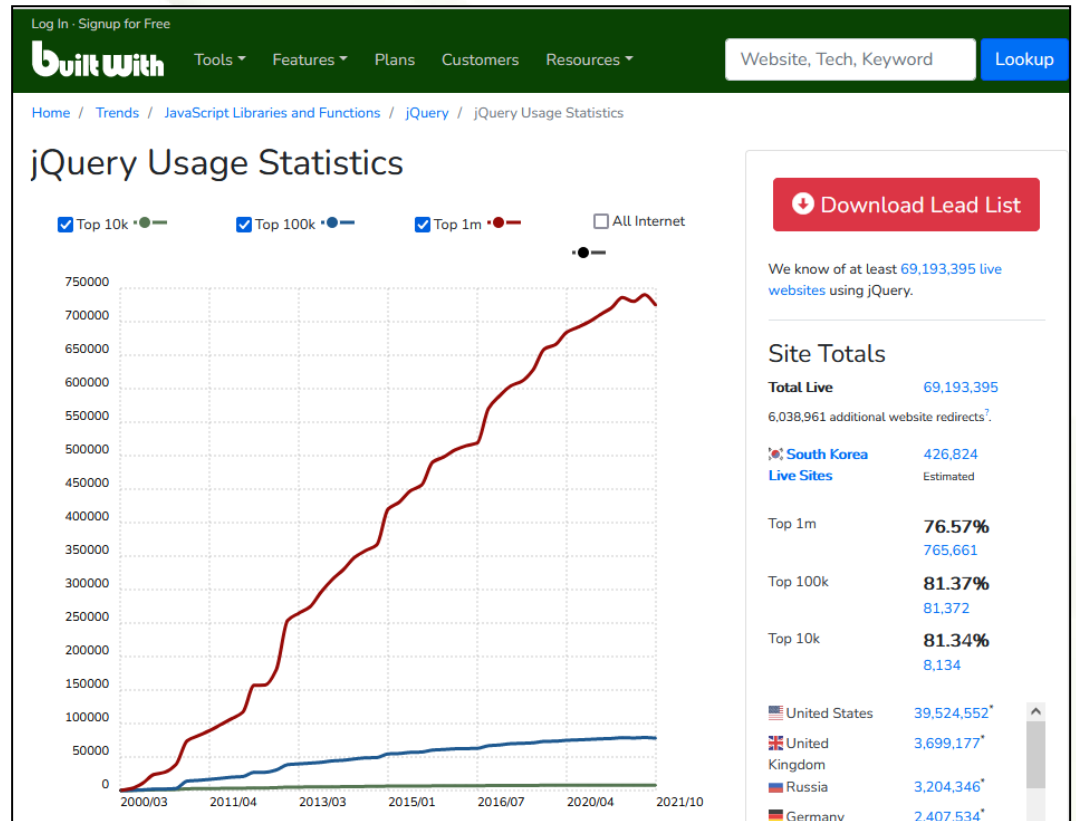


The background features a series of overlapping, wavy, ribbon-like shapes in various shades of green and white, creating a dynamic, flowing effect. The colors range from a deep forest green at the bottom to a bright lime green and finally to a pale, almost white green at the top.

# Introduction to jQuery

# jQuery 개요

- John Resig이 2006년 1월에 발표
- 모든 브라우저에서 동작하는 오픈소스 경량 클라이언트 자바스크립트 라이브러리
- 2021년 기준 상위 100만 개의 웹 페이지 중 76%가 jQuery 사용
- jQuery의 제작 목표
  - » DOM과 관련된 처리 쉽게 구현
  - » 일관된 이벤트 연결 쉽게 구현
  - » 시각적 효과 쉽게 구현
  - » Ajax 애플리케이션 쉽게 개발




# jQuery 특징


- 크로스 플랫폼 지원
  - » 모든 브라우저에서 동일하게 동작
  - » 브라우저간 호환성 유지를 위해 대체코드 작성 불필요
- 네이티브 DOM API에 비해 직관적이고 편리한 API 제공
  - » CSS 스타일의 선택자를 사용할 수 있으며 높은 생산성 제공
- 이벤트 처리, Ajax, Animation 효과를 쉽게 사용 가능
- 높은 수준의 시각적 효과 구현 가능
- 다양한 플러그인이 존재하며 다른 라이브러리들과 충돌을 일으키지 않음

# jQuery 사용

- Download Link → <http://www.jquery.com>




write less, do more.




