



자바스크립트 프로그래밍 입문 (2판)

Contents

학습목표

- jQuery를 HTML 페이지에서 사용할 수 있는 방법을 이해합니다.
- jQuery로 문서 객체를 선택하는 방법을 이해합니다.
- jQuery로 문서 객체를 조작하는 방법을 이해합니다.
- jQuery로 문서 객체에 이벤트를 연결하는 방법을 이해합니다.

내용

- jQuery 사용 준비
- jQuery 객체
- 문서 객체 선택
- 문서 객체 개별 조작
- 문서 객체 조작
- 문서 객체 생성
- 이벤트
- 애니메이션

1. jQuery 사용 준비

- 공식 웹 사이트 주소는 <http://jquery.com/>

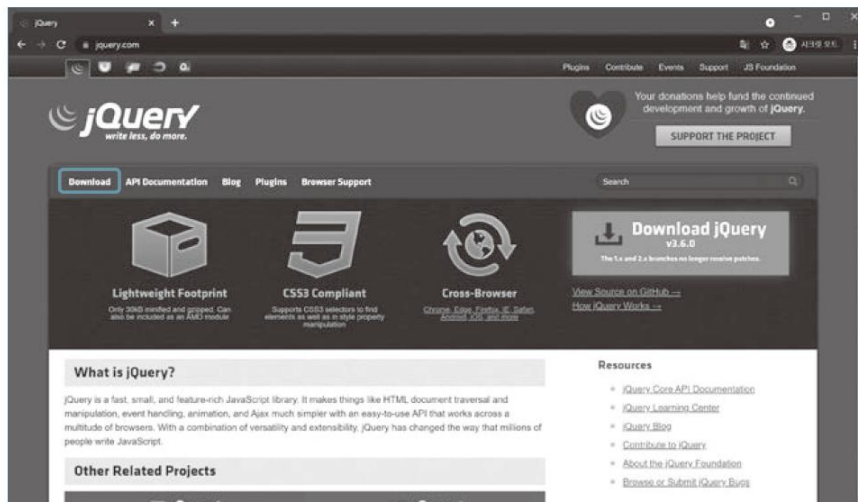


그림 14-1 jQuery(<http://jquery.com/>)

1. jQuery 사용 준비

- jQuery 사용 방법 1
- [예제 14-1] 파일 다운로드해 사용



그림 14-3 폴더 구성

코드 14-1 다운로드받은 파일을 활용한 jQuery 사용 준비

```
<!DOCTYPE html>
<html>
<head>
  <title>jQuery Basic</title>
  <script src="jquery.min.js"></script>
  <script>
    // 여기에 코드를 입력합니다.
  </script>
</head>
<body>

</body>
</html>
```

1. jQuery 사용 준비

■ jQuery 사용 방법 2

- CDN을 활용 (CDN은 사용자에게 간편하게 콘텐츠를 제공하는 방식)

Using jQuery with a CDN

CDNs can offer a performance benefit by hosting jQuery on servers spread across the globe. This also offers an advantage that if the visitor to your webpage has already downloaded a copy of jQuery from the same CDN, it won't have to be re-downloaded.

jQuery's CDN provided by StackPath

The jQuery CDN supports Subresource Integrity (SRI) which allows the browser to verify that the files being delivered have not been modified. This specification is currently being implemented by browsers. Adding the new integrity attribute will ensure your application gains this security improvement as browsers support it.

To use the jQuery CDN, just reference the file in the script tag directly from the jQuery CDN domain. You can get the complete script tag, including Subresource Integrity attribute, by visiting <https://code.jquery.com> and clicking on the version of the file that you want to use. Copy and paste that tag into your HTML file.

Starting with jQuery 1.9, sourcemap files are available on the jQuery CDN. However, as of version 1.10.0/2.1.0 the compressed jQuery no longer includes the sourcemap comment in CDN copies because it requires the uncompressed file and sourcemap file to be placed at the same location as the compressed file. If you are maintaining local copies and can control the locations all three files, you can add the sourcemap comment to the compressed file for easier debugging.

