라이브러리 사용 가능

```
<!DOCTYPE html>
<html>
<head>
  <script src="http://code.jquery.com/jquery-1.6.4.min.js"></script>
  <script>
    /* 자바스크립트나 제이쿼리 코드를 명세하는 영역 */
  </script>
</head>
<body>
  ... 생략 ...
</body>
</html>
```

제이쿼리 실행하기

● 제이쿼리 활용 예제

[예제9-2] 제이쿼리 실행하기

/ch09/jquery.html

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8"/>
  <title>제이쿼리 코드 실행</title>
  <script src="http://code.jquery.com/jquery-1.6.4.min.js"> </script>
  <script>
    $(document).ready( function() {
      msg1();
      $('#id1').click( function(){
        msg2();
      });
    });
    function msg1() {
      alert('msg1():문서로드후 자동으로 경고 창을 표시합니다.');
```

```

    }
    function msg2() {
      $('#id2').css('background','silver');
      $('#id2').text('msg2():HTML문서 변경되었습니다.');
```

```

    }
  </script>
</head>
<body>
  <button id="id1">선택하면 메시지를 변경합니다.</button> <br/> <br/>
  <div id="id2">메시지 : </div>
</body>
</html>

```

제이쿼리와 자바스크립트 비교

● [표 9-1] 자바스크립트와 제이쿼리 코드 비교

자바스크립트 코드	제이쿼리 코드
필요 없음	<code><script src="http://code.jquery.com/jquery-1.6.4.min.js"> </script></code>
<pre>function init() { msg1(); } ... 생략 ... <body onLoad="init();"> ... 생략 ...</pre>	<pre>\$(document).ready(function() { msg1(); }); ... 생략 ... <body> ... 생략 ...</pre>
<pre>... 생략 ... <button onClick="msg2();"> ... 생략 ...</pre>	<pre>\$('#id1').click(function(){ msg2(); }); ... 생략 ... <button id="id1"> ... 생략 ...</pre>
<pre>function msg2() { document.getElementById('id2').innerHTML = '<div style="background:silver"> msg2():HTML문서 변경되었습니다.</div>'; } ... 생략 ... <div id="id2"> ... 생략 ...</pre>	<pre>function msg2() { \$('#id2').html('<div style="background:silver"> msg2(): HTML문서 변경되었습니다.</div>'); } ... 생략 ... <div id="id2"> ... 생략 ...</pre>
	<pre>function msg2() { \$('#id2').css('background','silver'); \$('#id2').text('msg2():HTML문서 변경되었습니다.');</pre>
	<pre>... 생략 ... <div id="id2"> ... 생략 ...</pre>

1.4 DOM

- 제이쿼리를 활용하려면 DOM에 대한 이해가 꼭 필요함
- DOM(Document Object Model; 문서 객체 모델)
 - HTML5와 같은 구조화된 문서를 객체 개념으로 표현하고 접근하는 방식
 - W3C가 정한 공식 표준으로 W3C가 표준화한 여러 API의 기반
 - DOM API를 통해 HTML5와 XML 문서 안의 구성 요소 즉, 엘리먼트, 속성, 텍스트 등을 접근하고 변경할 수 있어 문서의 내용이나 구조, 스타일 등을 동적으로 제어 가능
 - 플랫폼과 언어에 독립적이며 동적으로 문서의 내용, 구조, 스타일을 접근하고 변경하는 중요한 수단
- 자바스크립트의 비호환성 문제 발생
 - 웹 브라우저 안에 자바스크립트 엔진이 제조사에 따라 일부 다르게 동작하는 문제
→ 해결 방법의 핵심이 바로 DOM
 - DOM이라는 공통된 표준 모델(표준 API)을 따르면 스크립트 엔진이나 언어가 다르더라도 얼마든지 동일하게 HTML5이나 XML과 같은 구조화된 문서에 접근 가능

DOM 트리 구조

- DOM 트리(DOM tree)

- HTML5이나 XML 문서의 구성 요소들은 계층적인 상호 관계를 갖고 구성
 - DOM 모델에서 각 객체(HTML 문서의 구성 요소)들은 자연스럽게 계층적인 트리 구조로 표현
- 노드 트리(node tree)
 - 3가지 노드 유형으로 표현

1) 엘리먼트 노드

- 태그와 내용으로 표현되는 엘리먼트를 표현하는 노드
- 하위에 또 다른 엘리먼트를 포함

2) 속성 노드

- 엘리먼트의 시작 태그 안에 포함되는 속성 이름과 속성 값을 쌍으로 갖는 노드

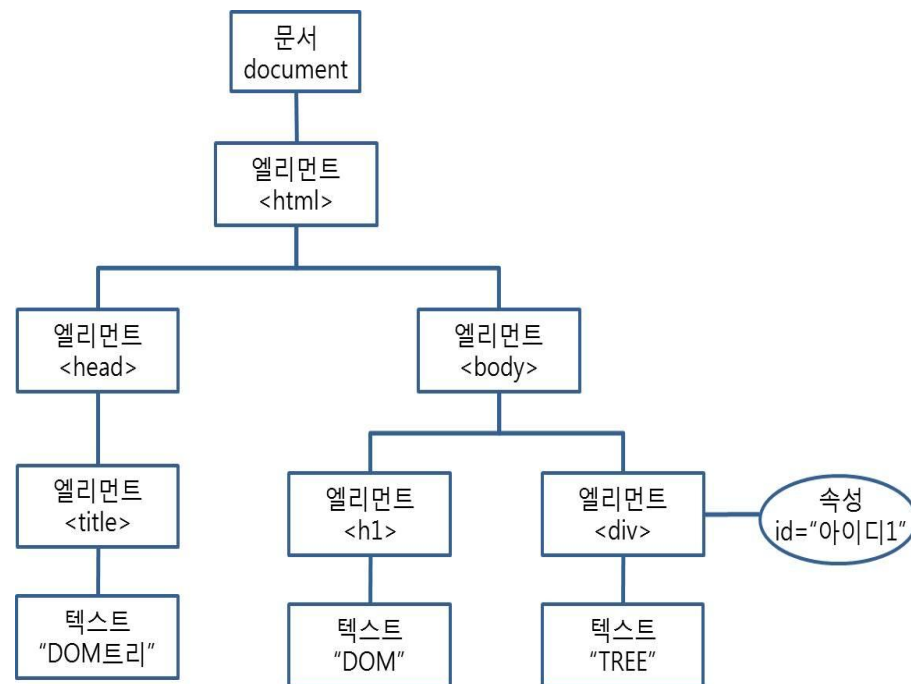
3) 텍스트 노드

- 엘리먼트가 갖는 실제 내용을 의미하는 노드
- DOM 트리에서 최하위 단말 노드가 해당

DOM 트리 구조 예

● DOM 트리 구조