Your donations help fund the continued development and growth of jQuery.

**SUPPORT THE PROJECT**


[Download](#) [API Documentation](#) [Blog](#) [Plugins](#) [Browser Support](#)




**Lightweight Footprint**  
Only 30kB minified and gzipped. Can also be included as an AMD module



**CSS3 Compliant**  
Supports CSS3 selectors to find elements as well as in style property manipulation



**Cross-Browser**  
[Chrome](#), [Edge](#), [Firefox](#), [IE](#), [Safari](#), [Android](#), [iOS](#), and more



**Download jQuery**  
v3.6.0  
The 1.x and 2.x branches no longer receive patches.

[View Source on GitHub →](#)  
[How jQuery Works →](#)

## What is jQuery?

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

## Resources

- [jQuery Core API Documentation](#)
- [jQuery Learning Center](#)
- [jQuery Blog](#)
- [Contribute to jQuery](#)

# jQuery 사용

- jQuery 1.x과 jQuery 2.x 두가지 버전이 존재
  - » 두 가지 버전 모두 동일한 API을 제공
  - » jQuery 2.x는 IE 8 이하를 지원하지 않음
- 2016년 6월 9일 jQuery 3.0 릴리스
  - » jQuery 1.x, jQuery 2.x 모두 jQuery 3.0으로 통합
  - » 오래된 버전의 브라우저를 지원하려면 jQuery 1.12 버전 사용
- CDN 링크 사용 또는 다운로드 후 웹사이트에 포함해서 지원

# CDN 배포

- CDN → Content Delivery Network
- 공개된 저장소를 사용해서 간편하게 콘텐츠를 제공하는 방식
  - » Google, MS, jQuery 등에서 jQuery 다운로드 링크 제공
  - » HTML 페이지 구성

```
<!DOCTYPE html>
<html>
  <head>
    <title>jQuery</title>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  </head>
  <body>

    <h1 id="main-heading">What is jQuery?</h1>

    <p>jQuery is the most popular JavaScript library.</p>

  </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 재구성한 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 특별한 기능을 제공하는 jQuery 메서드
  - » 주로 요소 집합을 제어하는 기능 제공
- 함수
  - » jQuery에 포함된 전역 함수
  - » 프로그램에 필요한 일반적인 기능을 제공



# jQuery & \$

- jQuery 사용을 위해 jQuery 객체 생성 필요
- jQuery 키워드는 jQuery 라이브러리의 최상위 객체 및 네임스페이스 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 & \$ 함수

- jQuery 객체 생성

```
jQuery() or $()
```

- 전달인자가 css 선택자인 경우

- » 선택자에의 의해 선택된 요소들을 포함하는 jQuery 객체 반환

```
jQuery('CSS 선택자') or $('CSS 선택자')
```

- 전달인자가 HTML 문자열인 경우

- » 새로 생성된 HTML 요소를 포함하는 jQuery 객체 반환

```
jQuery('HTML') or $('HTML')
```

```
jQuery("<p id='test'>My new text</p>") or $("<p id='test'>My new text</p>")
```

# jQuery & \$ 함수

- 전달인자가 javascript 객체인 경우
  - » javascript 객체를 포함하는 jQuery 객체 반환

