

The background features a series of flowing, wavy green lines that create a sense of movement and depth. The colors range from a vibrant lime green to a darker, more muted green, with some areas appearing as if they are part of a larger, curved surface. The overall effect is modern and organic.

# jQuery Core Concept

# jQuery 개요

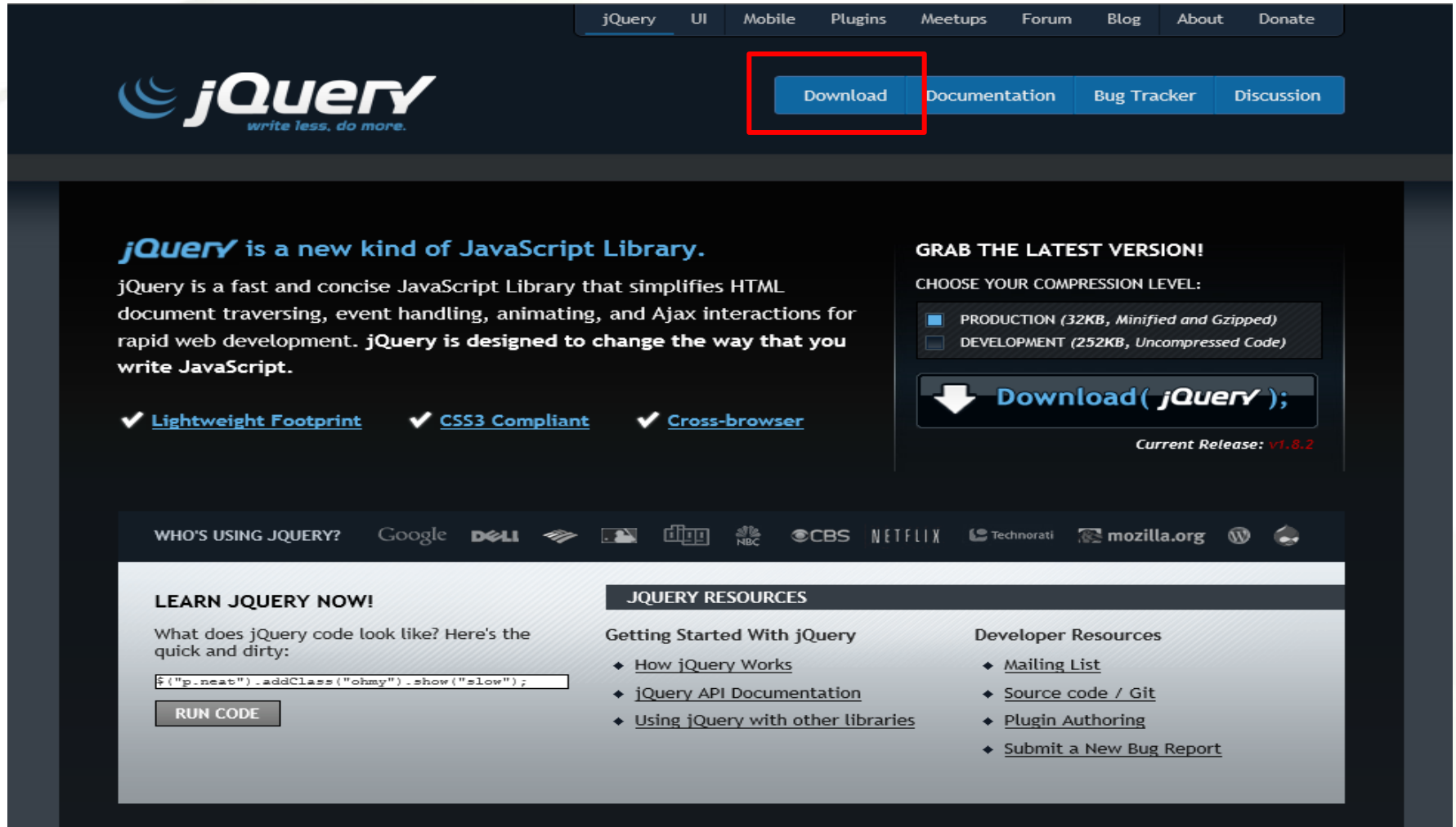
- 모든 브라우저에서 동작하는 클라이언트 자바스크립트 라이브러리
- 2006년 1월, 존 레식 John Resig 이 BarCamp NYC에서 발표
- 무료로 사용 가능한 오픈소스 라이브러리
- jQuery의 제작 목표
  - DOM과 관련된 처리 쉽게 구현
  - 일관된 이벤트 연결 쉽게 구현
  - 시각적 효과 쉽게 구현
  - Ajax 애플리케이션 쉽게 개발

# jQuery 장점

- 브라우저 특성을 고려하지 않고 코드 개발 가능
- 높은 생산성
- 플래시와 실버라이트로 구현한 웹 사이트와 비슷한 수준의 시각적 효과 구현

# jQuery 사용

- <http://www.jquery.com> 접속



The screenshot shows the jQuery website homepage. At the top, there is a navigation bar with links: jQuery, UI, Mobile, Plugins, Meetups, Forum, Blog, About, and Donate. Below this, the jQuery logo is displayed with the tagline "write less, do more.". To the right of the logo, there is a red box highlighting the "Download" button, which is part of a row of buttons including "Documentation", "Bug Tracker", and "Discussion".

The main content area features the text: "jQuery is a new kind of JavaScript Library. jQuery is a fast and concise JavaScript Library that simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development. jQuery is designed to change the way that you write JavaScript."

Below this text, there are three checkmarks with labels: "✓ Lightweight Footprint", "✓ CSS3 Compliant", and "✓ Cross-browser".

On the right side, there is a section titled "GRAB THE LATEST VERSION!" with the subtext "CHOOSE YOUR COMPRESSION LEVEL:". There are two radio buttons: "PRODUCTION (32KB, Minified and Gzipped)" (which is selected) and "DEVELOPMENT (252KB, Uncompressed Code)". Below this, there is a large "Download (jQuery);" button with a download icon. Underneath the button, it says "Current Release: v1.8.2".

At the bottom, there is a section titled "WHO'S USING JQUERY?" with logos for Google, DELL, IBM, NBC, CBS, NETFLIX, Technorati, and mozilla.org. Below this, there are two columns of links. The left column is titled "LEARN JQUERY NOW!" and contains the text "What does jQuery code look like? Here's the quick and dirty:" followed by a code snippet: `$("#p.neat").addClass("ohmy").show("slow");` and a "RUN CODE" button. The right column is titled "JQUERY RESOURCES" and contains two sub-sections: "Getting Started With jQuery" with links to "How jQuery Works", "jQuery API Documentation", and "Using jQuery with other libraries"; and "Developer Resources" with links to "Mailing List", "Source code / Git", "Plugin Authoring", and "Submit a New Bug Report".

# jQuery 사용

- 다운로드 경로 ➔ <http://jquery.com/download/>
- CDN Link를 이용해서 사용하거나
  - [http://code.jquery.com/jquery-\[version\].min.js](http://code.jquery.com/jquery-[version].min.js)
  - [http://ajax.googleapis.com/ajax/libs/jquery/\[version\]/jquery.min.js](http://ajax.googleapis.com/ajax/libs/jquery/[version]/jquery.min.js)
  - [http://ajax.aspnetcdn.com/ajax/jQuery/jquery-\[version\].min.js](http://ajax.aspnetcdn.com/ajax/jQuery/jquery-[version].min.js)
  - [http://cdnjs.cloudflare.com/ajax/libs/jquery/\[version\]/jquery.min.js](http://cdnjs.cloudflare.com/ajax/libs/jquery/[version]/jquery.min.js)
- 파일 다운로드 및 사이트에 포함해서 배포
  - 1.x
    - Download the compressed, production jQuery 1.11.0
    - Download the uncompressed, development jQuery 1.11.0
    - Download the map file for jQuery 1.11.0
    - jQuery 1.11.0 release notes
  - 2.x
    - Download the compressed, production jQuery 2.1.0
    - Download the uncompressed, development jQuery 2.1.0
    - Download the map file for jQuery 2.1.0
    - jQuery 2.1.0 release notes

# CDN 배포

- CDN은 Content Delivery Network 의 약자
- 사용자에게 간편하게 콘텐츠 제공하는 방식 의미
  - 구글, 마이크로소프트, jQuery측에서 사용자가 jQuery를 사용하기 편하게 콘텐츠 제공
  - HTML 페이지 구성

```
<!DOCTYPE html>
<html>
<head>
  <script src="http://code.jquery.com/jquery-1.7.js"></script>
  <script>

  </script>
</head>
<body>

</body>
</html>
```

# Download 및 파일 배포

- jQuery 파일명
  - ○○.js 파일
    - Uncompressed 버전
  - ○○.min.js 파일
    - ○○.min.js 파일은 Minified 버전 (용량이 다섯 배 이상 차이)
    - Minified 버전은 파일의 용량을 최소화하려고 압축한 파일
- 오프라인에서 jQuery 사용
  - 반드시 다운받아 사용

# jQuery 구문 요소

- 확장 요소 집합 객체 생성

- CSS의 선택자 문법을 이용해서 문서 내의 요소를 선택
- 선택된 요소를 감싸서 추가 기능을 포함하는 새로운 객체 제공

- 커맨드

- 확장 요소 집합 객체의 멤버로 구성 되어 특별한 기능을 제공하는 jQuery 메서드
- 주로 요소 집합을 제어하는 기능 제공

- 함수

- jQuery에 포함된 전역 함수
- 프로그램에 필요한 일반적인 기능을 제공



# jQuery & \$

- jQuery 키워드는 jQuery 라이브러리의 최상위 객체 및 이름 영역 역할
- jQuery의 모든 구문은 jQuery로 시작
- 구문을 단순화 하기 위해 jQuery 대신 \$ 사용 (\$는 jQuery의 약식 명칭)
- 다른 라이브러리와 함께 사용할 때 \$이름 충돌 문제 관리 필요

# jQuery(document).ready( )

## ■ \$(document).ready()

- 문서가 준비가 완료되면 매개 변수로 전달된 함수를 실행하라는 의미
- jQuery 이벤트 메서드 중 하나
- 아래 위 두 메서드는 비슷한 역할

코드 13-1 document 객체의 ready 이벤트 연결

```
<script>
    $(document).ready(function () {

    });
</script>
```

코드 13-2 window 객체의 load 이벤트 연결

```
<script>
    window.onload = function () {

    };
</script>
```

# 초기화

- \$(document).ready()
  - 3개의 경고창 띄우는 예제

코드 13-3 여러 개의 이벤트 연결

```
<script>
  $(document).ready(function () {
    alert('First READY');
  });

  $(document).ready(function () {
    alert('Second READY');
  });

  $(document).ready(function () {
    alert('Third READY');
  });
</script>
```

- 간단하게 사용되는 형태도 존재

코드 13-4 간단한 형식의 \$(document).ready()

```
<script>
  $(function () {

  });
</script>
```

# jQuery 기본 선택자

선택터	설명
*	모든 엘리먼트와 일치
E	태그명이 E인 모든 엘리먼트와 일치
E F	E의 자손이면서 엘리먼트의 이름이 F인 모든 엘리먼트
E>F	E의 바로 아래 F 엘리먼트
E+F	E의 형제 엘리먼트로 바로 다음 형제 F 엘리먼트
E~F	E의 형제 엘리먼트로 다음에 나오는 모든 F 엘리먼트
E:has(F)	엘리먼트 F의 자손을 하나 이상 가지고 있는 E 모든 엘리먼트
E.C	클래스명 C를 가지고 있는 모든 E 엘리먼트.
*.C	클래스명 C를 가지고 있는 모든 엘리먼트
E#I	아이디가 I 인 모든 E 엘리먼트.
*#I	아이디가 I인 모든 엘리먼트