```
<!DOCTYPE html>
<html>
  <head>
    <title>DOM트리 </title>
  </head>
  <body>
    <h1>DOM</h1>
    <div id="아이디1">TREE</div>
  </body>
</html>
```



2.1 제이쿼리의 기본 기능

● 제이쿼리의 9가지 기본 기능

- DOM API들을 활용하는 제이쿼리 메소드들의 사용 방법과 밀접하게 관련

분류	기능	지원 메소드	다루는 장
Core	핵심개념	jQuery() 선언 함수 정의 및 활용 방법	9장
Selectors	선택자	DOM 트리의 노드 선택 표현식	
CSS	스타일	CSS 스타일 속성값 변경 메소드	10장
Traversing	탐색	DOM 트리의 계층 구조를 이용한 노드 탐색 메소드	
Manipulation	조작	DOM 트리의 노드 변경 메소드	
Attributes	속성	엘리먼트 속성값의 조회 및 변경 메소드	10장
Events	이벤트	마우스, 키보드, 폼 및 문서 관련 이벤트 메소드	
Effects	효과	동적 스타일 변화를 위한 메소드	11장
Ajax	비동기교환방식	Ajax 관련 메소드	
UI	사용자 인터페이스	사용자 인터페이스용 라이브러리	-

2.2 jQuery() 함수

- 모든 시작은 jQuery() 함수를 통해 이루어짐
- 괄호 앞의 'jQuery' 문자열은 제이쿼리를 사용한다는 일종의 선언
- 자바스크립트 문장과 제이쿼리 문장을 구별하는 역할
- jQuery() 함수의 3가지 입력 인자 유형

- 1) 선택자 입력 인자

```
jQuery('body p');
```

- 2) HTML 입력 인자

```
jQuery('<h1>HTML 동적 추가</h1>').appendTo('body')
```

- 3) 함수 입력 인자

```
jQuery(function( ) { alert("DOM 트리가 생성됨") } );
```

문서 시작 이벤트 핸들러

● 문서 시작 이벤트 핸들러

```
jQuery(document).ready(function(){ ... });
```

- 'document' 선택자는 HTML5 문서 전체를 선택
- ready() 메소드는 HTML5 문서가 웹 브라우저에 표시될 준비가 완료되면 실행되는 이벤트 핸들러
- function() 함수 안에 실행되기 원하는 제이쿼리 코드를 추가
- 웹 브라우저가 HTML5 전체 문서의 DOM 트리를 생성하고 준비가 되었음을 알려주면 function()은 콜백 함수로서 스스로 호출되어 실행
- 제이쿼리 코드를 <head> 태그 안에 많이 명세할 때 문제점
 - <body> 태그 안의 HTML5 엘리먼트 태그들이 DOM 트리를 구성하기 전에 먼저 실행된다는 점 -> 제이쿼리 코드가 처리할 DOM 트리는 구성되지도 않았는데 코드가 먼저 실행될 수 있음
- jQuery(document).ready() 안에 제이쿼리 실행문들을 ';'으로 구분하여 나열하면 DOM 트리 구성이 완료된 후까지 실행이 지연되었다가 자동으로 호출되어 순차적으로 실행 -> C 언어의 main() 함수와 같은 역할

제이쿼리를 포함한 문서를 처음 시작하는 2가지 방법

- `jQuery(document).ready(function(){ ... });`
 - DOM 준비됨(문서 100% 로드 되기 전)
 - 페이지 DOM만 로드 완료된 이후 수행
 - .ready() 메소드가 여러 개 있더라도 순서대로 실행
 - 비교적 첫 페이지 로드 시간이 빠름
- `jQuery(window).load(function(){ ... });`
 - DOM 준비됨 (문서 100% 로드된 후)
 - 페이지 DOM + 리소스 전체(문서에 포함된 이미지 등의 모든 객체)가 모두 로드 완료된 이후 한 번만 수행
 - .load() 메소드가 여러 개 있으면 마지막만 실행
 - 기존 자바스크립트 `window.onload = function(){ ... }` 를 대체
 - 비교적 첫 페이지 로드 시간이 느림

jQuery() 함수 실행하기

● [그림 9-6] jquery-function.html의 실행 결과(예제9-3)

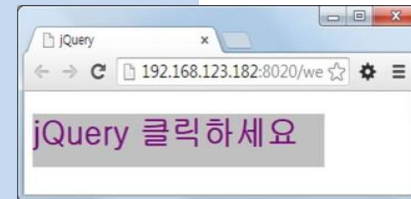
[예제9-3] jQuery() 함수 실행하기

/ch09/jquery-function.html