To see all available files and versions, visit <https://code.jquery.com>

Other CDNs

The following CDNs also host compressed and uncompressed versions of jQuery releases. Starting with jQuery 1.9 they may also host sourcemap files; check the site's documentation.

Note that there may be delays between a jQuery release and its availability there. Please be patient, they receive the files at the same time the blog post is made public. Beta and release candidates are not hosted by these CDNs.

- [Google CDN](#)
- [Microsoft CDN](#)
- [CDNJS CDN](#)
- [jsDelivr CDN](#)

그림 14-4 CDN 경로

1. jQuery 사용 준비

- 각 페이지에 접속하면 jQuery 파일과 관련된 경로가 나옴
- 예) 구글 CDN 클릭

jQuery

3.x snippet:

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
```

2.x snippet:

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.4/jquery.min.js"></script>
```

1.x snippet:

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
```

그림 14-5 구글 CDN

- 이 주소를 사용해 script 태그를 만들어 넣음

1. jQuery 사용 준비

- [예제 14-2] CDN을 활용한 jQuery 사용 준비
 - CDN을 활용해 HTML 페이지 내부로 jQuery를 불러옴. 구글 CDN을 활용

코드 14-2 CDN을 활용한 jQuery 사용 준비

```
<!DOCTYPE html>
<html>
<head>
  <title>jQuery Basic</title>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
</head>
<body>

</body>
</html>
```

2. jQuery 객체

- jQuery 라이브러리는 \$ 함수 활용
- \$ 함수의 매개 변수에는 문서 객체, CSS 형식, HTML 형식의 문자열 삽입

```
window.jQuery = window.$ = jQuery;
```

`$(매개_변수).메소드(매개_변수, 매개_변수)`

그림 14-7 jQuery 기본 사용 형태

- jQuery 객체생성

```
// 일반 문서 객체로 jQuery 객체를 생성합니다.  
$(document)
```

```
// CSS 선택자로 jQuery 객체를 생성합니다.  
$('h1')
```

```
// HTML 문자열로 jQuery 객체를 생성합니다.  
$('<h1></h1>')
```


2. jQuery 객체

- [예제 14-3] \$(document).ready()
 - 문서 객체의 생성 완료 시점을 잡는 이벤트 연결

코드 14-3 jQuery 기본 코드

```
<!DOCTYPE html>
<html>
<head>
  <title>jQuery Basic</title>
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <script>
    $(document).ready(function () {

    });
  </script>
</head>
<body>

</body>
</html>
```

3. 문서 객체 선택

- CSS 선택자를 사용해 문서 객체를 선택

```
// h1 태그를 선택합니다.  
$('h1')  
  
// h1 태그 중에 class 속성으로 logo를 가진 태그를 선택합니다.  
$('h1.logo')  
  
// id 속성이 head인 태그를 선택합니다.  
$('#head')  
  
// input 태그 중에 type 속성이 submit인 태그를 선택합니다.  
$('input[type=submit']')
```

표 14-1 객체 탐색 메소드

메소드	설명
parent()	부모 태그를 선택합니다.
find()	후손 태그를 찾습니다.

3. 문서 객체 선택

- 예) h1 태그에 parent () 메소드와 find () 메소드를 사용

```
// h1 태그의 부모 태그를 선택합니다.
```

```
$('#h1').parent()
```

```
// h1 태그 내부에 있는 i 태그를 선택합니다.
```

```
$('#h1').find('i')
```

4. 문서 객체 개별 조작

표 14-2 선택된 문서 객체의 수

속성	설명
length	선택된 문서 객체의 수를 구합니다.

표 14-3 선택된 문서 객체 추출

메소드	설명
get()	선택한 문서 객체 중 하나를 선택합니다.

- [예제 14-4] 문서 객체 반복(1)
 - h1 태그를 선택하고, 내부 요소에 반복을 걸어
 짝수 번째 위치하는 것에만 배경 색상을 적용함

코드 14-4 문서 객체 반복 (1)

jQueryLoopA.html

```
<!DOCTYPE html>
<html>
<head>
  <title>jQuery Event</title>
  <script src="jquery-3.6.0.min.js"></script>
</script>
  // 웹 페이지를 모두 불러오면
  $(document).ready(function () {
    // h1 태그를 모두 추출하고
    const $headers = $('h1');
```