# jQuery 기본 선택자

## ■ 전체 선택자

- CSS의 가장 기본적인 선택자는 전체 선택자 Wildcard Selector
- HTML 페이지에 있는 모든 문서 객체를 선택하는 선택자 \*
- 모든 문서 객체의 color 스타일 속성에 red 입력

코드 13-5 전체 선택자

```
<!DOCTYPE html>
<html>
<head>
  <script src="http://code.jquery.com/jquery-1.7.js"></script>
  <script>
    $(document).ready(function () {
      $('*').css('color', 'Red');
    });
  </script>
</head>
<body>
  <h1>Lorem ipsum</h1>
</body>
</html>
```

그림 13-6 전체 선택자

```
▼<html style="color: red; ">
  ▼<head style="color: red; ">
    <script src="http://code.jquery.com/jquery-1.6.1.js" style="color: red; "></script>
    ▶<script style="color: red; "></script>
  </head>
  ▼<body style="color: red; ">
    <h1 style="color: red; ">Lorem ipsum</h1>
  </body>
</html>
```

# jQuery 기본 선택자

## ■ 태그 선택자

- 태그 선택자는 특정한 태그만 선택하는 선택자
- 태그의 이름 그냥 사용

코드 13-6 body 태그 구성

```
<body>
  <h1>Lorem ipsum</h1>
  <p>Lorem ipsum dolor sit amet.</p>
  <h1>Lorem ipsum</h1>
  <p>consectetur adipiscing elit.</p>
</body>
```

코드 13-7 태그 선택자

```
<script>
  $(document).ready(function () {
    $('h1').css('color', 'Orange');
  });
</script>
```

그림 13-7 태그 선택자

```
▼<body>
  <h1 style="color: orange; ">Lorem ipsum</h1>
  <p>Lorem ipsum dolor sit amet.</p>
  <h1 style="color: orange; ">Lorem ipsum</h1>
  <p>consectetur adipiscing elit.</p>
</body>
```

# jQuery 기본 선택자

- 태그 선택자의 활용
  - 하나 이상의 태그 선택자를 동시에 사용하고 싶을 때
    - 콤마로 선택자 구분

코드 13-8 여러 개의 태그 선택

```
<script>
  $(document).ready(function () {
    $('h1, p').css('color', 'Orange');
  });
</script>
```

# jQuery 기본 선택자

- 아이디 선택자
  - 특정한 id 속성을 가지고 있는 문서 객체 선택하는 선택자
  - Ex) 예제 두 번째에 위치한 h1 태그가 id 속성으로 target 가짐

코드 13-9 body 태그 구성

```
<body>
  <h1>Header-0</h1>
  <h1 id="target">Header-1</h1>
  <h1>Header-2</h1>
</body>
```

코드 13-10 아이디 선택자

```
<script>
  $(document).ready(function () {
    $('#target').css('color', 'orange');
  });
</script>
```



# jQuery 기본 선택자

## ■ 아이디 선택자의 활용

- id 속성은 HTML 페이지 내에서 단 하나의 태그에만 적용
- 태그 선택자와 아이디 선택자 함께 사용하지 않아도 됨

코드 13-11 특정 id 속성을 가지는 태그 선택

```
<script>
  $(document).ready(function () {
    $('h1#target').css('color', 'Orange');
  });
</script>
```

# jQuery 기본 선택자

- 클래스 선택자
  - 특정한 class 속성 가진 문서 객체를 선택하는 선택자

코드 13-12 body 태그 구성

```
<body>
  <h1 class="item">Header-0</h1>
  <h1 class="item select">Header-1</h1>
  <h1 class="item">Header-2</h1>
</body>
```

코드 13-13 클래스 선택자와 특정 class 속성을 가지는 태그 선택

```
<script>
  $(document).ready(function () {
    $('.item').css('color', 'Orange');
    $('h1.item').css('background', 'Red');
  });
</script>
```

# jQuery 기본 선택자

- 두 클래스 속성을 모두 갖는 문서 객체를 선택하고 싶을 때
  - 두 클래스 선택자 붙여서 사용

코드 13-14 두 가지 클래스를 모두 가지는 태그 선택

```
<script>
    $(document).ready(function () {
        $('.item.select').css('color', 'Orange');
    });
</script>
```

그림 13-8 두 가지 클래스 선택

```
▼ <body>
  <h1 class="item">Header-0</h1>
  <h1 class="item select" style="color: orange;">Header-1</h1>
  <h1 class="item">Header-2</h1>
</body>
```

# jQuery 자식 선택자와 후손 선택자

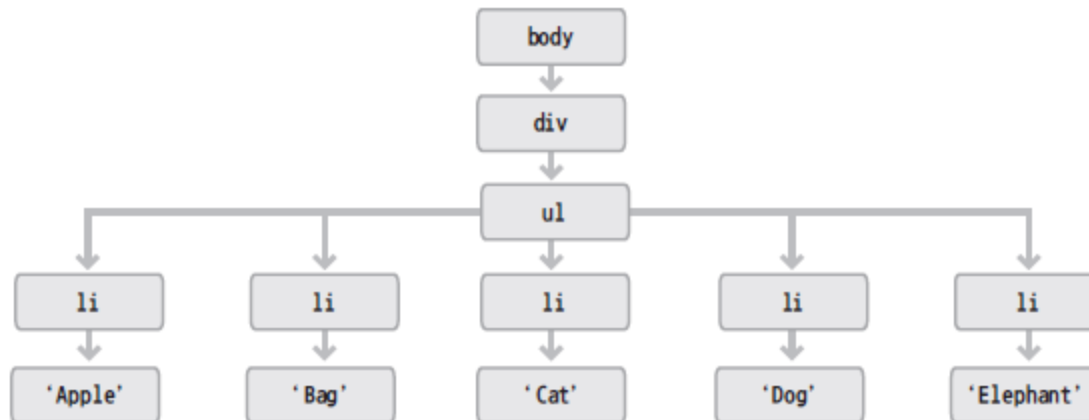
## ■ jQuery 자식 선택자와 후손 선택자

- 기본 선택자의 앞에 붙여 사용하며 기본 선택자의 범위 제한

코드 13-15 body 태그 구성

```
<body>
  <div>
    <ul>
      <li>Apple</li>
      <li>Bag</li>
      <li>Cat</li>
      <li>Dog</li>
      <li>Elephant</li>
    </ul>
  </div>
</body>
```

그림 13-9 자식과 후손



# jQuery 자식 선택자와 후손 선택자

## ■ 자식 선택자

- 자식을 선택하는 선택자
- ‘부모 > 자식’ 의 형태로 사용

코드 13-16 자식 선택자

```
<script>
    $(document).ready(function () {
        $('body > *').css('color', 'red');
    });
</script>
```

그림 13-10 자식 선택자

```
▼<html>
  ▶<head>...</head>
  ▼<body>
    ▼<div style="color: red; ">
      ▼<ul>
        <li>Apple</li>
        <li>Bag</li>
        <li>Cat</li>
        <li>Dog</li>
        <li>Elephant</li>
      </ul>
    </div>
  </body>
</html>
```

# jQuery 자식 선택자와 후손 선택자

## ■ 후손 선택자

- 후손을 선택하는 선택자
- ‘요소 A 요소 B’ 의 형태로 사
  - 요소 A의 후손으로 범위 한정
- 예제 코드 13-17]
  - body 태그의 모든 후손 선택

코드 13-17 후손 선택자

```
<script>
  $(document).ready(function () {
    $('body *').css('color', 'red');
  });
</script>
```

그림 13-11 후손 선택자

```
▼<html>
  ▶<head>...</head>
  ▼<body>
    ▼<div style="color: red; ">
      ▼<ul style="color: red; ">
        <li style="color: red; ">Apple</li>
        <li style="color: red; ">Bag</li>
        <li style="color: red; ">Cat</li>
        <li style="color: red; ">Dog</li>
        <li style="color: red; ">Elephant</li>
      </ul>
    </div>
  </body>
</html>
```

# jQuery 속성 선택자

- 기본 선택자 뒤에 붙여 사용
- 종류

표 13-1 jQuery 속성 선택자

선택자 형태	설명
요소[속성]	특정 속성을 가지고 있는 문서 객체를 선택합니다.
요소[속성=값]	속성 안의 값이 특정 값과 같은 문서 객체를 선택합니다.
요소[속성~=값]	속성 안의 값이 특정 값을 단어로써 포함하는 문서 객체를 선택합니다.
요소[속성^=값]	속성 안의 값이 특정 값으로 시작하는 문서 객체를 선택합니다.
요소[속성\$=값]	속성 안의 값이 특정 값으로 끝나는 문서 객체를 선택합니다.
요소[속성*=값]	속성 안의 값이 특정 값을 포함하는 문서 객체를 선택합니다.

# jQuery 속성 선택자

## ■ 속성 선택자 사용 예제 코드

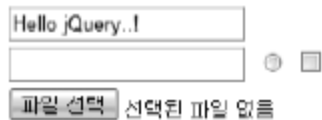
코드 13-18 body 태그 구성

```
<body>
  <input type="text" />
  <input type="password" />
  <input type="radio" />
  <input type="checkbox" />
  <input type="file" />
</body>
```

코드 13-19 속성 선택자

```
<script>
$(document).ready(function () {
  $('input[type=text]').val('Hello jQuery..!');
});
</script>
```

그림 13-12 속성 선택자



Hello jQuery..!

☐ ☐

파일 선택 선택된 파일 없음



# jQuery 입력 양식 필터 선택자

- input 태그의 type 속성 따라 문서 객체 선택 가능
- 종류

표 13-2 jQuery의 입력 양식 필터 선택자(1)

선택자 형태	설명
요소:button	input 태그 중 type 속성이 button인 문서 객체와 button 태그를 선택합니다.
요소:checkbox	input 태그 중 type 속성이 check인 문서 객체를 선택합니다.
요소:file	input 태그 중 type 속성이 file인 문서 객체를 선택합니다.
요소:image	input 태그 중 type 속성이 image인 문서 객체를 선택합니다.
요소:password	input 태그 중 type 속성이 password인 문서 객체를 선택합니다.
요소:radio	input 태그 중 type 속성이 radio인 문서 객체를 선택합니다.
요소:reset	input 태그 중 type 속성이 reset인 문서 객체를 선택합니다.
요소:submit	input 태그 중 type 속성이 submit인 문서 객체를 선택합니다.
요소:text	input 태그 중 type 속성이 text인 문서 객체를 선택합니다.

표 13-3 jQuery의 입력 양식 필터 선택자 (2)

선택자 형태	설명
요소:checked	체크된 입력 양식을 선택합니다.
요소:disabled	비활성화된 입력 양식을 선택합니다.
요소:enabled	활성화된 입력 양식을 선택합니다.
요소:focus	초점이 맞춰져 있는 입력 양식을 선택합니다.
요소:input	모든 입력 양식을 선택합니다(input, textarea, select, button 태그).
요소:selected	option 객체 중 선택된 태그를 선택합니다.