```
jQuery(js_object) or $(js_object)
```

```
$('#div.foo').click(function () {  
    $(this).slideUp();  
});
```

# jQuery(document).ready( )

- \$(document).ready()
  - » DOM이 완전히 로드되기 전까지 대기하다가 로드가 완료되면 매개변수로 전달된 콜백함수 실행
  - » jQuery 이벤트 메서드 중 하나로 프로그램의 entry point 역할

```
jQuery(document).ready(function () {  
    // Do something...  
});  
  
// Shorthand for jQuery(document).ready()  
$(document).ready(function () {  
    // Do something...  
});  
  
// Shorthand for $(document).ready()  
$(function () {  
    // Do something...  
});
```

# jQuery 프레임워크 충돌 방지

- jQuery와 다른 자바스크립트 라이브러리와 키워드 충돌 방지 방법
- \$.noConflict() 함수 사용
  - » 충돌을 방지할 때 사용하는 메서드
  - » 호출 후에는 더 이상 jQuery의 식별자 \$를 사용할 수 없음

```
<!DOCTYPE html>
<html>
<head>
  <script src="http://code.jquery.com/jquery-3.6.0.js"></script>
  <script>
    $.noConflict();
    jQuery(document).ready(function () {
      // do something
    });
  </script>
</head>
<body>

</body>
</html>
```

# 선택자

- jQuery는 CSS 스타일의 Selector를 이용하여 요소를 선택할 수 있다
- 기본 선택자

셀렉터	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

```
<!DOCTYPE html>
<html>
<head>
  <script src="http://code.jquery.com/jquery-3.6.0.js"></script>
  <script>
    $(document).ready(function () {
      $('*').css('color', 'red');
    });
  </script>
</head>
<body>
  <h1>Lorem ipsum</h1>
</body>
</html>
```

# jQuery 기본 선택자

## ■ 태그 선택자

- » 태그 선택자는 특정한 태그만 선택하는 선택자
- » 태그 이름 사용

```
<!DOCTYPE html>
<html>
<head>
  <script src="http://code.jquery.com/jquery-3.6.0.js"></script>
  <script>
    $(document).ready(function () {
      $('h1').css('color', 'orange');
    });
  </script>
</head>
<body>
  <h1>Lorem ipsum</h1>
  <p>Lorem ipsum dolor sit amet.</p>
  <h1>Lorem ipsum</h1>
  <p>consectetur adipiscing elit.</p>
</body>
</html>
```



# jQuery 기본 선택자

## ■ 아이디 선택자

- » 특정한 id 속성을 가지고 있는 문서 객체 # 기호를 사용해서 선택
- » id 속성은 HTML 페이지 내에서 단 하나의 태그에만 적용
- » 예제 두 번째에 위치한 h1 태그가 id 속성으로 target 가짐