```
    // h1 태그의 개수만큼 반복을 돌려서
    for (let i = 0; i < $headers.length; i++) {
      // 특정 위치에서
      if (i % 2 == 1) {
        // i번째 요소를 추출해
        const domElement = $headers.get(i);
        // 배경 색상을 적용합니다.
        $(domElement).css('backgroundColor', 'red');
      }
    }
  });
</script>
</head>
```

4. 문서 객체 개별 조작

- h1 태그를 선택하고, 내부 요소에 반복을 걸어
짝수 번째 위치하는 것에만 배경 색상을 적용함

```
<body>  
  <h1>HeaderA</h1>  
  <h1>HeaderB</h1>  
  <h1>HeaderC</h1>  
  <h1>HeaderD</h1>  
</body>  
</html>
```

HeaderA

HeaderB

HeaderC

HeaderD

4. 문서 객체 개별 조작

■ each() 메소드

표 14-4 선택된 문서 객체 반복 적용

메소드	설명
each()	선택한 문서 객체에 반복을 적용합니다.

- **주의점** : Array 객체의 forEach ()메소드와 인덱스, 요소 순서가 다름

표 14-5 each() 메소드의 콜백 함수

Array 객체의 forEach() 메소드	jQuery의 each() 메소드
<pre>[].forEach(function (item, index) { });</pre>	<pre>\$('#h1').each(function (index, item) { });</pre>

4. 문서 객체 개별 조작

■ [예제 14-5] 문서 객체 반복(2)

코드 14-5

문서 객체 반복 (2)

jQueryLoopB.html

```
<!DOCTYPE html>
<html>
<head>
  <title>jQuery Event</title>
  <script src="jquery-3.6.0.min.js"></script>
  <script>
    // 웹 페이지를 모두 불러오면
    $(document).ready(function () {
      // h1 태그를 모두 추출하고 반복을 돌려
      $('h1').each(function (index, element) {
        // 특정 위치에서
        if (index % 2 == 1) {
          // 배경 색상을 적용합니다.
          $(this).css('backgroundColor', 'red');
        }
      });
    });
  </script>
</head>
<body>
  <h1>HeaderA</h1>
  <h1>HeaderB</h1>
  <h1>HeaderC</h1>
  <h1>HeaderD</h1>
</body>
</html>
```

5. 문서 객체 조작

■ 글자 조작

표 14-6 글자 조작 메소드

메소드	설명
text()	html 태그 내부의 문자를 조작합니다.
html()	html 태그 내부의 문자를 조작합니다(HTML 태그 인식).

```
// h1 태그 내부의 문자를 가져옵니다.
```

```
$('#h1').text()
```

```
// h1 태그 내부의 문자를 가져옵니다.
```

```
$('#h1').html()
```

- 선택자로 여러 개의 문서 객체를 선택할 때 text () 메소드는 모든 문서 객체 내부의 문자를 출력, html () 메소드는 첫 번째 문서 객체 내부의 문자를 출력

5. 문서 객체 조작

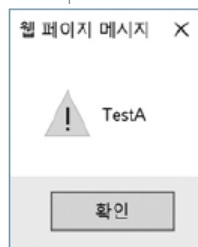
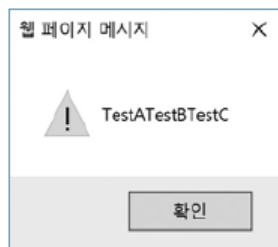
- [예제 14-6] text() 메소드와 html() 메소드의 GET 형태

코드 14-7

text() 메소드와 html() 메소드의 GET 형태

GetTextAndHtml.html

```
<!DOCTYPE html>
<html>
<head>
  <title>jQuery Text</title>
  <script src="jquery-3.6.0.min.js"></script>
  <script>
    $(document).ready(function () {
      // text() 메소드와 html() 메소드의 리턴 값을 출력합니다.
      alert($('p').text());
      alert($('p').html());
    });
  </script>
</head>
<body>
  <p>TestA</p>
  <p>TestB</p>
  <p>TestC</p>
</body>
</html>
```



5. 문서 객체 조작

■ 문서 객체 내부의 문자를 변경