```

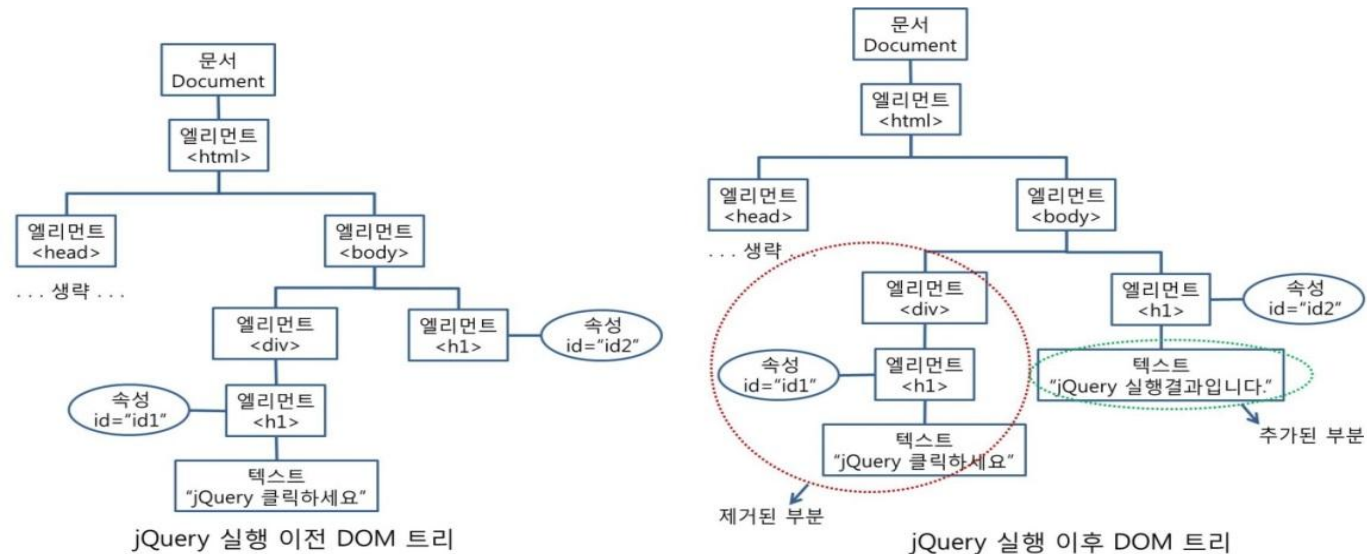
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>jQuery</title>
  <!-- 제이쿼리 라이브러리 선언 -->
  <script src="http://code.jquery.com/jquery-1.6.4.min.js"> </script>
  <!-- 제이쿼리 코드 명세 -->
  <script>
    jQuery(document).ready(function() { // 문서 준비 이벤트 핸들러
      console.log("#1# 문서준비 이벤트 핸들러가 동작했습니다.");
      jQuery('div').click(function ( ) {
        console.log("#2# div 노드 개수 : " + $('div').length);
        console.log("#3# h1 노드 개수 : " + $('h1').length);
        console.log("#4# h1 텍스트 노드 내용 : " + $('h1').text());
        console.log("#5# h1 속성 노드 개수 : " + $('*[id]').length);
        jQuery(this).remove();
        jQuery('#id2').text('jQuery 실행결과입니다. ');
        console.log("#6# div 노드 개수 : " + $('div').length);
        console.log("#7# h1 노드 개수 : " + $('h1').length);
        console.log("#8# h1 텍스트 노드 내용 : " + $('h1').text());
        console.log("#9# h1 속성 노드 개수 : " + $('*[id]').length);
      });
    });
  </script>
  <!-- 스타일 명세 -->
  <style>
    div { width: 300px; height: 50px; color: purple; background-color: silver; }
  </style>
</head>
<body>
  <div><h1 id="id1">jQuery 클릭하세요</h1> </div>
  <h1 id="id2"></h1>
</body>
</html>

```

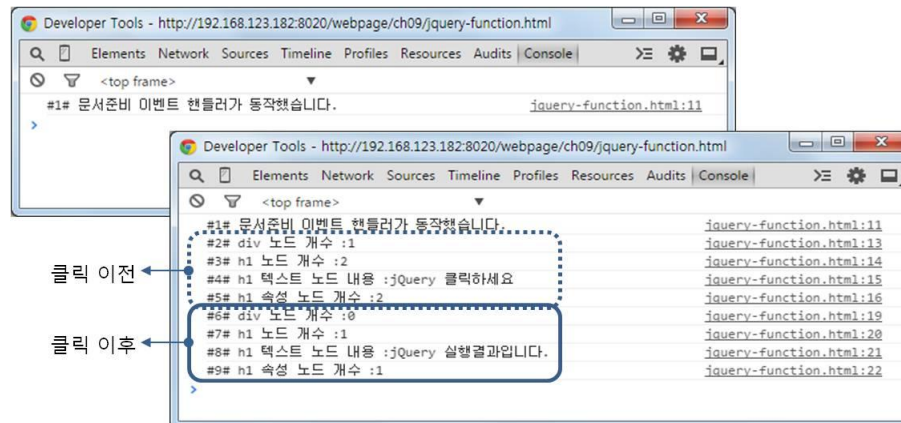


제이쿼리 실행 결과

- [그림 9-7] jquery-function.html의 실행에 따른 DOM 트리 변화(예제9-3)



- [그림 9-8] jquery-function.html의 실행에 따른 콘솔 창의 변화(예제9-3)



jQuery() 함수의 단축형

● \$ 기호

- 'jQuery' 예약어의 별칭(alias)
- 제이쿼리 변수, 함수를 자바 스크립트의 변수, 함수 등과 구별하는 역할

● \$()

- 제이쿼리 함수 또는 제이쿼리 래퍼(wrapper)
- 코드 안 'jQuery()' 함수 식별이 어렵고 불편하기 때문에 줄여서 '\$()'로 표시
- '제이쿼리 객체'와 '제이쿼리 객체 집합'
 - \$(선택자) 형태로 제이쿼리 선택자를 입력 인자로 받아 들여 선택자 조건을 만족하는 엘리먼트 노드들을 DOM 트리에서 찾아 '제이쿼리 객체' 형식으로 반환
 - 일치하는 DOM 노드들이 여러 개이면 배열 형태의 '제이쿼리 객체 집합'을 반환
 - 반환하는 DOM 엘리먼트들을 제이쿼리 객체 개념으로 감싸고 미리 준비된 메소드를 사용할 수 있도록 확장