```
<!DOCTYPE html>
<html>
<head>
  <script src="http://code.jquery.com/jquery-3.6.0.js"></script>
  <script>
    $(document).ready(function () {
      $('#target').css('color', 'orange');
    });
  </script>
</head>
<body>
  <h1>Header-0</h1>
  <h1 id="target">Header-1</h1>
  <h1>Header-2</h1>
</body>
</html>
```

# jQuery 기본 선택자

## ■ 클래스 선택자

- » 특정한 class 속성 가진 문서 객체를 .기호를 사용해서 선택하는 선택자
- » 여러 개의 문서 객체 선택 가능
- » 두 클래스 속성을 모두 지정하려면 두 클래스 선택자 붙여서 사용

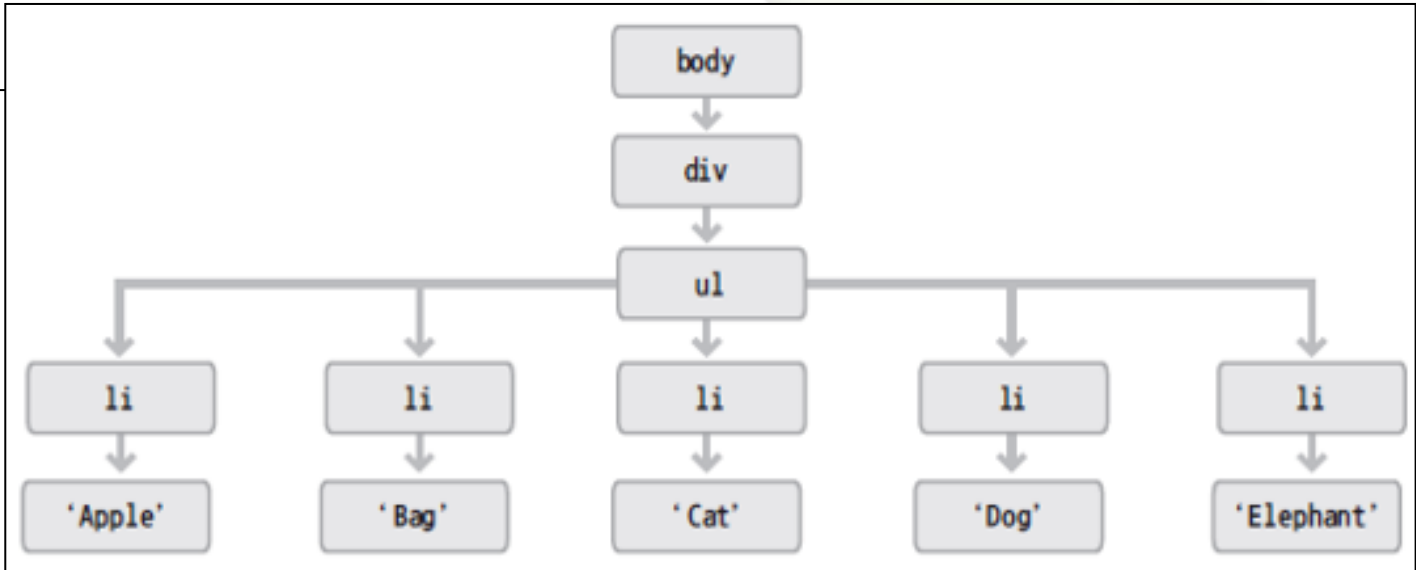
```
<!DOCTYPE html>
<html>
<head>
  <script src="http://code.jquery.com/jquery-3.6.0.js"></script>
  <script>
    $(document).ready(function () {
      $('.item').css('color', 'orange');
      $('h1.item').css('background', 'red');
      $('.item.select').css('color', 'blue');
    });
  </script>
</head>
<body>
  <h1 class="item">Header-0</h1>
  <h1 class="item select">Header-1</h1>
  <h1 class="item">Header-2</h1>
</body>
</html>
```

# jQuery 자식 선택자와 후손 선택자

## ■ jQuery 자식 선택자와 후손 선택자

» 기본 선택자의 앞에 붙여 사용하며 기본 선택자의 범위 제한

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



# jQuery 자식 선택자와 후손 선택자

## ■ 자식 선택자

- » 한 단계 아래의 자식 요소를 선택하는 선택자
- » '부모 > 자식의 형태로 사용

```
<!DOCTYPE html>
<html>
<head>
  <script src="http://code.jquery.com/jquery-3.6.0.js"></script>
  <script>
    $(document).ready(function () {
      $('body > *').css('color', 'red');
    });
  </script>
</head>
<body>
  <div>
    <ul>
      <li>Apple</li>
      <li>Bag</li>
      <li>Cat</li>
    </ul>
  </div>
</body>
</html>
```

# jQuery 자식 선택자와 후손 선택자

## ■ 후손 선택자

- » 계층의 깊이와 관계 없이 후손을 선택하는 선택자
- » '요소A 요소B'의 형태로 사용 → 요소 A의 후손으로 범위 한정