```
// h1 태그 내부의 문자를 설정합니다.  
$('#h1').text('안녕하세요')  
  
// h1 태그 내부의 문자를 설정합니다.  
$('#h1').html('안녕하세요')
```

- [예제 14-7] text() 메소드와 html() 메소드의 SET 형태
 - h1 태그에 text () 메소드와 html() 메소드로 내부 문자를 변경

코드 14-8 text() 메소드와 html() 메소드의 SET 형태

SetTextAndHtml.html

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title>jQuery Text</title>  
    <script src="jquery-3.6.0.min.js"></script>  
    <script>  
      $(document).ready(function () {  
        $('.text-1').text('h1>text() 메소드</h1>');  
        $('.text-2').html('h1>html() 메소드</h1>');  
      });  
    </script>  
  </head>
```

5. 문서 객체 조작

```
<p class="text-2">TestB</p>  
<p class="text-2">TestB</p>  
</body>  
</html>
```

```
<h1>text() 메소드 </h1>  
<h1>text() 메소드 </h1>
```

text() 메소드

text() 메소드

- SET 형태는 두 메소드 모두 선택한 문서 객체에 적용함

5. 문서 객체 조작

■ 스타일 조작

■ CSS()

표 14-7 스타일 조작 메소드

메소드	설명
css()	스타일을 조작합니다.

```
// h1 태그의 color 스타일 속성을 가져옵니다.
```

```
$('#h1').css('color')
```

```
// h1 태그의 color 스타일 속성을 red로 설정합니다.
```

```
$('#h1').css('color', 'red')
```

```
// h1 태그의 color 스타일 속성과 backgroundColor 스타일 속성을 한꺼번에 설정합니다.
```

```
$('#h1').css({  
  color: 'red',  
  backgroundColor: 'orange'  
})
```

5. 문서 객체 조작

- [예제 14-8] 스타일 조작 - [예제 13-6]을 jQuery로 변경

코드 14-9 스타일 조작

jQueryStyle.html

```
<!DOCTYPE html>
<html>
<head>
  <title>jQuery Event</title>
  <script src="jquery-3.6.0.min.js"></script>
  <script>
    // 웹 페이지를 모두 불러오면
    $(document).ready(function () {
      // 0부터 255까지 반복문을 돌리고
      for (let i = 0; i < 256; i++) {
        // div 태그에 스타일을 적용해 body 태그에 추가합니다.
        $('<div></div>').css({
          height: 2,
          background: 'rgb(' + i + ', ' + i + ', ' + i + ')'
        }).appendTo('body');
      }
    });
  </script>
</head>
<body>

</body>
</html>
```



5. 문서 객체 조작

■ 속성 조작

■ attr()

표 14-8 속성 조작 메소드

메소드	설명
attr()	속성을 조작합니다.

```
// img 태그의 src 속성을 가져옵니다.
```

```
$('img').attr('src')
```

```
// img 태그의 src 속성을 https://via.placeholder.com/100x100으로 설정합니다.
```

```
$('img').css('src', 'https://via.placeholder.com/100x100')
```

```
// img 태그의 src 속성과 alt 속성을 한꺼번에 지정합니다.
```

```
$('img').css({  
    src: 'https://via.placeholder.com/100x100',  
    alt: 'placeholder.it'  
})
```

5. 문서 객체 조작

- [예제 14-9] jQuery를 사용한 속성 조작
 - each() 메소드를 사용해 너비가 (index + 1) * 100인 이미지를 src 속성에 넣기

코드 14-10

jQuery를 사용한 속성 조작

jQueryAttr.html

```
<!DOCTYPE html>
<html>
<head>
  <title>jQuery Attribute</title>
  <script src="jquery-3.6.0.min.js"></script>
  <script>
    // 웹 페이지를 모두 불러오면
    $(document).ready(function () {
      // img 태그에
      $('img').each(function (index, element) {
        // src 속성을 조작합니다.
        $(this).attr('src',
          'https://via.placeholder.com/' + (index + 1) * 100 + 'x100');
      });
    });
  </script>
</head>
```