```
jQuery('div').click();  
→ $('div').click();
```

● 제이쿼리 시작 이벤트 핸들러

```
jQuery( document).ready( function() { . . . . . } );  
→ $( document).ready( function() { . . . . . } );  
→ $( function() { . . . . . } );
```

3.1 제이쿼리 선택자:제이쿼리 실습 예제 템플릿

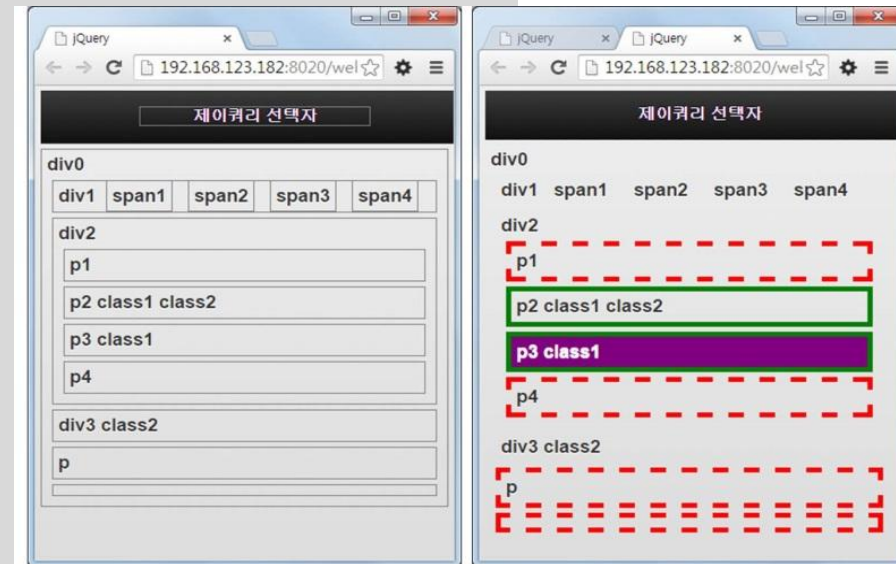
[실습9-1] 제이쿼리 예제 템플릿

/ch09/jq-example-template.html

```

01 <!DOCTYPE html>
02 <html>
03 <head>
04 <meta charset="utf-8">
05 <meta name="viewport" content="width=device-width, initial-scale=1"/>
06 <title>jQuery</title>
07 <link rel="stylesheet" href="http://code.jquery.com/mobile/1.0/jquery.mobile-1.0.min.css" />
08 <script src="http://code.jquery.com/jquery-1.6.4.min.js"></script>
09 <script src="http://code.jquery.com/mobile/1.0/jquery.mobile-1.0.min.js"></script>
10 <style>
11     body * { margin: 5px; padding: 5px; border: solid thin gray; }
12     * { font-size: 18px; font-weight: bold; }
13 </style>
14 <script>
15     $(document).ready(function() {
16         // 제이쿼리 선택자(메소드) 실습 변경 부분
17     });
18 </script>
19 </head>
20 <body>
21 <section data-role="page">
22     <section data-role="header">
23         <h1>제이쿼리 선택자</h1>
24     </section>
25     <div id="div0">div0
26         <div id="div1">div1
27             <span id="span1">span1</span>
28             <span id="span2">span2</span>
29             <span id="span3">span3</span>
30             <span>span4</span>
31         </div>
32         <div id="div2">div2
33             <p id="p1">p1</p>
34             <p id="p2" class="class1 class2">p2 class1 class2</p>
35             <p id="p3" class="class1">p3 class1</p>
36             <p id="p4">p4</p>
37         </div>
38         <div id="div3" class="class2">div3 class2</div>
39         <p>p</p>
40         <p></p>
41     </div>
42 </section>
43 </body>
44 </html>

```



Jq-example-template.html의 실행 결과

basic-selector.html의 적용 결과

3.2 제이쿼리 선택자 유형[1]

● 기본 선택자

- 가장 많이 사용되는 선택자, 형식은 CSS3 선택자와 동일
- \$('선택자')는 선택자 조건을 만족하는 DOM 노드들을 제이쿼리 객체 집합으로 반환
- 제이쿼리 메소드와 제이쿼리 이벤트 등에서 사용하는 기본 요소

형식	기능
*	모든 엘리먼트를 선택 예) \$('*')
태그명	특정 엘리먼트를 모두 선택 예) \$('div'), \$('h1, h3, p')
#아이디	고유한 아이디 속성값을 가진 엘리먼트를 선택 예) \$('#id1'), \$('div#id1')
.클래스	특정 클래스 속성값을 가진 엘리먼트들을 선택 예) \$('.class1'), \$('div.class1'), \$('.class1.class2')

제이쿼리 기본 선택자 실행하기

- [그림 9-10] basic-selector.html의 실행 결과(예제9-4)

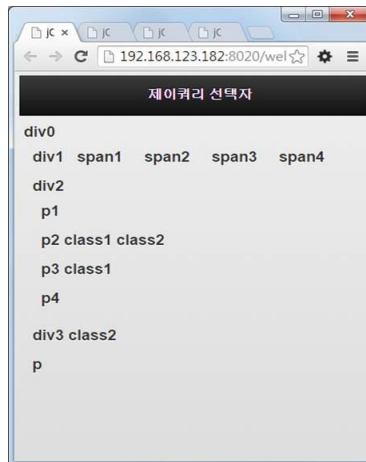
[예제9-4] 제이쿼리 기본 선택자 실행하기

/ch09/basic-selector.html

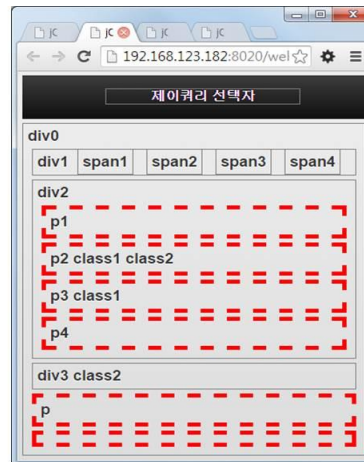
```