```
<!DOCTYPE html>
<html>
<head>
  <script src="http://code.jquery.com/jquery-3.6.0.js"></script>
  <script>
    $(document).ready(function () {
      $('body *').css('color', 'red');
    });
  </script>
</head>
<body>
  <div>
    <ul>
      <li>Apple</li>
      <li>Bag</li>
      <li>Cat</li>
    </ul>
  </div>
</body>
</html>
```

# jQuery 속성 선택자

- 속성의 값을 사용해서 문서 객체를 선택하는 선택자
- 종류

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 사례

```
<!DOCTYPE html>
<html>
<head>
  <script src="http://code.jquery.com/jquery-3.6.0.js"></script>
  <script>
    $(document).ready(function () {
      $('input[type="text"]').val('Hello jQuery..!');
    });
  </script>
</head>
<body>
  <input type="text" />
  <input type="password" />
  <input type="radio" />
  <input type="checkbox" />
  <input type="file" />
</body>
</html>
```

# jQuery 입력 양식 필터 선택자

- input 태그의 type 속성 또는 상태에 따라 문서 객체 선택
- 종류

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.

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 사례

```
<!DOCTYPE html>
<html>
<head>
  <script src="http://code.jquery.com/jquery-3.6.0.js"></script>
  <script>
    $(document).ready(function () {
      setTimeout(function () {
        var value = $('select > option:selected').val();
        alert(value);
      }, 5000);
    });
  </script>
</head>
<body>
  <select>
    <option>Apple</option>
    <option>Bag</option>
    <option>Cat</option>
    <option>Dog</option>
    <option>Elephant</option>
  </select>
</body>
</html>
```

# jQuery 필터 선택자

## ■ 위치 지정 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

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

## ■ 필터 선택자 사용 사례

```
<!DOCTYPE html>
<html>
<head>
  <script src="http://code.jquery.com/jquery-3.6.0.js"></script>
  <script>
    $(document).ready(function () {
      $('tr:odd').css('background', '#F9F9F9');
      $('tr:even').css('background', '#9F9F9F');
      $('tr:first').css('background', '#000000').css('color', '#FFFFFF');
    });
  </script>
</head>
<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 노량진구</td></tr>
  </table>
</body>
</html>
```

# jQuery 필터 선택자

## ■ 필터 선택자 사용 사례

```
<html>
<head>
  <script src="http://code.jquery.com/jquery-3.6.0.js"></script>
  <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>
</head>
<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 노량진구</td></tr>
  </table>
</body>
</html>
```

# 기본 필터링 메서드

- 객체 집합에서 조건에 부합하는 객체를 선택 및 추출

```
<!DOCTYPE html>
<html>
<head>
  <script src="http://code.jquery.com/jquery-3.6.0.js"></script>
  <script>
    $(document).ready(function () {
      $('h3').filter(':even').css({
        backgroundColor: 'black',
        color: 'white'
      });
    });
  </script>
</head>
<body>
  <h3>Header-0</h3>
  <h3>Header-1</h3>
  <h3>Header-2</h3>
  <h3>Header-3</h3>
  <h3>Header-4</h3>
  <h3>Header-5</h3>
</body>
</html>
```

필터 조건으로 선택자 사용

# 기본 필터링 메서드

- 객체 집합에서 조건에 부합하는 객체를 선택 및 추출

```
<html>
<head>
  <script src="http://code.jquery.com/jquery-3.6.0.js"></script>
  <script>
    $(document).ready(function () {
      $('h3').filter(function (index) {
        return index % 3 == 0;
      }).css({
        backgroundColor: 'black',
        color: 'white'
      });
    });
  </script>
</head>
<body>
  <h3>Header-0</h3>
  <h3>Header-1</h3>
  <h3>Header-2</h3>
  <h3>Header-3</h3>
  <h3>Header-4</h3>
  <h3>Header-5</h3>
</body>
</html>
```

필터 조건으로 함수 사용

# 기본 필터링 메서드

- 함수 체이닝을 사용해 한 문장으로 서로 다른 문서 객체 제어 가능
- 필터링 이전 객체 집합 (필터 적용 제거)으로 이동하기 위해 `end()` 함수 사용