5. 문서 객체 조작

```
<body>  
  <img />  
  <img />  
  <img />  
</body>  
</html>
```

100×100

200×100

300×100

6. 문서 객체 생성

- \$() 함수의 매개 변수에 HTML 형식의 문자열을 입력

```
$(' <h1></h1>')
```

```
$(' <h1></h1>')  
  .text('안녕하세요')  
  .attr('data-test', 'test')  
  .css({  
    backgroundColor: 'red',  
    color: 'white'  
  });
```

표 14-9 문서 객체 추가 메소드

메소드	설명
\$(A).prependTo(B)	A를 B 왼쪽 앞에 추가합니다.
\$(A).appendTo(B)	A를 B 왼쪽 뒤에 추가합니다.
\$(A).insertBefore(B)	A를 B 앞에 추가합니다.
\$(A).insertAfter(B)	A를 B 뒤에 추가합니다.

```
<h1>insertBefore</h1>  
<div class="target">  
  <h1>prependTo</h1>  
  <h1>appendTo</h1>  
</div>  
<h1>insertAfter</h1>
```

그림 14-8 문서 객체 추가 메소드 적용

6. 문서 객체 생성

- [예제 14-10] jQuery를 사용한 문서 객체 생성
 - h1 태그를 생성하고 글자, 속성, 스타일을 설정해서 body 태그 내부에 추가

코드 14-11 jQuery를 사용한 문서 객체 생성

CreateElement.html

```
<!DOCTYPE html>
<html>
<head>
  <title>jQuery Basic</title>
  <script src="jquery-3.6.0.min.js"></script>
  <script>
    $(document).ready(function () {
      $('h1</h1>')
        .text('안녕하세요')
        .attr('data-test', 'test')
        .css({
          backgroundColor: 'red',
          color: 'white'
        })
        .appendTo('body');
    });
  </script>
</head>
<body>
```

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

안녕하세요

7. 이벤트

표 14-10 jQuery의 이벤트 메소드

메소드	설명
on()	이벤트를 연결합니다.
off()	이벤트를 제거합니다.

■ 이벤트 직접 연결

- 특정 태그에 이벤트를 연결하고, 특정 태그를 눌렀을 때 이벤트가 발생
- jQuery로 이벤트를 직접 연결할 때는 on() 메소드 사용
- 기본 형태

```
$(선택자).on(이벤트_이름, 콜백_함수)
```

7. 이벤트

- [예제 14-11] 이벤트 직접 연결 - h1 태그에 click 이벤트를 연결

코드 14-12 이벤트 직접 연결

DirectEvent.html

```
<!DOCTYPE html>
<html>
<head>
  <title>jQuery Event</title>
  <script src="jquery-3.6.0.min.js"></script>
  <script>
    // 웹 페이지를 모두 불러오면
    $(document).ready(function () {
      // h1 태그에 click 이벤트를 연결합니다.
      $('h1').on('click', function (event) {
        // 이벤트를 발생시킨 문서 객체의 문자를 추출합니다.
        const text = $(this).text();
        // 출력합니다.
        alert(text);
      });
    });
  </script>
</head>
<body>
  <h1>Header_1</h1>
  <h1>Header_2</h1>
</body>
</html>
```

7. 이벤트

표 14-11 키보드 이벤트

메소드	설명
keydown()	키보드 키를 눌렀을 때
keypress()	키가 입력되었을 때
keyup()	키보드 키를 떼었을 때

표 14-12 마우스 이벤트

메소드	설명
click()	마우스를 클릭했을 때
dblclick()	마우스를 더블 클릭했을 때
mousedown()	마우스 버튼을 눌렀을 때
mouseenter()	마우스 커서가 해당 태그로 들어갔을 때
mouseleave()	마우스 커서가 해당 태그에서 나갔을 때
mousemove()	마우스가 움직일 때
mouseup()	마우스 버튼을 땄 때