$('*').css('border-style', 'hidden');           // (1)
$('p').css('border', 'dashed thick red');       // (2)
$('.class1').css('border', 'solid thick green'); // (3)
$('#p3').css({'background-color':'purple', 'color': 'white'}); // (4)

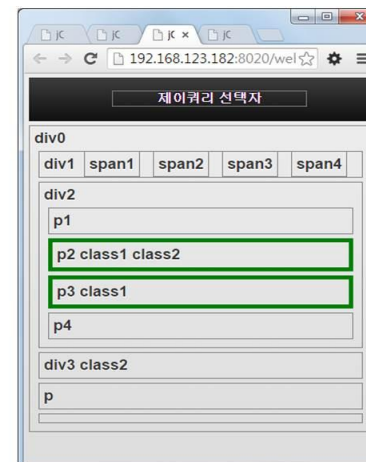
```



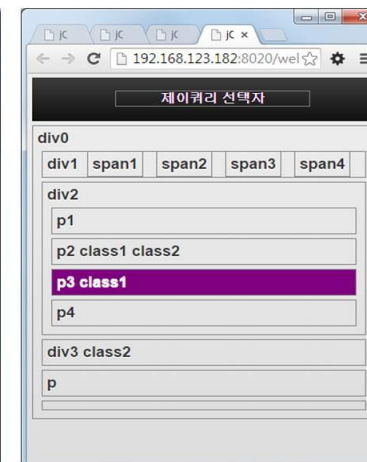
(1)



(2)



(3)



(4)

제이쿼리 선택자 유형[2]

● 계층 선택자

- DOM 트리에서 노드 사이의 상하 관계나 형제 관계를 이용하여 노드를 선택
- 형식은 CSS3 선택자와 동일

형식	기능
선택자 > 자식선택자	특정 엘리먼트 바로 밑에 위치한 하위 엘리먼트(자식 엘리먼트)를 선택 예) \$('ul > li')
선택자 자손선택자	특정 엘리먼트 안에 포함된 모든 하위 엘리먼트(자손 엘리먼트)를 선택 예) \$('div p')
선택자 + 형제선택자	특정 엘리먼트에 바로 다음에 나오는 특정한 형제 엘리먼트 하나를 선택 예) \$('li#id1 + li')
선택자 ~ 형제선택자	특정 엘리먼트 다음에 나오는 특정한 모든 형제 엘리먼트들을 선택 예) \$('li#id1 ~ li')

제이쿼리 계층 선택자 실행하기

- [그림 9-11] hierarchy-selector.html의 실행 결과(예제9-5)

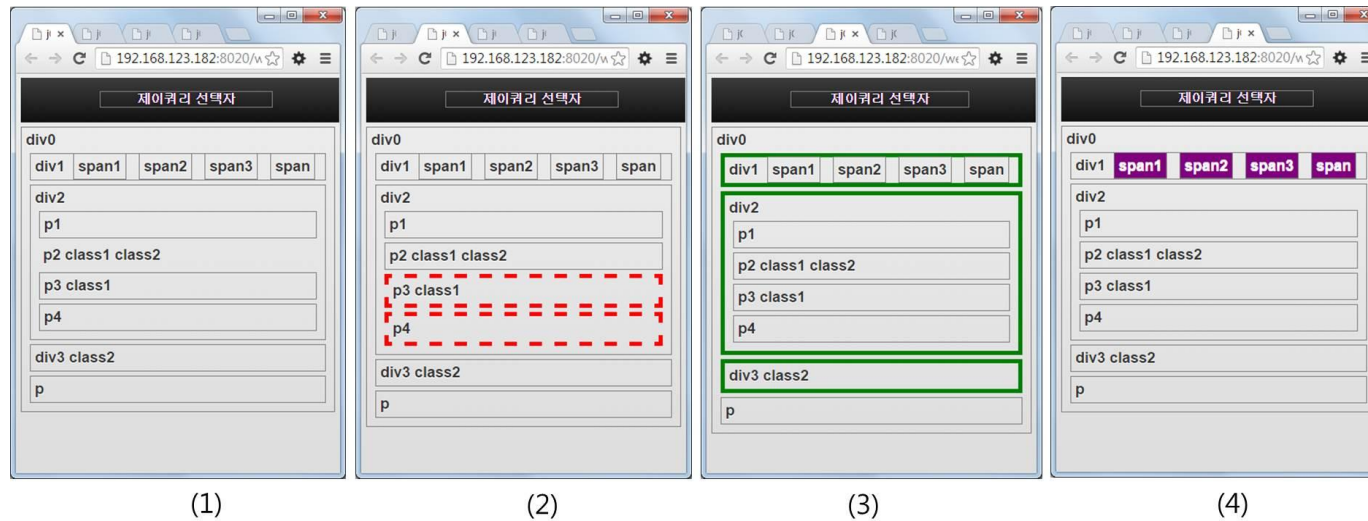
[예제9-5] 제이쿼리 계층 선택자 실행하기

/ch09/hierarchy-selector.html