# jQuery 입력 양식 필터 선택자

## ■ 속성 선택자 사용 예제

- setTimeout() 함수 사용해 웹 페이지 실행 → 5초 정도 후 선택자 사용
- body 태그 구성

코드 13-20 body 태그 구성

```
<body>
  <select>
    <option>Apple</option>
    <option>Bag</option>
    <option>Cat</option>
    <option>Dog</option>
    <option>Elephant</option>
  </select>
</body>
```

코드 13-21 입력 양식 필터 선택자

```
<script>
  $(document).ready(function () {
    // 5초 후에 코드를 실행합니다.
    setTimeout(function () {
      // 변수를 선언합니다.
      var value = $('select > option:selected').val();

      // 출력합니다.
      alert(value);
    }, 5000);
  });
</script>
```

그림 13-13 필터 선택자를 사용한 선택한 문서 객체 선택



# jQuery 필터 선택자

## ■ 위치 지정 필터 선택자

표 13-4 jQuery 기본 필터 선택자

선택자 형태	설명
요소:odd	홀수 번째에 위치한 문서 객체를 선택합니다.
요소:even	짝수 번째에 위치한 문서 객체를 선택합니다.
요소:first	첫 번째 위치한 문서 객체를 선택합니다.
요소:last	마지막에 위치한 문서 객체를 선택합니다.

## ■ 함수 형태 필터 선택자

표 13-5 jQuery 함수 필터 선택자

선택자 형태	설명
요소:contains(문자열)	특정 문자열을 포함하는 문서 객체를 선택합니다.
요소:eq(n)	n번째에 위치하는 문서 객체를 선택합니다.
요소:gt(n)	n번째 초과에 위치하는 문서 객체를 선택합니다.
요소:has(h1)	h1 태그를 가지고 있는 문서 객체를 선택합니다.
요소:lt(n)	n번째 미만에 위치하는 문서 객체를 선택합니다.
요소:not(선택자)	선택자와 일치하지 않는 문서 객체를 선택합니다.
요소:nth-child(3n+1)	3n+1번째에 위치하는 문서 객체를 선택합니다(1, 4, 7, .....).

# jQuery 필터 선택자

- body 태그 구성
  - 줄무늬 모양의 표 만들기

코드 13-22 body 태그 구성

```
<body>
  <table>
    <tr><th>이름</th><th>혈액형</th><th>지역</th></tr>
    <tr><td>강민수</td><td>AB형</td><td>서울특별시 송파구</td></tr>
    <tr><td>구지연</td><td>B형</td><td>미국 캘리포니아</td></tr>
    <tr><td>김미화</td><td>AB형</td><td>미국 메사추세츠</td></tr>
    <tr><td>김선화</td><td>O형</td><td>서울 강서구</td></tr>
    <tr><td>남기주</td><td>A형</td><td>서울 노랑진구</td></tr>
    <tr><td>윤하린</td><td>B형</td><td>서울 용산구</td></tr>
  </table>
</body>
```

코드 13-23 필터 선택자

```
<script>
  $(document).ready(function () {
    $('tr:odd').css('background', '#F9F9F9');
    $('tr:even').css('background', '#9F9F9F');

    $('tr:first').css('background', '#000000').css('color', '#FFFFFF');
  });
</script>
```

그림 13-14 필터 선택자를 사용한 테이블 줄무늬 부여 (1)

이름	혈액형	지역
강민수	AB형	서울특별시 송파구
구지연	B형	미국 캘리포니아
김미화	AB형	미국 메사추세츠
김선화	O형	서울 강서구
남기주	A형	서울 노랑진구
윤하린	B형	서울 용산구

# jQuery 필터 선택자

## ■ 함수 형태의 필터 선택자

### – :nth-child() 필터 선택자

- 함수의 괄호 안에 식별자 n 사용해 문서 객체의 순서 나타냄
- 예제 코드 13-24의 첫 번째 nth-child() 필터 선택자
  - 3n+1번째에 위치하는 선택자 선택
    - » 1, 4, 7번째에 위치하는 문서 객체 선택

코드 13-24 함수 필터 선택자

```
<script>
$(document).ready(function () {
    $('tr:eq(0)').css('background', '#000000').css('color', 'White');

    $('td:nth-child(3n+1)').css('background', '#565656');
    $('td:nth-child(3n+2)').css('background', '#A9A9A9');
    $('td:nth-child(3n)').css('background', '#F9F9F9');
});
</script>
```

그림 13-15 필터 선택자를 사용한 테이블 줄무늬 부여 (2)

이름	혈액형	지역
강민수	AB형	서울특별시 송파구
구지연	B형	미국 캘리포니아
김미화	AB형	미국 메사추세츠
김선화	O형	서울 강서구
남기주	A형	서울 노랑진구
윤하림	B형	서울 용산구

# jQuery를 사용한 객체 확장

- \$.extend() 메서드
  - 객체 생성 후 속성 추가 – 데이터 수가 적을 때는 편리

코드 13-35 객체 생성

```
<script>
    $(document).ready(function () {
        var object = {};
    });
</script>
```

코드 13-36 객체에 속성 추가

```
<script>
    $(document).ready(function () {
        var object = {};
        object.name = 'RintIanTta';
        object.gender = 'Male';
        object.part = 'Second Guitar';
    });
</script>
```

# jQuery를 사용한 객체 확장

- \$.extend() 메서드
  - 많은 수의 속성 추가할 때 생기는 문제를 해결하는 메서드
  - 사용법
    - \$.extend(object, addObject, addObject, ...)

코드 13-37 \$.extend() 메서드

```
<script>
$(document).ready(function () {
    // 변수를 선언합니다.
    var object = { name: 'RintlanTta' };

    // $.extend() 메서드를 사용합니다.
    $.extend(object, {
        gender: 'Male',
        part: 'Second Guitar'
    });

    // 출력합니다.
    var output = '';
    $.each(object, function (key, item) {
        output += key + ': ' + item + '\n';
    });
    alert(output);
});
</script>
```

그림 13-18 객체의 결합



# jQuery 프레임워크 충돌 방지

- \$.noConflict()
  - 충돌을 방지할 때 사용하는 메서드
  - \$.noConflict() 메서드 사용
    - 더 이상 jQuery의 식별자 \$를 사용할 수 없음

코드 13-38 \$.noConflict() 메서드

```
<script>
    $.noConflict();
    jQuery(document).ready(function () {

    });
</script>
```

코드 13-39 \$.noConflict() 메서드 활용

```
<script>
    // 플러그인 간의 충돌을 제거합니다.
    $.noConflict();
    var J = jQuery;

    // jQuery를 사용합니다.
    J(document).ready(function () {
        J('h1').removeClass('hight_light');
    });
</script>
```



# 확장 집합 제어

# 기본 필터링 메서드

## ■ 기본 필터링 메서드

- jQuery의 선택자를 사용하면 원하는 문서 객체 대부분 선택 가능
- 기본 지원하지 않는 필터를 사용해 문서 객체를 선택해야 한다면?
  - 표 14-1의 메서드 사용
  - 이 절에서는 filter() 메서드 사용법 연습

표 14-1 jQuery 메서드

메서드 이름	설명
filter()	문서 객체를 필터링합니다.

filter() 메서드는 다음과 같은 두 가지 형태로 사용합니다.

```
1 $(selector).filter(selector);  
2 $(selector).filter(function () { });
```

# 기본 필터링 메서드

## ▪ filter() 메서드 사용법 예제

- h3 태그에서 홀수 번째에 위치하는 문서 객체 선택
  - 코드 14-2는 홀수 번째에 위치하는 h3 태그의 스타일 속성 변경 (선택자)

코드 14-1 body 태그 구성

```
<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>
</body>
```

코드 14-2 필터 선택자

```
<script>
  $(document).ready(function () {
    $('h3:even').css({
      backgroundColor: 'Black',
      color: 'White'
    });
  });
</script>
```

# 기본 필터링 메서드

- filter() 메서드 사용법 예제
  - 1번 형태의 filter() 메서드 사용
  - 예제 코드 14-3]
    - filter() 메서드의 매개 변수에 선택자 입력
    - ‘:even’ 을 입력할 수도 있음

코드 14-3 filter() 메서드

```
<script>
    $(document).ready(function () {
        $('h3').filter(':even').css({
            backgroundColor: 'Black',
            color: 'White'
        });
    });
</script>
```

# 기본 필터링 메서드

- filter() 메서드 사용법 예제
  - 2번 형태의 filter() 메서드를 사용
    - filter() 메서드의 매개 변수에 함수를 넣음
    - 입력하는 함수는 매개 변수로 index를 가짐
    - 함수에서 리턴하는 값에 따라 문서 객체 선택

코드 14-4 filter() 메서드의 매개 변수로 함수를 넣을 경우

```
<script>
    $(document).ready(function () {
        $('h3').filter(function (index) {
            return index % 3 == 0;
        }).css({
            backgroundColor: 'Black',
            color: 'White'
        });
    });
</script>
```

그림 14-1 filter() 메서드의 사용

**Header-0**

**Header-1**

**Header-2**

**Header-3**

**Header-4**

**Header-5**

# 기본 필터링 메서드

- 체이닝을 사용해 한 줄로 서로 다른 문서 객체에 스타일 적용 가능

코드 14-5 HTML 페이지 구성

```
<!DOCTYPE html>
<html>
<head>
  <script src="http://code.jquery.com/jquery-1.7.js">/script>
  <script>
    $(document).ready(function () {
      $('h1').css('background', 'Orange').filter(':even').css('color', 'Red');
    });
  </script>
</head>
<body>
  <h1>Header-0</h1>
  <h1>Header-1</h1>
  <h1>Header-2</h1>
</body>
</html>
```

## 이전 결과 집합 반환

- 체이닝할 때 추가한 filter() 메서드 제거하려면 표 14-2의 메서드 사용

코드 14-6 연습

```
<script>
$(document).ready(function () {
    $('h1').css('background', 'Orange');
    $('h1:even').css('color', 'White');
    $('h1:odd').css('color', 'Red');
});
</script>
```

표 14-2 jQuery 메서드

메서드 이름	설명
end()	문서 객체 선택을 한 단계 뒤로 돌립니다.

end() 메서드를 사용하면 코드 14-6을 코드 14-7의 형태로 사용할 수 있습니다.

코드 14-7 end() 메서드를 사용한 체이닝

```
$('h1').css('background', 'Orange').filter(':even').css('color',
'White').end().filter(':odd').css('color', 'Red');
```

# 특정 위치의 문서 객체 선택

## ■ 필터 선택자 이용

- 특정 위치에 존재하는 문서 객체 선택
- 필터 선택자는 자주 사용하므로 메서드로 제공

표 14-3 jQuery 메서드

메서드 이름	설명
eq()	특정 위치에 존재하는 문서 객체를 선택합니다.
first()	첫 번째에 위치하는 문서 객체를 선택합니다.
last()	마지막에 위치하는 문서 객체를 선택합니다.

코드 14-8 body 태그 구성

```
<body>
  <div>
    <h1>Header-0</h1>
    <h1>Header-1</h1>
    <h1>Header-2</h1>
  </div>
</body>
```



# 특정 위치의 문서 객체 선택

## ▪ eq() 메서드

- 매개 변수에 숫자 입력
  - 양수를 입력할 수도 있지만, 음수를 입력할 수도 있음
  - 음수를 입력하면 뒤쪽 기준으로 선택
  - 맨 뒤의 문서 객체가 -1번째

코드 14-9 eq() 메서드

```
<script>
    $(document).ready(function () {
        $('h1').eq(0).css('background', 'Orange');
        $('h1').eq(-1).css('background', 'Red');
    });
</script>
```

그림 14-2 eq() 메서드의 사용

**Header-0**

**Header-1**

**Header-2**

# 문서 객체 추가 선택

## ■ add() 메서드

- jQuery는 문서 객체의 체이닝을 더 유연하게 도와줌
- 표 14-4의 메서드 제공

표 14-4 jQuery 메서드

메서드 이름	설명
add()	문서 객체를 추가로 선택합니다.