7. 이벤트

표 14-13 입력 양식 이벤트

메소드	설명
<code>blur()</code>	입력 양식에 값 입력을 종료할 때
<code>change()</code>	입력 양식의 값이 변경될 때
<code>focus()</code>	입력 양식에 값 입력을 시작할 때
<code>select()</code>	<code>type</code> 속성이 <code>select</code> 인 입력 양식의 목록에서 값을 선택했을 때
<code>submit()</code>	<code>type</code> 속성이 <code>submit</code> 인 입력 양식을 클릭했을 때

표 14-14 웹 브라우저 이벤트

메소드	설명
<code>resize()</code>	웹 브라우저의 크기를 변경할 때
<code>scroll()</code>	웹 브라우저를 스크롤할 때

7. 이벤트

코드 14-14 이벤트가 발생하면 새로운 객체 생성

```
<!DOCTYPE html>
<html>
<head>
  <title>jQuery Event</title>
  <script src="jquery-3.6.0.min.js"></script>
  <script>
    // 웹 페이지를 모두 불러오면
    $(document).ready(function () {
      // h1 태그에 click 이벤트를 연결합니다.
      $('h1').on('click', function (event) {
        // h1 태그를 생성해서 body 태그에 추가합니다.
        $('<h1></h1>').text($(this).text()).appendTo('body');
      });
    });
  </script>
</head>

  <h1>Header</h1>
</body>
</html>
```

Header
Header
Header

그림 14-9 이벤트 직접 연결은 새로 생성한 태그에 이벤트를 연결하지 않음

7. 이벤트

코드 14-15 이벤트 추가 연결

```
<script>
  // 웹 페이지를 모두 불러오면
  $(document).ready(function () {
    // 변수를 선언합니다.
    const handler = function (event) {
      // h1 태그를 생성해서 body 태그에 추가합니다.
      $('<h1></h1>')
        .text($(this).text())
        .on('click', handler)
        .appendTo('body');
    };

    // h1 태그에 click 이벤트를 연결합니다.
    $('h1').on('click', handler);
  });
</script>
```


7. 이벤트

이벤트 간접 연결

- 부모에게 이벤트를 위임해서 부모가 이벤트를 처리하게 하는 것
- [예제 14-12] 이벤트 간접 연결
 - body 태그 내부의 h1 태그에서 click 이벤트가 발생할 때 h1 태그를 생성해 추가

코드 14-16 이벤트 간접 연결

RemoveEvent.html

```
<!DOCTYPE html>
<html>
<head>
  <title>jQuery Event</title>
  <script src="jquery-3.6.0.min.js"></script>
  <script>
    // 웹 페이지를 모두 불러오면
    $(document).ready(function () {
      // body 태그에 h1 태그의 click 이벤트를 위임합니다.
```

```
      $('body').on('click', 'h1', function (event) {
        // h1 태그를 생성해서 body 태그에 추가합니다.
        $('<h1></h1>').text($(this).text()).appendTo('body');
      });
    });
  </script>
</head>
<body>
  <h1>Header</h1>
</body>
</html>
```

7. 이벤트

이벤트 제거

- off () 메소드 사용
- [예제 14-13] 이벤트 제거
 - h1 태그를 클릭하면 새로 h1 태그를 생성해 추가, 기존 태그에 서는 이벤트 제거

코드 14-17 이벤트 제거

RemoveEvent.js

```
<!DOCTYPE html>
<html>
<head>
  <title>jQuery Event</title>
  <script src="jquery-3.6.0.min.js"></script>
  <script>
    // 웹 페이지를 모두 불러오면
    $(document).ready(function () {
      const handler = function (event) {
        // h1 태그를 생성해서 body 태그에 추가합니다.
        $('<h1></h1>')
          .text($(this).text())
          .click(handler)
          .appendTo('body');
        // 현재 태그의 이벤트를 제거합니다.
        $(this).off();
      };
      // h1 태그를 클릭하면
      $('h1').on('click', handler);
    });
  </script>
</head>
<body>
  <h1>Header</h1>
</body>
</html>
```

7. 이벤트

```
// h1 태그의 모든 이벤트를 제거합니다.  
$('h1').off();  
  
// h1 태그의 click 이벤트를 모두 제거합니다.  
$('h1').off('click');  
  
// h1 태그의 click 이벤트로 연결한 콜백 함수 중 특정한 콜백 함수만 제거합니다.  
$('h1').off('click', handler);
```

표 14-15 이벤트를 한 번만 연결하는 메소드

메소드	설명
one()	이벤트를 한 번만 연결합니다.