```

$('#p1 + p').css('border-style', 'hidden');           // (1)
$('#p2 ~ p').css('border', 'dashed thick red');       // (2)
$('div > div').css('border', 'solid thick green');     // (3)
$('div span').css({'background-color': 'purple', 'color': 'white'}); // (4)

```



제이쿼리 선택자 유형[3]

● 속성 선택자

- 속성 관련 제이쿼리 선택자도 CSS3 선택자와 동일

형식	기능
선택자[속성명]	해당 속성을 포함하는 엘리먼트들을 선택 예) \$('div[class]', \$('img[alt][src]') (alt와 src 속성을 모두 갖는 img 태그)
선택자[속성명="조건값"]	해당 속성값이 조건값과 일치하는 엘리먼트들을 선택 예) \$('input[type="text"]')
선택자[속성명*="조건값"]	해당 속성값이 조건값을 포함하는 엘리먼트들을 선택 예) \$('input[type*="text"]') (text, textarea 모두 해당)
선택자[속성명^="조건값"]	해당 속성값이 조건값으로 시작하는 엘리먼트들을 선택 예) \$('a[href^="http://"]')
선택자[속성명\$="조건값"]	해당 속성값이 조건값으로 끝나는 엘리먼트들을 선택 예) \$('img[src\$="png"]')

제이쿼리 속성 선택자 실행하기

- [그림 9-12] attribute-selector.html의 실행 결과(예제9-6)

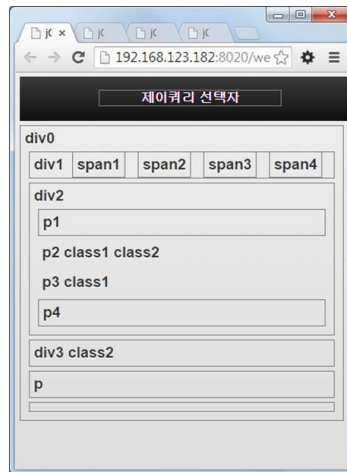
[예제9-6] 제이쿼리 속성 선택자 실행하기

/ch09/attribute-selector.html

```

$(p[class] ').css('border-style', 'hidden');           // (1)
$(*[id*= "2"]).css('border', 'dashed thick red');      // (2)
$(*[class^= "cla"]).css('border', 'solid thick green'); // (3)
$(p[id$= "2"]).css({'background-color': 'purple', 'color': 'white'}); // (4)

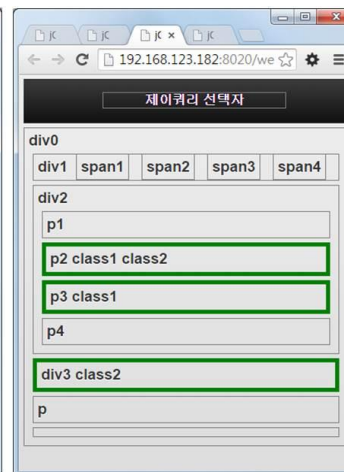
```



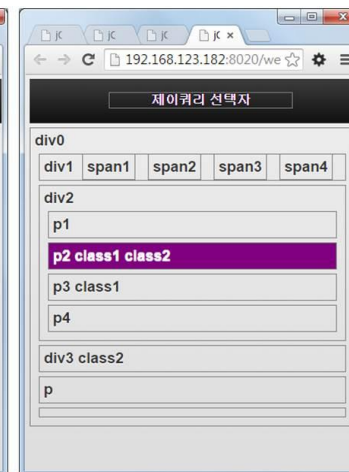
(1)



(2)



(3)



(4)

제이쿼리 선택자 유형[4]

● 위치필터 기본 선택자

- 필터(filter)는 선택자에 의해 선택된 엘리먼트들 중에서 필터 조건에 맞는 엘리먼트들로 선택 대상을 좀 더 제한(한번 더 거르는)하는 선택자 유형
- 선택자 뒤에 콜론(:)을 붙여 명세
- 반환된 엘리먼트들은 자바스크립트의 배열 첨자를 적용하므로 첨자가 '0'부터 부여됨
 - 예) :first는 :eq(0)과 :lt(1)과 그 의미가 모두 같음

형식	기능(첨자가 0부터 시작됨)
선택자:first	선택된 엘리먼트들 중에서 첫 번째 엘리먼트를 선택 예) \$('li:first')
선택자:last	선택된 엘리먼트들 중에서 마지막 엘리먼트를 선택 예) \$('li:last')
선택자:odd	선택된 엘리먼트들 중에서 홀수 첨자 엘리먼트들을 선택 예) \$('li:odd')
선택자:even	선택된 엘리먼트들 중에서 짝수 첨자 엘리먼트들을 선택 예) \$('li:even')
선택자:eq(n)	선택된 엘리먼트들 중에서 첨자 n인 엘리먼트를 선택 예) \$('li:eq(4)') (다섯 번째 li 태그)
선택자:lt(n)	선택된 엘리먼트들 중에서 첨자 n인 엘리먼트보다 앞 엘리먼트들을 선택 예) \$('li:lt(3)') (첫 번째, 두 번째, 세 번째 li 태그들)
선택자:gt(n)	선택된 엘리먼트들 중에서 첨자 n인 엘리먼트보다 뒤 엘리먼트들을 선택 예) \$('li:gt(3)') (다섯 번째 li 태그를 포함한 이후의 li 태그들)
선택자:not()	선택된 엘리먼트들 중에서 필터링 조건을 만족하지 않는 엘리먼트들을 선택 예) \$('li:not(even)') (홀수 번째 노드들을 선택)

제이쿼리 위치필터 선택자 실행하기

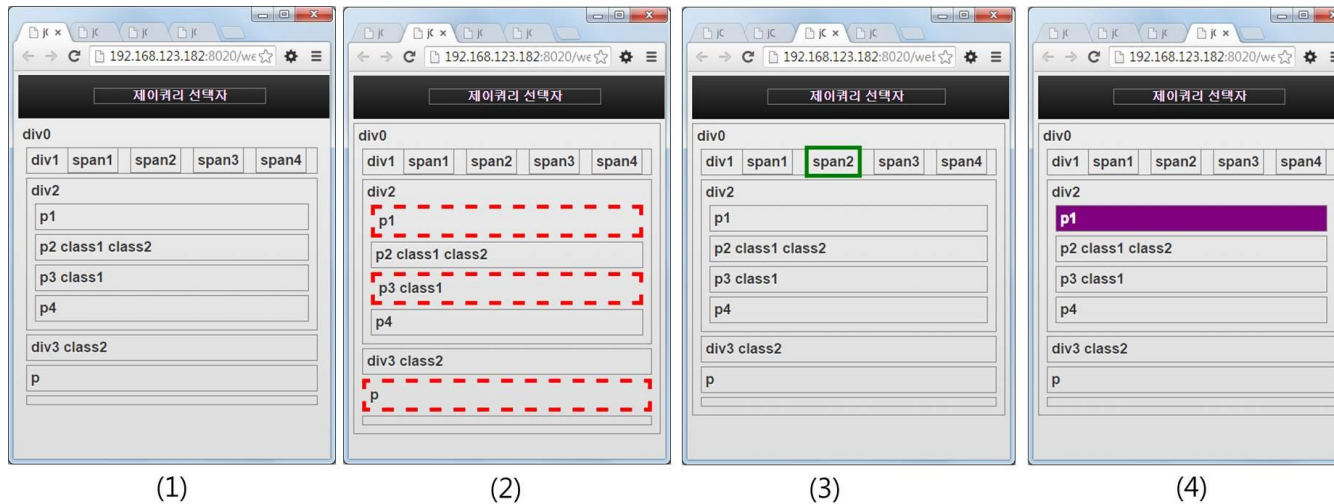
- [그림 9-13] position-filter-selector.html의 실행 결과(예제9-7)

[예제9-7] 제이쿼리 위치필터 선택자 실행하기 /ch09/position-filter-selector.html