코드 14-10처럼 body 태그를 구성해주세요.

코드 14-10 body 태그 구성

```
<body>
  <h1>Header-0</h1>
  <h2>Header-1</h2>
  <h1>Header-2</h1>
  <h2>Header-3</h2>
  <h1>Header-4</h1>
</body>
```

# 문서 객체 추가 선택

## ■ 예제 코드 14-11]

- add() 메서드 활용 예제
- h1 태그의 background 스타일 속성에 Gray 입력
- h2 태그를 추가로 선택, float 스타일 속성에 left 입력

코드 14-11 add() 메서드

```
<script>
    $(document).ready(function () {
        $('h1').css('background', 'Gray').add('h2').css('float','left');
    });
</script>
```

그림 14-3 add() 메서드의 사용

**Header-0**Header-1**Header-2**Header-3Header-4

# 문서 객체의 특징 판별

- 문서 객체가 특징을 가지고 있는지 판단
  - 표 14-5의 메서드 사용
    - is() 메서드는 매개 변수로 선택자 입력
    - 선택한 객체가 선택자와 일치하는지 판별해 불리언 리턴

표 14-5 jQuery 메서드

메서드 이름	설명
is()	문서 객체의 특징을 판별합니다.

코드 14-12 body 태그 구성

```
<body>
  <h1 class="select">Header-0</h1>
  <h1>Header-1</h1>
  <h1 class="select">Header-2</h1>
</body>
```

# 문서 객체의 특징 판별

## ■ 예제 코드 14-13

- h1 태그 중 select 클래스 속성이 있는 문서 객체에 스타일 적용

코드 14-13 is() 메서드

```
<script>
    $(document).ready(function () {
        $('h1').each(function () {
            if ($(this).is('.select')) {
                $(this).css('background', 'orange');
            }
        });
    });
</script>
```

그림 14-4 is() 메서드의 사용

**Header-0**

**Header-1**

**Header-2**

# 특정 태그 선택

- 문서 객체에서 특정 태그를 선택하는 방법
  - XML 문서에서 데이터를 추출시 많이 사용하므로 예제 이해하도록
  - 특정 태그를 선택할 때는 표 14-6의 메서드 사용

표 14-6 jQuery 메서드

메서드 이름	설명
find()	특정 태그를 선택합니다.

# 문서 객체의 클래스 속성 추가

- 문서 객체에 클래스 속성을 추가할 때 사용하는 메서드
  - addClass() 메서드 - 매개 변수에 추가하려는 클래스 속성 이름 입력

표 15-1 jQuery의 메서드

메서드 이름	설명
addClass()	문서 객체의 클래스 속성을 추가합니다.

코드 15-1 body 태그 구성

```
<body>
  <h1>Header-0</h1>
  <h1>Header-1</h1>
  <h1>Header-2</h1>
</body>
```

코드 15-2 addClass() 메서드

```
<script>
  $(document).ready(function () {
    $('h1').addClass('item');
  });
</script>
```

그림 15-2 클래스 추가

```
▼<body>
  <h1 class="item">Header-0</h1>
  <h1 class="item">Header-1</h1>
  <h1 class="item">Header-2</h1>
</body>
```

# 문서 객체의 클래스 속성 추가

- 함수를 매개 변수로 입력하기

코드 15-3 addClass() 메서드의 콜백 함수

```
<script>
  $(document).ready(function () {
    $('h1').addClass(function (index) {
      return 'class' + index;
    });
  });
</script>
```

그림 15-3 클래스가 순서대로 추가

```
▼<body>
  <h1 class="class0">Header-0</h1>
  <h1 class="class1">Header-1</h1>
  <h1 class="class2">Header-2</h1>
</body>
```



# 문서 객체의 클래스 속성 제거

## ■ 문서 객체의 클래스 속성 제거

표 15-2 jQuery의 메서드

메서드 이름	설명
removeClass()	문서 객체의 클래스 속성을 제거합니다.

코드 15-4 body 태그 구성

```
<body>
  <h1 class="item">Header-0</h1>
  <h1 class="item select">Header-1</h1>
  <h1 class="item">Header-2</h1>
</body>
```

코드 15-5 removeClass() 메서드

```
<script>
  $(document).ready(function () {
    $('h1').removeClass('select');
  });
</script>
```

그림 15-4 클래스 제거

```
▼<body>
  <h1 class="item">Header-0</h1>
  <h1 class="item">Header-1</h1>
  <h1 class="item">Header-2</h1>
</body>
```

# 문서 객체의 속성 검사

- 문서 객체 속성 검사
  - attr() 메서드로 특정 속성의 값을 알아내기
    - 매개 변수에 속성 이름 입력

코드 15-6 body 태그 구성

```
<body>
  
  
  
</body>
```

코드 15-7 attr() 메서드 - Getter

```
<script>
  $(document).ready(function () {
    // 변수를 선언합니다.
    var src = $('img').attr('src');

    // 출력합니다.
    alert(src);
  });
</script>
```

그림 15-5 attr() 메서드를 사용한 속성 검사



# 문서 객체의 속성 추가

## ■ attr() 메서드

- 문서 객체에 속성을 추가할 때도 사용
  - \$(selector).attr(name, value);
  - \$(selector).attr(name, function(index, attr){ });
  - \$(selector).attr(object);
- 속성 추가 첫 번째 방법

코드 15-8 body 태그구성

```
<body>
  
  
  
</body>
```

코드 15-9 attr() 메서드 - Setter (1)

```
<script>
  $(document).ready(function () {
    $('img').attr('width', 200);
  });
</script>
```

그림 15-6 그림 15-5 attr() 메서드를 사용한 속성 부여 (1)

```
▼<body>
  
  
  
</body>
```

# 문서 객체의 속성 추가

- attr() 메서드
  - 속성 추가 두 번째 방법

코드 15-10 attr() 메서드 - Setter (2)

```
<script>
$(document).ready(function () {
    $('img').attr('width', function (index) {
        return (index + 1) * 100;
    });
});
</script>
```

그림 15-7 attr() 메서드를 사용한 속성 부여 (2)



# 문서 객체의 속성 추가

- attr() 메서드
  - 속성 추가 세 번째 방법

코드 15-11 attr() 메서드 - Setter (3)

```
<script>
  $(document).ready(function () {
    $('img').attr({
      width: function (index) {
        return (index + 1) * 100;
      },
      height: 100
    });
  });
</script>
```

그림 15-8 그림 15-7 attr() 메서드를 사용한 속성 부여 (3)

```
▼ <body>
  
  
  
</body>
```

# 문서 객체의 속성 제거

## ▪ removeAttr() 메서드

- 첫 번째 매개 변수에 삭제하려는 속성의 이름 입력
- 예제 코드 15-13]
  - h1 태그에 있는 data-index 속성 제거

표 15-4 jQuery의 속성 제거 메서드

메서드 이름	설명
removeAttr(name)	문서 객체의 속성을 제거합니다.

코드 15-12 body 태그 구성

```
<body>
  <h1 data-index="0">Header-0</h1>
  <h1 data-index="1">Header-1</h1>
  <h1 data-index="2">Header-2</h1>
</body>
```

코드 15-13 removeAttr() 메서드

```
<script>
  $(document).ready(function () {
    $('h1').removeAttr('data-index');
  });
</script>
```

그림 15-9 removeAttr() 메서드를 사용한 속성 제거

```
▼<body>
  <h1>Header-0</h1>
  <h1>Header-1</h1>
  <h1>Header-2</h1>
</body>
```

# 문서 객체의 스타일 검사

## ■ css() 메서드

표 15-5 jQuery의 메서드

메서드 이름	설명
css()	스타일과 관련된 모든 기능을 수행합니다.

코드 15-14 HTML 페이지 구성

```
<!DOCTYPE html>
<html>
<head>
  <style>
    .first { color:Red; }
    .second { color:Pink; }
    .third { color:Orange; }
  </style>
  <script src="http://code.jquery.com/jquery-1.7.js" />
  <script>
    $(document).ready(function () {

      });
  </script>
</head>
<body>
  <h1 class="first">Header-0</h1>
  <h1 class="second">Header-1</h1>
  <h1 class="third">Header-2</h1>
</body>
</html>
```

# 문서 객체의 스타일 검사

## ■ 예제 코드 15-15

- css() 메서드의 매개 변수에 스타일 속성 이름 입력
- 코드 실행하면 첫 번째 문서 객체의 스타일 속성 가져옴

코드 15-15 css() 메서드 - Getter

```
<script>
$(document).ready(function () {
    // 변수를 선언합니다.
    var color = $('h1').css('color')

    // 출력합니다.
    alert(color);
});
</script>
```

그림 15-10 css() 메서드를 사용한 속성 검사





# 문서 객체의 스타일 추가

## ■ css 사용법

- \$(selector).css(name, value);
- \$(selector).css(name, function(index, style){ });
- \$(selector).css(object);
- attr() 메서드와 같은 방법

## ❖ \$(selector).css(name, value); 방법

코드 15-16 body 태그 구성

```
<body>
  <h1>Header-0</h1>
  <h1>Header-1</h1>
  <h1>Header-2</h1>
</body>
```

코드 15-17 css() 메서드 - Setter (1)

```
<script>
  $(document).ready(function () {
    $('h1').css('color', 'Red');
  });
</script>
```

# 문서 객체의 스타일 추가

- `$(selector).css(name, function(index, style){ });`

코드 15-18 `css()` 메서드 - Setter (2)

```
<script>
    $(document).ready(function () {
        // 변수를 선언합니다.
        var color = ['Red', 'White', 'Purple'];

        // 문서 객체의 스타일을 변경합니다.
        $('h1').css('color', function (index) {
            return color[index];
        });
    });
</script>
```

그림 15-11 `css()` 메서드를 사용한 스타일 속성 부여 (1)

```
▼<body>
  <h1 style="color: red; ">Header-0</h1>
  <h1 style="color: white; ">Header-1</h1>
  <h1 style="color: purple; ">Header-2</h1>
</body>
```

## 문서 객체의 스타일 추가

- `$(selector).css(object);`

코드 15-19 `css()` 메서드 - Setter (3)

```
<script>
    $(document).ready(function () {
        // 변수를 선언합니다.
        var color = ['Red', 'White', 'Purple'];

        // 문서 객체의 스타일을 변경합니다.
        $('h1').css({
            color: function (index) {
                return color[index];
            },
            backgroundColor: 'Black'
        });
    });
</script>
```

# 문서 객체의 내부 검사

- 문서 객체의 innerHTML 속성과 관련된 jQuery 메서드

표 15-6 jQuery의 메서드

메서드 이름	설명
html()	문서 객체 내부의 글자와 관련된 모든 기능을 수행합니다(HTML 태그 인식).
text()	문서 객체 내부의 글자와 관련된 모든 기능을 수행합니다.