8. 애니메이션

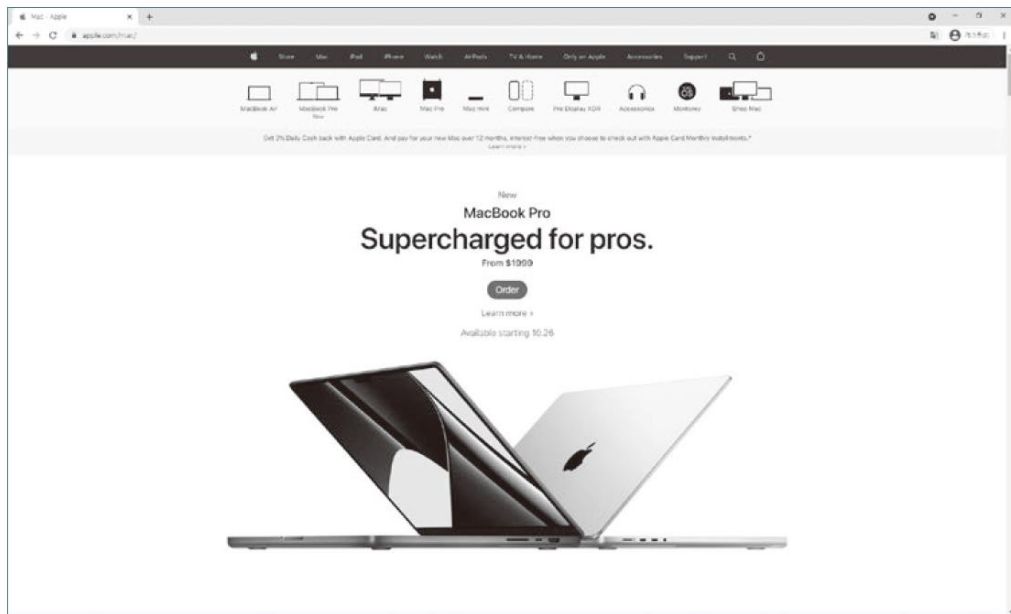


그림 14-10 애플 웹사이트

8. 애니메이션

■ animate()

표 14-16 애니메이션 메소드

메소드	설명
animate()	애니메이션을 적용합니다.

- 스타일에 적용
- 숫자 적용할 수 있는 모든 속성에 animate() 메소드 사용 가능
- 콜백 함수는 애니메이션이 종료되었을 때 호출, 생략 가능함

```
$(선택자).animate(속성, 시간, 콜백_함수)
```

8. 애니메이션

- [예제 14-14] 애니메이션 - 사각형 클릭하면 오른쪽으로 움직임

코드 14-18 애니메이션

Animate.html

```
<!DOCTYPE html>
<html>
<head>
  <title>jQuery Animate</title>
  <style>
    .box {
      /* 크기 */
      width: 100px; height: 100px;
      /* 색상 */
      background-color: red;
      /* 위치 */
      position: absolute;
      left: 10px; top: 10px;
    }
  </style>
```

```
<script src="jquery-3.6.0.min.js"></script>
<script>
  // 웹 페이지를 모두 불러오면
  $(document).ready(function () {
    // .box 태그를 클릭해서
    $('.box').click(function () {
      // 1초 동안 left를 1000픽셀로 변경합니다.
      $(this).animate({
        left: 1000
      }, 1000);
    });
  });
</script>
</head>
<body>
  <div class="box"></div>
</body>
</html>
```

8. 애니메이션

- 스타일에 적용
- 숫자 적용할 수 있는 모든 속성에 `animate()` 메소드 사용 가능
- 콜백 함수는 애니메이션이 종료되었을 때 호출, 생략 가능함