```
<!DOCTYPE html>
<html>
<head>
  <script src="http://code.jquery.com/jquery-3.6.0.js"></script>
  <script>
    $(document).ready(function () {
      $('h1').css('background', 'orange')
        .filter(':even').css('color', 'white')
        .end()
        .filter(':odd').css('color', 'red');
    });
  </script>
</head>
<body>
  <h1>Header-0</h1>
  <h1>Header-1</h1>
  <h1>Header-2</h1>
</body>
</html>
```

## 특정 위치의 문서 객체 선택

- eq() 메서드 사용
- 매개 변수에 숫자 입력
  - » 양수를 입력하면 앞쪽 기준 선택, 음수를 입력하면 뒤쪽 기준 선택
  - » 첫 번째 객체는 0번째, 맨 뒤의 문서 객체가 -1번째

```
<html>
<head>
  <script src="http://code.jquery.com/jquery-3.6.0.js"></script>
  <script>
    $(document).ready(function () {
      $('h1').eq(0).css('background', 'orange');
      $('h1').eq(-1).css('background', 'red');
    });
  </script>
</head>
<body>
  <div>
    <h1>Header-0</h1>
    <h1>Header-1</h1>
    <h1>Header-2</h1>
  </div>
</body>
</html>
```



# 문서 객체 추가 선택

- add() 메서드 사용

```
<!DOCTYPE html>
<html>
<head>
  <script src="http://code.jquery.com/jquery-3.6.0.js"></script>
  <script>
    $(document).ready(function () {
      $('h1').css('background', 'Gray')
      .add('h2').css('float', 'left');
    });
  </script>
</head>
<body>
  <h1>Header-0</h1>
  <h2>Header-1</h2>
  <h1>Header-2</h1>
  <h2>Header-3</h2>
  <h1>Header-4</h1>
</body>
</html>
```

## 문서 객체의 특징 판별

- 문서 객체가 어떤 특징을 가지고 있는지 판단하기 위해 `is()` 함수 사용

```
<!DOCTYPE html>
<html>
<head>
  <script src="http://code.jquery.com/jquery-3.6.0.js"></script>
  <script>
    $(document).ready(function () {
      $('h1').each(function () {
        if ($(this).is('.select')) {
          $(this).css('background', 'orange');
        }
      });
    });
  </script>
</head>
<body>
  <h1 class="select">Header-0</h1>
  <h1>Header-1</h1>
  <h1 class="select">Header-2</h1>
</body>
</html>
```

## 부모 객체 선택

- 부모 객체를 선택하기 위해 parent() 함수 사용

```
<!DOCTYPE html>
<html>
<head>
  <script src="http://code.jquery.com/jquery-3.6.0.js"></script>
</head>
<body>
  <script>
    $(document).ready(function () {
      $('span').parent().css('background', 'red');
    });
  </script>
  <div>
    <span>비활성화</span>
    <button>활성화하기</button>
  </div>
</body>
</html>
```

## 특정 태그 선택

- 확장 객체 집합에서 특정 태그를 선택하기 위해 find() 함수 사용