코드 15-20 body 태그 구성

```
<body>
  <h1>Header-0</h1>
  <h1>Header-1</h1>
  <h1>Header-2</h1>
</body>
```

# 문서 객체의 내부 검사

## ■ 예제 코드 15-21

- html() 메서드 사용해 문서 객체의 내용 출력

코드 15-21 html() 메서드 - Getter

```
<script>
$(document).ready(function () {
    // 변수를 선언합니다.
    var html = $('h1').html();

    // 출력합니다.
    alert(html);
});
</script>
```

그림 15-12 html() 메서드를 사용한 객체 내부 문자 검사



# 문서 객체의 내부 검사

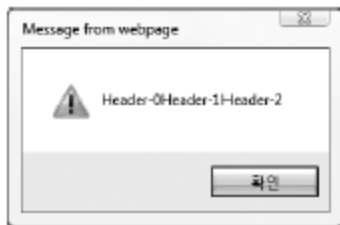
- text() 메서드 사용해 문서 객체의 내용 출력

코드 15-22 text() 메서드 - Getter

```
<script>
$(document).ready(function () {
    // 변수를 선언합니다.
    var text = $('h1').text();

    // 출력합니다.
    alert(text);
});
</script>
```

그림 15-13 text() 메서드를 사용한 객체 내부 문자 검사



# 문서 객체의 내부 추가

- 문서 객체의 내부에 내용물을 추가하고 싶을 때
  - html() 메서드와 text() 메서드 사용
    - 1 \$(selector).html(value);
    - \$(selector).text(value);
    - 2 \$(selector).html(function(index, html){ });
    - \$(selector).text(function(index, text){ });

코드 15-23 body 태그 구성

```
<body>
  <div></div>
  <div></div>
  <div></div>
</body>
```

# 문서 객체의 내부 추가

- html() 메서드 사용
  - 매개 변수로 입력하고자 하는 HTML 문자열 입력

코드 15-24 html() 메서드 - Setter (1)

```
<script>
  $(document).ready(function () {
    $('div').html('<h1>$.html() Method</h1>');
  });
</script>
```

그림 15-14 html() 메서드를 사용한 문서 객체 내부 문자 추가 (1)

```
▼<body>
  ▼<div>
    <h1>$.html() Method</h1>
  </div>
  ▼<div>
    <h1>$.html() Method</h1>
  </div>
  ▼<div>
    <h1>$.html() Method</h1>
  </div>
</body>
```



# 문서 객체의 내부 추가

## ■ text() 메서드 사용

코드 15-25 text() 메서드 - Setter

```
<script>
    $(document).ready(function () {
        $('div').text('<h1>$.html() Method</h1>');
    });
</script>
```

그림 15-15 text() 메서드를 사용한 문서 객체 내부 문자 추가

```
▼<body>
  <div>&lt;h1&gt;$.text() Method&lt;/h1&gt;</div>
  <div>&lt;h1&gt;$.text() Method&lt;/h1&gt;</div>
  <div>&lt;h1&gt;$.text() Method&lt;/h1&gt;</div>
</body>
```

# 문서 객체의 내부 추가

- 함수를 매개 변수로 입력
  - index를 매개 변수로

코드 15-26 html() 메서드 - Setter (2)

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

그림 15-16 html() 메서드를 사용한 문서 객체 내부 문자 추가 (2)

**Header-0**

**Header-1**

**Header-2**

## 문서 객체의 내부 추가

- html() 메서드의 매개변수로 입력한 함수
  - 두 번째 매개변수 사용시

코드 15-27 body 태그 구성

```
<body>
  <h1>Header-0</h1>
  <h1>Header-1</h1>
  <h1>Header-2</h1>
</body>
```

코드 15-28 html() 메서드 - Setter (3)

```
<script>
  $(document).ready(function () {
    $('h1').html(function (index, html) {
      return '★' + html + '★';
    });
  });
</script>
```

그림 15-17 html() 메서드를 사용한 문서 객체 내부 문자 추가 (3)

**★Header-0★**  
**★Header-1★**  
**★Header-2★**

# 문서 객체의 제거

- 문서 객체의 제거
  - remove()는 한 개의 객체만 지움

표 15-7 jQuery의 문서 객체 제거 메서드

메서드 이름	설명
remove()	특정 문서 객체를 제거합니다.
empty()	특정 문서 객체 의 후손을 모두 제거합니다.

코드 15-29 body 태그 구성

```
<body>
  <div>
    <h1>Header-0</h1>
    <h1>Header-1</h1>
  </div>
</body>
```

코드 15-30 remove() 메서드

```
<script>
  $(document).ready(function () {
    $('h1').first().remove();
  });
</script>
```

그림 15-18 remove() 메서드를 사용한 문서 객체 제거

**Header-1**

# 문서 객체의 제거

- 문서 객체의 제거
  - div 태그에 empty() 메서드 적용
    - body 태그의 모든 div 태그 제거

코드 15-31 empty() 메서드

```
<script>
    $(document).ready(function () {
        $('div').empty();
    });
</script>
```

# 문서 객체의 생성 (1)

## ■ 텍스트 노드를 갖는 문서 객체 생성

표 15-8 jQuery의 메서드

메서드 이름	설명
<code>\$()</code>	문서 객체를 생성합니다.

코드 15-35 문서 객체 연결

```
<script>
$(document).ready(function () {
    $('<h1>X/h1>').html('Hello World .. !').appendTo('body');
});
</script>
```

그림 15-19 텍스트 노드를 갖는 문서 객체 생성

**Hello World .. !**

코드 15-36 간단한 문서 객체 생성

```
<script>
$(document).ready(function () {
    $('<h1>Hello World .. !</h1>').appendTo('body');
});
</script>
```

## 문서 객체의 생성 (2)

- 텍스트 노드 갖지 않는 문서 객체 생성
  - img 태그 생성시
    - \$() 메서드 사용해 문서 객체 생성
    - attr() 메서드 사용해 속성 입력

코드 15-37 문서 객체의 생성과 속성 지정

```
<script>
    $(document).ready(function () {
        $('<img />').attr('src', 'Chrysanthemum.jpg').appendTo('body');
    });
</script>
```

- 좀 더 간단한 태그 생성법

코드 15-38 문서 객체 속성 지정

```
<script>
    $(document).ready(function () {
        $('<img />', {
            src: 'Chrysanthemum.jpg',
            width: 350,
            height: 250
        }).appendTo('body');
    });
</script>
```

그림 15-20 속성을 갖는 문서 객체의 생성

```
▼<body>
    
</body>
```

# 문서 객체의 삽입 (1)

- 문서 객체에 문서 객체를 추가하는 메서드
  - 삽입 메서드 Part 1

표 15-9 jQuery의 문서 객체 삽입 메서드 (1)

메서드 이름	설명
<code>\$(A).appendTo(B)</code>	A를 B의 뒷 부분에 추가합니다.
<code>\$(A).prependTo(B)</code>	A를 B의 앞 부분에 추가합니다.
<code>\$(A).insertAfter(B)</code>	A를 B의 뒤에 추가합니다.
<code>\$(A).insertBefore(B)</code>	A를 B의 앞에 추가합니다.

그림 15-21 삽입 메서드 정리





## 문서 객체의 삽입 (2)

- 문서 객체에 문서 객체를 추가하는 메서드
  - 삽입 메서드 Part 2
    - Part 1과 반대의 순서로 메서드 진행

표 15-10 jQuery의 문서 객체삽입 메서드 (2)

메서드 이름	설명
<code>\$(A).append(B)</code>	B를 A의 뒷 부분에 추가합니다.
<code>\$(A).prepend(B)</code>	B를 A의 앞 부분에 추가합니다.
<code>\$(A).after(B)</code>	B를 A의 뒤에 추가합니다.
<code>\$(A).before(B)</code>	B를 A의 앞에 추가합니다.

## 문서 객체의 삽입 (2)

### ▪ Append() 메서드 활용

- 1 `$(selector).append(content, content, ..... , content)`
  - 1번 형태는 content 위치에 문자열이 들어갈 수 있음
  - jQuery 문서 객체도 들어갈 수 있음
- 2 `$(selector).append(function (index) { });`

코드 15-39 body 태그 구성

```
<body>
```

```
</body>
```

## 문서 객체의 삽입 (2)

- Append() 메서드 활용
  - 두 번째 형태 활용
    - 선택한 객체에 서로 다른 내용 추가 (배열 사용)

코드 15-41 body 태그 구성

```
<body>
  <div></div>
  <div></div>
  <div></div>
</body>
```

코드 15-42 append() 메서드

```
<script>
  $(document).ready(function () {
    // 변수를 선언합니다.
    var content = [
      { name: '윤인성', region: '서울특별시 강서구' },
      { name: '윤하린', region: '서울특별시 광진구' },
      { name: '윤인아', region: '미국 메사추세츠' }
    ];

    // 문서 객체를 추가합니다.
    $('div').append(function (index) {

    });
  });
</script>
```

## 문서 객체의 삽입 (2)

- Append() 메서드 활용
  - append() 메서드 부분에 문서 객체 만들어 리턴
    - 각 div 태그에 서로 다른 문서 객체 추가

코드 15-43 배열을 사용한 문서 객체 생성과 추가

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

    return output;
});
```

그림 15-23 배열과 append() 메서드를 사용한 문서 객체 추가

```
▼ <body>
  ▼ <div>
    <h1>윤인성</h1>
    <h2>서울특별시 강남구</h2>
  </div>
  ▼ <div>
    <h1>윤하린</h1>
    <h2>서울특별시 광진구</h2>
  </div>
  ▼ <div>
    <h1>윤원아</h1>
    <h2>미국 메사추세츠</h2>
  </div>
</body>
```

# 문서 객체의 이동

- 기존 존재 문서 객체 선택하고 문서 객체 삽입 메서드 사용
  - 문서 객체 쉽게 다른 곳으로 이동 가능

코드 15-45 appendTo() 메서드를 사용한 문서 객체의 이동

```
<script>
    $(document).ready(function () {
        $('img').first().appendTo('body');
    });
</script>
```

코드 15-46 지속적 이미지 순서 변경

```
<script>
    $(document).ready(function () {
        // .image의 크기를 조정합니다.
        $('img').css('width', 250);

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

그림 15-24 첫 번째 이미지가 2초마다 맨 뒤로 이동



# 문서 객체의 복제

- 기존 문서 객체 선택하고 문서 객체 삽입 메서드 사용
  - 문서 객체 이동

코드 15-47 append() 메서드를 사용한 문서 객체의 이동

```
<!DOCTYPE html>
<html>
<head>
  <script src="http://code.jquery.com/jquery-1.7.js"×/script>
  <script>
    $(document).ready(function () {
      $('div').append($('h1'));
    });
  </script>
</head>
<body>
  <h1>HEADER</h1>
  <hr/×div×/div×hr/>
</body>
</html>
```

그림 15-25 문서 객체가 이동

---

**HEADER**

---

# 문서 객체의 복제

## ■ 문서 복제해서 추가

### – jQuery의 clone() 메서드 사용

- 쉽게 문서 객체 복제 가능
- 1 \$(selector).clone()
- 2 \$(selector).clone(Boolean dataAndEvents)
- 3 \$(selector).clone(Boolean dataAndEvents, Boolean deepDataAndEvents)

표 15-11 jQuery의 메서드

메서드 이름	설명
clone()	문서 객체를 복제합니다.