```

$('div:first').css('border-style', 'hidden');           // (1)
$('p:even').css('border', 'dashed thick red');         // (2)
$('span:eq(1)').css('border', 'solid thick green');    // (3)
$('p:lt(1)').css({'background-color': 'purple', 'color': 'white'}); // (4)

```



제이쿼리 선택자 유형[5]

● 위치필터 계층 선택자

- 선택된 엘리먼트 노드들을 자식 노드로서의 위치 필터를 통해 한번 더 거르는 자식 유형 선택자

형식	기능(참자가 1부터 시작됨)
선택자:first-child	선택된 엘리먼트들 중에서 부모의 첫 번째 자식 엘리먼트들만을 선택 예) \$('li:first-child')
선택자:last-child	선택된 엘리먼트들 중에서 부모의 마지막 자식 엘리먼트들만을 선택 예) \$('li:last-child')
선택자:nth-child(odd)	선택된 엘리먼트들 중에서 부모의 홀수 번째 자식 엘리먼트들만을 선택 예) \$('li:nth-child(odd)')
선택자:nth-child(even)	선택된 엘리먼트들 중에서 부모의 짝수 번째 자식 엘리먼트들만을 선택 예) \$('li:nth-child(even)')
선택자:nth-child(n)	선택된 엘리먼트들 중에서 부모의 n번째 자식 엘리먼트만을 선택 예) \$('li:nth-child(3)')
선택자:nth-child(Xn+Y)	선택된 엘리먼트들 중에서 부모의 X배수 번째(Y번째부터 시작해서)에 있는 자식 엘리먼트들만을 선택 예) \$('li:nth-child(3n)'), \$('li:nth-child(3n+1)')
선택자:only-child	선택된 엘리먼트들 중에서 (형제 엘리먼트가 없는) 유일한 자식 엘리먼트들만을 선택 예) \$('li:only-child')

제이쿼리 계층필터 선택자 실행하기

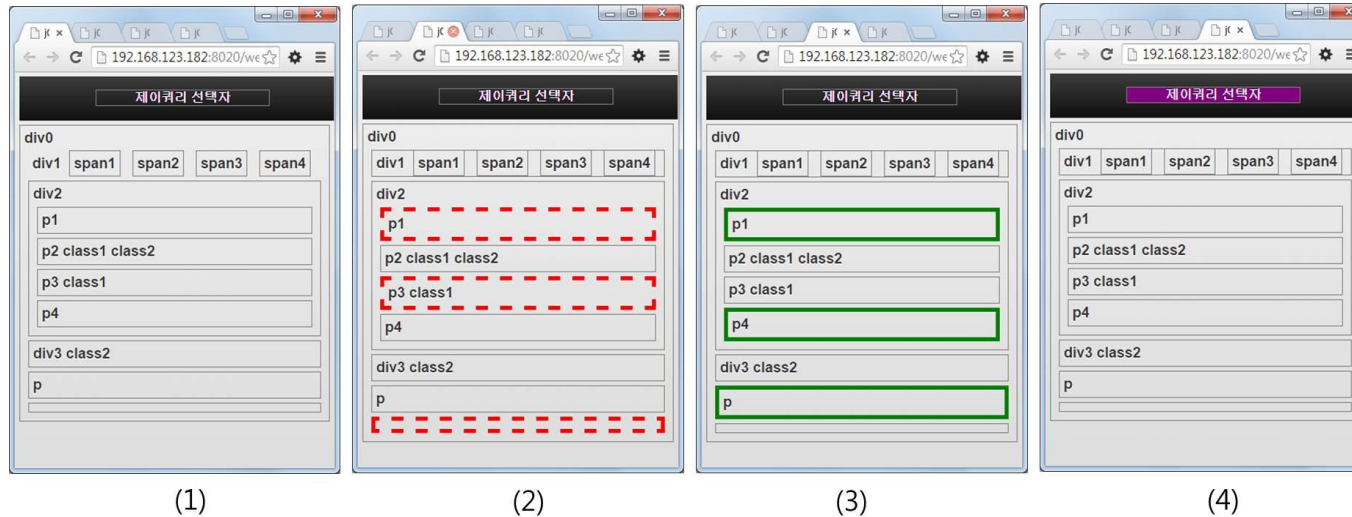
- [그림 9-14] hierarchy-filter-selector.html의 실행 결과(예제9-8)

[예제9-8] 제이쿼리 계층필터 선택자 실행하기

/ch09/hierarchy-filter-selector.html