```
<head>
  <script src="http://code.jquery.com/jquery-3.6.0.js"></script>
</head>
<body>
  <script>
    $(document).ready(function () {
      $('button').click(function () {
        $(this).text('비활성화하기');
        $(this).parent().css('background', 'red');
        $(this).parent().find('span').text('활성화');
      })
    });
  </script>
  <div>
    <span>비활성화</span>
    <button>활성화하기</button>
  </div>
  <div>
    <span>비활성화</span>
    <button>활성화하기</button>
  </div>
</body>
```

## 문서 객체 제어 함수

함수	설명
<code>addClass()</code>	문서 객체에 클래스 속성 추가 매개 변수에 추가하려는 클래스 속성 문자열 또는 함수 사용
<code>removeClass()</code>	문서 객체의 클래스 속성을 제거
<code>attr()</code>	문서 객체 속성 값 읽기 또는 쓰기
<code>removeAttr()</code>	문서 객체 속성 제거
<code>css()</code>	문서 객체 스타일 속성 읽기 또는 쓰기
<code>html()</code>	문서 객체의 <code>html</code> 문자열을 읽거나 새로운 <code>html</code> 추가
<code>text()</code>	문서 객체의 <code>text</code> 문자열을 읽거나 새로운 <code>text</code> 추가
<code>remove()</code>	문서 객체 제거
<code>empty()</code>	문서 객체의 내용 제거

## 문서 객체 제어 함수

함수	설명
appendTo()	문서 객체를 다른 문서 객체의 마지막 하위 요소로 추가
prependTo()	문서 객체를 다른 문서 객체의 첫 번째 하위 요소로 추가
append()	문서 객체에 다른 문서 객체를 마지막 하위 요소로 추가
prepend()	문서 객체에 다른 문서 객체를 첫 번째 하위 요소로 추가
insertBefore()	문서 객체를 다른 문서 객체와 같은 수준에서 앞에 추가
insertAfter()	문서 객체를 다른 문서 객체와 같은 수준에서 뒤에 추가
before()	문서 객체 앞에 같은 수준에서 다른 문서 객체 추가
after()	문서 객체 뒤에 같은 수준에서 다른 문서 객체 추가
clone()	문서 객체를 복제

# 이벤트 처리

## ■ 이벤트 연결 및 해제 함수

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

```
<style>
    .reverse {
        background: black;
        color: white;
    }
</style>
<script src="http://code.jquery.com/jquery-3.6.0.js"></script>
<script>
    $(document).ready(function () {
        $('h1').on('click', function () {
            $(this).html(function (index, html) {
                return html + '+';
            });
        });
        $('h1').on({
            mouseenter: function () { $(this).addClass('reverse'); },
            mouseleave: function () { $(this).removeClass('reverse'); }
        });
    });
</script>
```

# 이벤트 처리

## ■ 이벤트 연결 및 해제 함수

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

```
<head>
  <script src="http://code.jquery.com/jquery-3.6.0.js"></script>
  <script>
    $(document).ready(function () {
      $('#wrap').on('click', 'h1', function () {
        var length = $('h1').length;
        var targetHTML = $(this).html();
        $('#wrap').append('<h1>' + length + ' - ' + targetHTML + '</h1>');
        $('#wrap').off('click', 'h1');
      });
    });
  </script>
</head>
<body>
  <div id="wrap">
    <h1>Header</h1>
  </div>
</body>
```



# 이벤트 처리

## ■ 개별 이벤트 단위 함수 제공

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

메서드 이름

설명

toggle()

click 이벤트를 여러 이벤트 핸들러를 번갈아가며 실행할 수 있게 연결합니다.

hover()

mouseenter 이벤트와 mouseleave 이벤트를 동시에 연결합니다.

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

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

# 이벤트 객체

- jQuery에서 모든 브라우저에서 동일하게 동작하는 이벤트 객체 제공
- 자주 사용하는 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.

```
$(canvas).on({  
  mousedown: function (event) {  
    var position = $(this).offset();  
    var x = event.pageX - position.left;  
    var y = event.pageY - position.top;  
  },  
  mouseup: function (event) {  
    var position = $(this).offset();  
    var x = event.pageX - position.left;  
    var y = event.pageY - position.top;  
  }  
});
```

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

# 이벤트 강제 발생

- trigger() 함수 사용

- » \$(selector).trigger(eventName)

- » \$(selector).trigger(eventName, data)

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

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