코드 15-48 clone() 메서드

```
<script>
  $(document).ready(function () {
    $('div').append($('h1').clone());
  });
</script>
```

그림 15-26 문서 객체를 복제해 삽입

**HEADER**

**HEADER**

# 이벤트 연결

- jQuery 사용해 이벤트 연결
  - 가장 기본적인 방법은 bind() 메서드 사용
  - bind() 메서드는 다음과 같은 두 가지 형태로 사용
    - 1 \$(selector).bind(eventName, function(event) { })
    - 2 \$(selector).bind(object)

코드 16-2 body 태그 구성

```
<body>
  <h1>Header-0</h1>
  <h1>Header-1</h1>
  <h1>Header-2</h1>
</body>
```



# 이벤트 연결

- `$(selector).bind(eventName, function(event) { })`
  - `bind()` 메서드의 첫 번째 매개 변수에 이벤트 이름 입력
  - 두 번째 매개 변수에 이벤트 핸들러 입력
  - 이벤트 핸들러 안에서 `this` 키워드는 이벤트 발생한 객체 의미
  - 예제 코드 16-3]
    - `h1` 태그를 `click` 이벤트에 연결
    - 이벤트 발생 시 이벤트 발생 객체에 ‘+’ 글자 추가

코드 16-3 `bind()` 메서드(1)

```
<script>
  $(document).ready(function () {
    // 이벤트를 연결합니다.
    $('h1').bind('click', function () {
      $(this).html(function (index, html) {
        return html + '+';
      });
    });
  });
</script>
```

# 이벤트 연결

## ■ 2 \$(selector).bind(object)

- bind() 메서드의 매개 변수에 객체 넣어줌
- 속성 이름과 속성값에 이벤트 이름과 이벤트 핸들러 넣음
- 예제 코드 16-4]
  - 마우스가 진입하면 reverse 클래스 속성 추가
  - 마우스가 빠져나가면 reverse 클래스 속성 제거

코드 16-4 bind() 메서드(2)

```
<script>
    $(document).ready(function () {
        // 이벤트를 연결합니다.
        $('h1').bind('click', function () {
            $(this).html(function (index, html) {
                return html + '+';
            });
        });

        // 이벤트를 연결합니다.
        $('h1').bind({
            mouseenter: function () { $(this).addClass('reverse'); },
            mouseleave: function () { $(this).removeClass('reverse'); }
        });
    });
</script>
```

코드 16-5 style 태그 구성

```
<style>
    .reverse {
        background: Black;
        color: White;
    }
</style>
```

# 간단한 이벤트 연결

## ■ jQuery 단축 이벤트 메서드

### – 사용법

- `$(selector).method(function(event) { });`

표 16-2 jQuery 단축 이벤트 메서드

blur	focus	focusin	focusout	load
resize	scroll	unload	click	dblclick
mousedown	mouseup	mousemove	mouseover	mouseout
mouseenter	mouseleave	change	select	submit
keydown	keypress	keyup	error	ready

표 16-3 jQuery의 메서드

메서드 이름	설명
<code>toggle()</code>	click 이벤트를 여러 이벤트 핸들러를 번갈아가며 실행할 수 있게 연결합니다.
<code>hover()</code>	mouseenter 이벤트와 mouseleave 이벤트를 동시에 연결합니다.

# 간단한 이벤트 연결

- jQuery 단축 이벤트 메서드 구현
  - HTML 페이지 구성

코드 16-6 HTML 페이지 구성

```
<!DOCTYPE html>
<html>
<head>
  <style>
    .reverse {
      background: Black;
      color: White;
    }
  </style>
  <script src="http://code.jquery.com/jquery-1.7.js" />
  <script>
    $(document).ready(function () {

      });
  </script>
</head>
<body>
  <h1>Header-0</h1>
  <h1>Header-1</h1>
  <h1>Header-2</h1>
</body>
</html>
```

## 간단한 이벤트 연결

- hover() 메서드
  - 두 개의 함수를 매개 변수로 넣어줌
    - 첫 번째 매개 변수에 넣는 함수는 mouseenter 이벤트 핸들러
    - 두 번째 매개 변수에 넣는 함수는 mouseleave 이벤트 핸들러

코드 16-7 hover() 메서드

```
<script>
    $(document).ready(function () {
        // 이벤트를 연결합니다.
        $('h1').hover(function () {
            $(this).addClass('reverse');
        }, function () {
            $(this).removeClass('reverse');
        });
    });
</script>
```

# 간단한 이벤트 연결

## ■ toggle() 메서드

- 매개 변수에 여러 개의 함수 입력 가능 (click 이벤트 핸들러와 관련)

코드 16-8 toggle() 메서드

```
<script>
$(document).ready(function () {
    // 이벤트를 연결합니다.
    $('h1').hover(function () {
        $(this).addClass('reverse');
    }, function () {
        $(this).removeClass('reverse');
    });

    // 이벤트를 연결합니다.
    $('h1').toggle(function () {
        // 문자열의 앞에 별을 추가합니다.
        $(this).html(function (index, html) {
            return '★' + html;
        });
    }, function () {
        // 문자열의 뒤에 별을 추가합니다.
        $(this).html(function (index, html) {
            return html + '★';
        });
    }, function () {
        // 문자열의 별을 제거합니다.
        $(this).html(function (index, html) {
            return html.substring(1, html.length - 1);
        });
    });
});
</script>
```

# 이벤트 연결 제거

## ■ 이벤트 제거 시 unbind() 메서드 사용

### – unbind() 메서드의 사용 형태

- 1 `$(selector).unbind()` - 해당 문서 객체와 관련된 모든 이벤트 제거
- 2 `$(selector).unbind(eventName)` - 특정 이벤트와 관련된 모든 이벤트 제거
- 3 `$(selector).unbind(eventName, function)` - 특정 이벤트 핸들러 제거

표 16-4 jQuery의 메서드

메서드 이름	설명
<code>unbind()</code>	이벤트를 제거합니다.

### – body 태그 구성

코드 16-9 body 태그 구성

```
<body>
  <h1>Header-0</h1>
  <h1>Header-1</h1>
  <h1>Header-2</h1>
</body>
```

# 이벤트 연결 제거

## ■ 예제 코드 16-10]

- h1 태그에 click 이벤트 연결하고 이벤트 발생 알람
- 이후에 unbind() 메서드 사용해 이벤트 제거

코드 16-10 unbind() 메서드

```
<script>
$(document).ready(function () {
    // 이벤트를 연결합니다.
    $('h1').click(function () {
        // 출력합니다.
        $(this).html('CLICK');
        alert('이벤트가 발생했습니다.');
```

```
        // 이벤트를 제거합니다.
        $(this).unbind();
    });
});
</script>
```

표 16-5 jQuery의 메서드

메서드 이름	설명
one()	이벤트를 한번만 연결합니다.



# 매개 변수 context

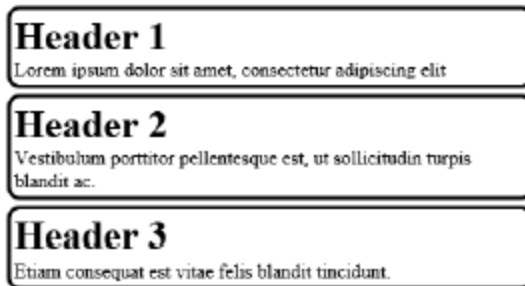
## ■ 매개 변수 context

- jQuery 메서드는 매개 변수 두 개 입력 가능
- selector 매개 변수 이외에 context 매개 변수 가능
- 매개 변수 context = selector 적용하는 범위 한정

그림 16-4 jQuery 메서드의 형태

```
$(  
  $(selector, context)
```

그림 16-5 예제



# 매개 변수 context

- 매개 변수 context 예제
  - 특정 부분에 선택자 적용하고 싶을 때 사용하는 매개 변수
  - Body 태그 구성

코드 16-11 body 태그 구성

```
<body>
  <div>
    <h1>Header 1</h1>
    <p>Paragraph</p>
  </div>
  <div>
    <h1>Header 2</h1>
    <p>Paragraph</p>
  </div>
  <div>
    <h1>Header 3</h1>
    <p>Paragraph</p>
  </div>
</body>
```

---

# 매개 변수 context

- 매개 변수 context 예제
  - Style 태그 구성

코드 16-12 style 태그 구성

```
<style>
  * { margin:0px; padding:0px }
  div
  {
    margin:5px; padding:3px;

    border-width:3px;
    border-style:solid;
    border-color:Black;
    border-radius:10px;
  }
</style>
```

---

# 매개 변수 context

- 매개 변수 context 예제
  - Context 객체 사용

코드 16-13 context 객체 사용

```
<script>
    $(document).ready(function () {
        // 이벤트를 연결합니다.
        $('div').click(function () {
            // 변수를 선언합니다.
            var header = $('h1', this).text();
            var paragraph = $('p', this).text();

            // 출력합니다.
            alert(header + '\n' + paragraph);
        });
    });
</script>
```

# 이벤트 객체

- jQuery가 이벤트 객체 정형화
  - jQuery의 이벤트 객체는 모든 브라우저가 같은 방법으로 사용
  - 같은 속성을 가짐
- 자주 사용하는 jQuery 이벤트 객체의 속성

표 16-6 jQuery 이벤트 객체의 속성

이벤트 객체 속성	설명
event.pageX	브라우저의 화면을 기준으로 한 마우스의 X 좌표 위치
event.pageY	브라우저의 화면을 기준으로 한 마우스의 Y 좌표 위치
event.preventDefault()	기본 이벤트를 제거합니다.
event.stopPropagation()	이벤트 전달을 제거합니다.

# 이벤트 객체

## ■ 이벤트 객체 활용 예제

- HTML5 Canvas 사용하는 예제 작성
  - canvas 태그에 id 속성 입력
  - width 속성 과 height 속성 지정
- canvas 태그는 HTML5부터 지원하는 태그
  - 인터넷 익스플로러 8 이하에서는 지원 불가
- body 태그 구성

코드 16-14 body 태그 구성

```
<body>  
  <canvas id="canvas" width="500" height="300">  
  
  </canvas>  
</body>
```

# 이벤트 객체

- 이벤트 객체 활용 예제
  - Document 객체의 getElementById() 메서드 사용
    - canvas 태그에 id 속성 부여

코드 16-15 2d context 객체 추출과 이벤트 연결

```
<script>
    $(document).ready(function () {
        // 변수를 선언합니다.
        var canvas = document.getElementById('canvas');
        var context = canvas.getContext('2d');

        // 이벤트를 연결합니다.
        $(canvas).toggle(function (event) {

            }, function (event) {

            });
    });
</script>
```

# 이벤트 객체

## ■ 예제 코드 16-16]

- 문서 객체와 마우스 커서의 상대 위치 구하는 방법
  - 이벤트 객체에서 pageX 속성과 pageY 속성 사용

코드 16-16 jQuery 이벤트 객체

```
// 이벤트를 연결합니다.
$(canvas).toggle(function (event) {
    // 위치를 얻어냅니다.
    var position = $(this).offset();
    var x = event.pageX - position.left;
    var y = event.pageY - position.top;
}, function (event) {
    // 위치를 얻어냅니다.
    var position = $(this).offset();
    var x = event.pageX - position.left;
    var y = event.pageY - position.top;
});
```



# 이벤트 객체

- context 객체의 beginPath(), moveTo(),.lineTo(), stroke() 메서드 사용
  - canvas 객체 에 선 그릴 수 있음

코드 16-17 Canvas를 사용한 선 그리기

```
// 이벤트를 연결합니다.
$(canvas).toggle(function (event) {
    // 위치를 얻어냅니다.
    var position = $(this).offset();
    var x = event.pageX - position.left;
    var y = event.pageY - position.top;

    // 선을 그립니다.
    context.beginPath();
    context.moveTo(x, y);

}, function (event) {
    // 위치를 얻어냅니다.
    var position = $(this).offset();
    var x = event.pageX - position.left;
    var y = event.pageY - position.top;

    // 선을 그립니다.
    context.lineTo(x, y);
    context.stroke();
});
```

그림 16-6 실행 결과



# 이벤트 강제 발생

## ■ 이벤트 강제 발생

### – trigger() 메서드 사용 예

- 1 \$(selector).trigger(eventName)
- 2 \$(selector).trigger(eventName, data)

– 2번 형태의 매개 변수 data는 일반적으로 배열

표 16-7 jQuery의 메서드

메서드 이름	설명
trigger()	이벤트를 강제로 발생시킵니다.