```

$('div:first-child').css('border-style', 'hidden');           // (1)
$('p:nth-child(odd)').css('border', 'dashed thick red');      // (2)
$('p:nth-child(3n+1)').css('border', 'solid thick green');    // (3)
$('*:only-child').css({'background-color': 'purple', 'color': 'white'}); // (4)
  
```



제이쿼리 선택자 유형[6]

● 폼 필터 선택자

- 폼 관련 엘리먼트들 중 특정 입력 유형의 필터를 통해 한번 더 거르는 선택자
- type 속성값에 따라 특정 <input> 엘리먼트만을 선별하기 위해 사용

형식	기능
선택자:input	모든 폼 입력 양식의 엘리먼트를 선택 (<input>, <select>, <button>, <textarea> 태그들)
선택자:button	모든 버튼 유형의 엘리먼트를 선택 (<button> 태그와 type 속성값이 'submit', 'reset', 'button'인 <input> 태그들)
선택자:text	텍스트 엘리먼트만 선택 (type 속성값이 'text'인 <input> 태그)
선택자:checkbox	체크박스 선택 버튼 엘리먼트만 선택 (type 속성값이 'checkbox'인 <input> 태그)
선택자:radio	라디오 선택 버튼 엘리먼트만 선택 (type 속성값이 'radio'인 <input> 태그)
선택자:file	파일 엘리먼트를 선택 (type 속성값이 'file'인 <input> 태그)
선택자:image	폼 이미지 엘리먼트를 선택 (type 속성값이 'image'인 <input> 태그)
선택자:password	패스워드 엘리먼트를 선택 (type 속성값이 'password'인 <input> 태그)
선택자:submit	전송 버튼 엘리먼트를 선택 (type 속성값이 'submit'인 <input> 태그)
선택자:reset	초기화 버튼 엘리먼트를 선택 (type 속성값이 'reset'인 <input> 태그)

제이쿼리 선택자 유형[7]

● 상태 필터 선택자

- 현재 웹 브라우저 실행 화면에서의 특정 상황 조건을 만족하는지 여부에 따라 엘리먼트를 한번 더 거르는 필터 선택자
- 폼 엘리먼트의 현재 상태를 검사

형식	기능
선택자:selected	지정된 엘리먼트들 중에서 현재 선택된 엘리먼트만 선택
선택자:checked	지정된 엘리먼트들 중에서 현재 설정된 엘리먼트(체크박스,라디오버튼)만 선택
선택자:enabled	지정된 엘리먼트들 중에서 현재 활성화된 엘리먼트만 선택
선택자:disabled	지정된 엘리먼트들 중에서 현재 비활성화된 엘리먼트만 선택
선택자:visible	지정된 엘리먼트들 중에서 현재 보여지는 엘리먼트만 선택
선택자:hidden	지정된 엘리먼트들 중에서 현재 숨겨진 엘리먼트만 선택
선택자:focus	지정된 엘리먼트들 중에서 현재 포커싱된 엘리먼트만 선택
선택자:animated	지정된 엘리먼트들 중에서 현재 애니메이션이 동작 중인 엘리먼트만 선택

제이쿼리 선택자 유형(8)

● 내용 필터 선택자

- 엘리먼트의 내용 조건 필터를 통해 한번 더 거르는 선택자
- 특정 엘리먼트가 내용이 있는지 또 내용 중에 특정 문자열이 포함되는지 확인

형식	기능
선택자:has(자손엘리먼트)	선택된 엘리먼트들 중에서 하위에 특정 자손 엘리먼트를 하나이상 포함하고 있는 엘리먼트만 선택
선택자:contains(문자열)	선택된 엘리먼트들 중에서 내용으로 특정 문자열을 포함하고 있는 엘리먼트만 선택
선택자:empty	선택된 엘리먼트들 중에서 내용이 없는 엘리먼트만 선택
선택자:parent	선택된 엘리먼트들 중에서 내용(공백 포함)이 있는 엘리먼트(다른 엘리먼트의 부모에 해당하는 엘리먼트)만 선택

제이쿼리 내용필터 선택자 실행하기

- [그림 9-15] content-filter-selector.html의 실행 결과(예제9-9)

[예제9-9] 제이쿼리 내용필터 선택자 실행하기

/ch09/content-filter-selector.html

```

$('*:contains(2)').css('border-style', 'hidden');           // (1)
$('div:has(span)').css('border', 'dashed thick red');      // (2)
$('*:empty').css('border', 'solid thick green');           // (3)
$('#div0 *:parent').css({'background-color': 'purple', 'color': 'white'}); // (4)

```