### – 1번 형태 예제 Body 태그 구성

코드 16-18 body 태그 구성

```
<body>
  <h1>Start: </h1>
  <h1>Start: </h1>
</body>
```

# 이벤트 강제 발생

## ▪ trigger( ) 메서드

- h1 태그에 click 이벤트 연결
  - 클릭하면 뒤에 별 추가
  - setInterval() 함수 사용
    - 마지막에 위치한 h1 태그는 자동으로 1초마다 클릭 이벤트 실행

코드 16-19 trigger() 메서드

```
<script>
    $(document).ready(function () {
        // 이벤트를 연결합니다.
        $('h1').click(function () {
            $(this).html(function (index, html) {
                return html + '★';
            });
        });

        // 1초마다 함수를 실행합니다.
        setInterval(function () {
            $('h1').last().trigger('click');
        }, 1000);
    });
</script>
```

그림 16-7 실행 결과

Start: ★★★★★★★★★★

Start: ★★★★★

# 이벤트 강제 발생

- 간단한 방식으로 이벤트 강제 실행

코드 16-20 간단한 이벤트 강제 실행

```
// 1초마다 함수를 실행합니다.  
setInterval(function () {  
    $('h1').last().click();  
}, 1000);
```

---

# 이벤트 강제 발생

- trigger( ) 메서드의 2번 형태 실행
  - 2 \$(selector).trigger(eventName, data)

코드 16-21 trigger() 메서드를 사용한 매개 변수 전달

```
<script>
$(document).ready(function () {
    // 이벤트를 연결합니다.
    $('h1').click(function (event, data1, data2) {
        alert(data1 + ' : ' + data2);
    });

    // 이벤트를 강제로 발생시킵니다.
    $('h1').eq(1).trigger('click', [273, 52]);
});
</script>
```

# 기본 이벤트와 이벤트 전달

## ■ 기본 이벤트 제거, 이벤트 전달을 막는 메서드

표 16-8 이벤트 객체의 메서드

메서드 이름	설명
<code>preventDefault()</code>	기본 이벤트를 제거합니다.
<code>stopPropagation()</code>	이벤트 전달을 제거합니다.

코드 16-22 body 태그 구성

```
<body>
  <h1>
    <a href="http://hanb.co.kr">Hanb Media</a>
  </h1>
</body>
```

코드 16-23 style 태그 구성

```
<style>
  * {
    margin:5px; padding:5px;
    border:3px solid Black;
  }
</style>
```

# 기본 이벤트와 이벤트 전달

## ■ 예제 코드 16-24]

### – preventDefault() 메서드

- H1 태그 눌렀을 때 자동 실행되는 a 태그 기본 이벤트 제거

코드 16-24 preventDefault() 메서드

```
<script>
  $(document).ready(function () {
    $('a').click(function (event) {
      $(this).css('background-color', 'Blue');
      event.preventDefault();
    });

    $('h1').click(function () {
      $(this).css('background-color', 'Red');
    });
  });
</script>
```

그림 16-8 anchor 태그를 클릭했을 경우의 실행 결과



# 기본 이벤트와 이벤트 전달

- 예제 코드 16-25
  - 이벤트 객체의 stopPropagation() 메서드 사용

코드 16-25 stopPropagation() 메서드

```
$('#a').click(function (event) {  
    $(this).css('background-color', 'Blue');  
    event.stopPropagation();  
    event.preventDefault();  
});
```

그림 16-9 anchor 태그를 클릭했을 경우의 실행 결과





# 기본 이벤트와 이벤트 전달

## ■ 예제 코드 16-26

- stopPropagation() 메서드와 preventDefault() 메서드 함께 사용
- jQuery는 return false 사용
  - 이 두 가지 함께 적용하는 것으로 인식

코드 16-26 return false

```
$('#a').click(function (event) {  
    $(this).css('background-color', 'Blue');  
    return false;  
});
```

# 이벤트 연결의 종류

## ■ jQuery의 이벤트 연결 메서드

표 16-9 jQuery의 이벤트 연결 메서드

메서드 이름	설명
<code>bind()</code>	현재 존재하는 문서 객체에만 이벤트를 연결합니다.
<code>delegate()</code>	현재 또는 미래에 존재하는 문서 객체에 이벤트를 연결합니다.
<code>live()</code>	현재 또는 미래에 존재하는 문서 객체에 이벤트를 연결합니다.
<code>one()</code>	한번만 이벤트에 연결합니다.

## ■ jQuery의 이벤트 제거 메서드

표 16-10 이벤트 연결 제거 관련 메서드

메서드 이름	설명
<code>unbind()</code>	<code>bind()</code> 메서드를 사용해 연결한 이벤트를 제거합니다.
<code>die()</code>	<code>live()</code> 메서드를 사용해 연결한 이벤트를 제거합니다.
<code>undelegate()</code>	<code>delegate()</code> 메서드를 사용해 연결한 이벤트를 제거합니다.

# 이벤트 연결의 종류

- 이벤트 제거 메서드 비교 예제
  - Body 태그 작성

코드 16-27 body 태그 구성

```
<body>
  <div id="wrap">
    <h1>Header</h1>
  </div>
</body>
```

- bind( ) 메서드를 사용한 이벤트 연결

코드 16-28 bind() 메서드를 사용한 이벤트 연결

```
<script>
  $(document).ready(function () {
    $('h1').bind('click', function () {
      var length = $('h1').length;
      var targetHTML = $(this).html();
      $('#wrap').append('<h1>' + length + ' - ' + targetHTML + '</h1>');
    });
  });
</script>
```

그림 16-10 Header를 클릭하면 생성됩니다.

**Header**

**1 - Header**

**2 - Header**

**3 - Header**

# 이벤트 연결의 종류

## ■ 이벤트 제거 메서드 비교 예제

### – live() 메서드

- 현재 존재하는 h1 태그 + 앞으로 계속 생성하는 h1 태그에도 이벤트 연결

코드 16-29 live() 메서드를 사용한 이벤트 연결

```
<script>
    $(document).ready(function () {
        $('h1').live('click', function () {
            var length = $('h1').length;
            var targetHTML = $(this).html();
            $('#wrap').append('<h1>' + length + ' - ' + targetHTML + '</h1>');
        });
    });
</script>
```

그림 16-11 어떠한 h1 태그를 선택해도 요소가 추가됨

**Header**

**1 - Header**

**2 - 1 - Header**

**3 - 2 - 1 - Header**

**4 - 1 - Header**

# 이벤트 연결의 종류

## ■ 이벤트 제거 메서드 비교 예제

### – Die() 메서드

#### • 사용 방식

- 1 `$(selector).die()`
- 2 `$(selector).die(eventName)`
- 3 `$(selector).die(eventName, function)`

#### • 매개 변수 지정하지 않으면 해당 선택자와 관련된 모든 이벤트 제거

코드 16-30 die() 메서드

```
<script>
    $(document).ready(function () {
        // 이벤트를 연결합니다.
        $('h1').live('click', function () {
            var length = $('h1').length;
            var targetHTML = $(this).html();
            $('#wrap').append('<h1>' + length + ' - ' + targetHTML + '</h1>');
        });

        // 5초 이후에는 모든 이벤트가 제거됩니다.
        setTimeout(function () {
            $('h1').die();
        }, 5000);
    });
</script>
```

# 이벤트 통합 메서드

## ■ jQuery 1.7의 이벤트 연결 메서드

표 16-11 jQuery 1.7의 이벤트 연결 메서드

메서드 이름	설명
on()	이벤트를 연결합니다.
off()	이벤트를 제거합니다.

- on() 메서드의 기존 이벤트 연결 메서드 대체
  - 1 bind(): \$(selector).on(eventName, eventHandler)
  - 2 live(): \$(document).on(eventName, selector, eventHandler)
  - 3 delegate(): \$(selector).on(eventName, selector, eventHandler)
- off() 메서드의 기존 이벤트 연결 메서드 대체
  - 이벤트의 모든 이벤트 핸들러 제거하고 싶을 때
    - 매개 변수 eventHandler 생략
  - 1 unbind(): \$(selector).off(eventName[, eventHandler])
  - 2 die(): \$(document).off(eventName, selector[, eventHandler])
  - 3 undelegate(): \$(selector).off(eventName, selector[, eventHandler])

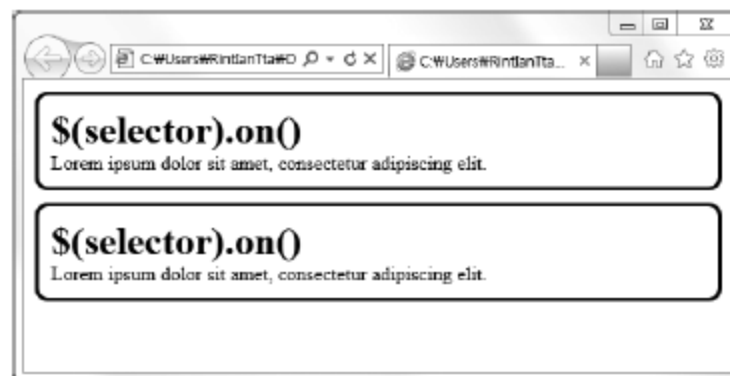
# 이벤트 통합 메서드

## ■ on( ) 메서드의 bind( ) 메서드 대체 형태

코드 16-31 on() 메서드의 bind() 메서드 대체 형태

```
<!DOCTYPE html>
<html>
<head>
  <style>
    * { margin:0px; padding:0px;}
    div {
      margin:10px; padding:10px;
      border:3px Solid Black;
      border-radius:10px;
    }
  </style>
  <script src="http://code.jquery.com/jquery-1.7b1.js"></script>
  <script>
    $(document).ready(function () {
      $('div').on('click', function () {
        $(this).clone().appendTo('body');
      });
    });
  </script>
</head>
<body>
<div>
  <h1>$(selector).on()</h1>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
</div>
</body>
</html>
```

그림 16-12 bind() 메서드의 대체 형태



# 이벤트 통합 메서드

- on( ) 메서드의 delegate( ) 메서드 대체 형태
  - 선택자 부분에서 이벤트 지속 범위 한정
    - 첫 번째 매개 변수에 이벤트 이름
    - 두 번째 매개 변수에 이벤트가 지정될 선택자 입력
    - 세 번째 매개 변수에 이벤트 핸들러 지정

코드 16-32 on() 메서드의 delegate() 메서드 대체 형태

```
<script>
    $(document).ready(function () {
        $('body').on('click', 'div', function () {
            $(this).clone().appendTo('body');
        });
    });
</script>
```



# 이벤트 통합 메서드

## ■ 이벤트 지속 범위 한정

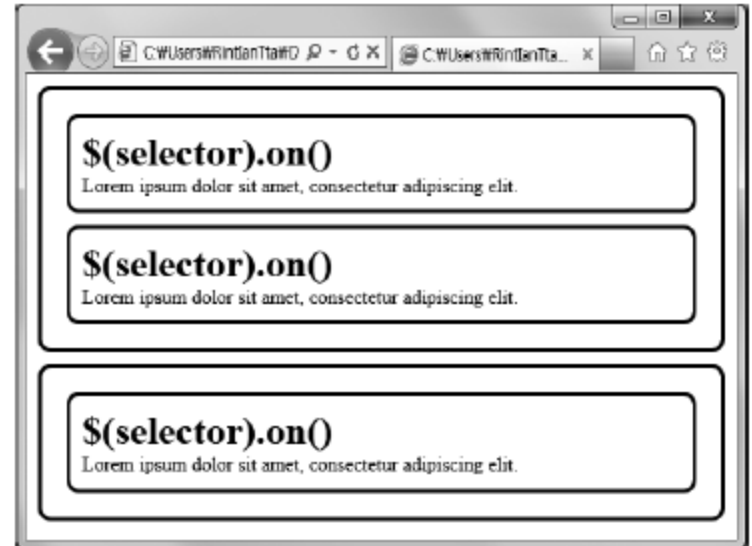
### – 예제 코드 16-33

- id 속성을 upper\_div로 지정한 div 태그, lower\_div로 지정한 div 태그
- on() 메서드를 delegate() 메서드의 대체 형태로 사용
- 선택자 부분에서 이벤트 지속 범위를 id 속성을 upper\_div로 지정한 태그로 한정

코드 16-33 이벤트 범위 한정

```
<script>
$(document).ready(function () {
    $('#upper_div').on('click', 'div', function () {
        $(this).clone().appendTo('#upper_div');
        $(this).clone().appendTo('#lower_div');
    });
});
</script>
```

그림 16-13 delegate() 메서드의 대체 형태



# 이벤트 통합 메서드

- `delegate()` 메서드
  - 이벤트의 지속 범위 한정
- `live()` 메서드
  - 문서 전체가 이벤트의 지속 범위
  - 예제 코드 16-34
    - 선택자에 `document` 객체 지정

코드 16-34 `on()` 메서드의 `live()` 메서드 대체 형태

```
<script>
    $(document).ready(function () {
        $(document).on('click', 'div', function () {
            $(this).clone().appendTo( '#upper_div' );
            $(this).clone().appendTo( '#lower_div' );
        });
    });
</script>
```

# 마우스 이벤트

## ■ 마우스 이벤트

표 16-12 마우스 이벤트

이벤트 이름	설명
click	마우스를 클릭할 때 발생합니다.
dblclick	마우스를 더블클릭할 때 발생합니다.
mousedown	마우스 버튼을 누를 때 발생합니다.
mouseup	마우스 버튼을 떼를 때 발생합니다.
mouseenter	마우스가 요소의 경계 외부에서 내부로 이동할 때 발생합니다.
mouseleave	마우스가 요소의 경계 내부에서 외부로 이동할 때 발생합니다.
mousemove	마우스를 움직일 때 발생합니다.
mouseout	마우스가 요소를 벗어날 때 발생합니다.
mouseover	마우스를 요소 안에 들어올 때 발생합니다.

# 마우스 이벤트

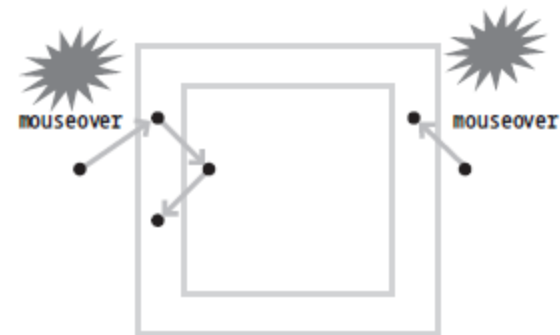
## ■ 예제 코드 16-35]

- div 태그를 두 개 중첩해서 만든 후
- 외부 div 태그에 mouseover 이벤트와 mouseenter 이벤트 모두 연결

코드 16-35 mouseover와 mouseenter 이벤트의 차이

```
<!DOCTYPE html>
<html>
<head>
  <style>
    .outer { width:200px; height:200px; background:Orange; padding:50px; margin:10px;}
    .inner { width:100%; height:100%; background:Pink; }
  </style>
  <script src="http://code.jquery.com/jquery-1.7.js" />
  <script>
    $(document).ready(function(){
      $('.outer').mouseover(function(){
        $('body').append('<h1>MOUSEOVER</h1>');
      }).mouseenter(function(){
        $('body').append('<h1>MOUSEENTER</h1>');
      });
    });
  </script>
</head>
<body>
  <div class="outer">
    <div class="inner" />
  </div>
</body>
</html>
```

그림 16-14 mouseover와 mouseenter 이벤트의 차이



# 키보드 이벤트

- 키보드 이벤트
  - keypress 이벤트는 한글로 사용할 수 없음

표 16-13 키보드 이벤트


이벤트 이름	설명
keydown	키보드가 눌려질 때 발생합니다..
keypress	글자가 입력될 때 발생합니다.
keyup	키보드가 떼어질 때 발생합니다.

그림 16-15 마투데이의 글쓰기

지금 내 생각을

마티마투에

150



✓ 올리기

# 키보드 이벤트

- 동적으로 글자 수 세기
  - 키보드 이벤트와 키보드 이벤트의 주의할 점 숙지
  - HTML 페이지 구성

코드 16-36 HTML 페이지 구성

```
<!DOCTYPE html>
<html>
<head>
  <script src="http://code.jquery.com/jquery-1.7.js">/script>
  <script>
  </script>
</head>
<body>
  <div>
    <p>지금 내 생각을</p>
    <h1>150</h1>
    <textarea cols="70" rows="5">/textarea>
  </div>
</body>
</html>
```

## 16.11 키보드 이벤트

- textarea 태그에 keyup 이벤트 연결
  - keyup 이벤트 발생하면 글자 개수 받아 출력

코드 16-37 keyup 이벤트 연결

```
<script>
    $(document).ready(function (event) {
        $('textarea').keyup(function () {
            // 남은 글자 수를 구합니다.
            var inputLength = $(this).val().length;
            var remain = 150 - inputLength;

            // 문서 객체에 입력합니다.
            $('h1').html(remain);
        });
    });
</script>
```

# 키보드 이벤트

- 키보드 이벤트의 실행 순서
  - 1. 사용자가 키보드 누름
  - 2. keydown 이벤트 발생
  - 3. 글자 입력
  - 4. keypress 이벤트 발생
  - 5. 사용자가 키보드에서 손을 뗀
  - 6. keyup 이벤트 발생
  - keydown 이벤트가 발생한 순간에는 글자가 입력돼 있지 않음
  - 입력한 글자 수를 표시해야 하므로 keyup 이벤트 사용



# 키보드 이벤트

## ■ 글자 개수에 따른 스타일 변경

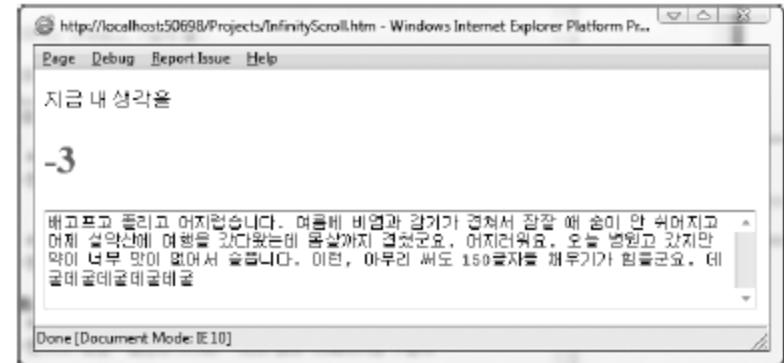
코드 16-38 글자 개수에 따른 스타일 변경

```
<script>
    $(document).ready(function (event) {
        $('textarea').keyup(function () {
            // 남은 글자 수를 구합니다.
            var inputLength = $(this).val().length;
            var remain = 150 - inputLength;

            // 문서 객체에 입력합니다.
            $('h1').html(remain);

            // 문서 객체의 색상을 변경합니다.
            if (remain >= 0) {
                $('h1').css('color', 'Black');
            } else {
                $('h1').css('color', 'Red');
            }
        });
    });
</script>
```

그림 16-16 실행 결과



# 윈도우 이벤트

## ■ 윈도우 이벤트란?

- window 객체와 document 객체 이외에 img 태그 등이 사용할 수 있는 이벤트

표 16-14 윈도우 이벤트

이벤트	설명
ready	문서 객체가 준비를 완료합니다.
load	윈도우(문서 객체)를 불러들일 때 발생합니다.
unload	윈도우(문서 객체)를 닫을 때 발생합니다.
resize	윈도우의 크기를 변화시킬 때 발생합니다.
scroll	윈도우를 스크롤할 때 발생합니다.
error	에러가 있을 때 발생합니다.

# 입력 양식 이벤트

## ■ 입력 양식 이벤트

표 16-15 입력 양식 이벤트

이벤트 이름	설명
change	입력 양식의 내용을 변경할 때 발생합니다.
focus	입력 양식에 초점을 맞추면 발생합니다.
focusin	입력 양식에 초점이 맞춰지기 바로 전에 발생합니다.
focusout	입력 양식에 초점이 사라지기 바로 전에 발생합니다.
blur	입력 양식에 초점이 사라지면 발생합니다.
select	입력 양식을 선택할 때 발생합니다(input[type=text]태그와 textarea 태그 제외).
submit	submit 버튼을 누르면 발생합니다.
reset	reset 버튼을 누르면 발생합니다.

# 입력 양식 이벤트

- submit 이벤트
  - 입력 양식이 있어야 하므로 body 태그 구성

코드 16-43 body 태그 구성

```
<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>
```

# 입력 양식 이벤트

## ■ submit 이벤트

- form 태그에서 발생하는 이벤트
  - form 객체에 submit() 메서드 연결
  - 유효성 검사를 할 때는 기본 이벤트 제거해야 함

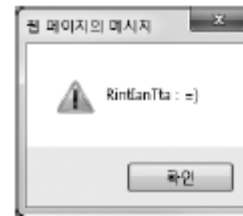
코드 16-44 submit 이벤트와 기본 이벤트 제거

```
<script>
$(document).ready(function () {
    $('#my_form').submit(function (event) {
        // 입력 양식의 value를 가져옵니다.
        var name = $('#name').val();
        var password = $('#password').val();

        // 출력합니다.
        alert(name + ' : ' + password);

        // 기본 이벤트를 제거합니다.
        event.preventDefault();
    });
});
</script>
```

그림 16-20 실행 결과



# 입력 양식 이벤트

- change 이벤트
  - type 속성이 checkbox와 radio인 input 태그의 상태 변경하는 이벤트
  - click 이벤트가 아님

코드 16-45 check 속성 변경

```
<script>
  $(document).ready(function () {
    $('#all_check').change(function () {
      if (this.checked) {
        $('#check_item').children().attr('checked', true);
      } else {
        $('#check_item').children().attr('checked', false);
      }
    });
  });
</script>
<body>
  <input type="checkbox" id="all_check"/>
  <label>All</label>

  <div id="check_item">
    <input type="checkbox"/>
    <label>A Option</label>
    <input type="checkbox"/>
    <label>B Option</label>
    <input type="checkbox"/>
    <label>C Option</label>
  </div>
</body>
```