



13 jQuery 기본

모던 웹을 위한 Javascript jQuery 입문

❖ 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

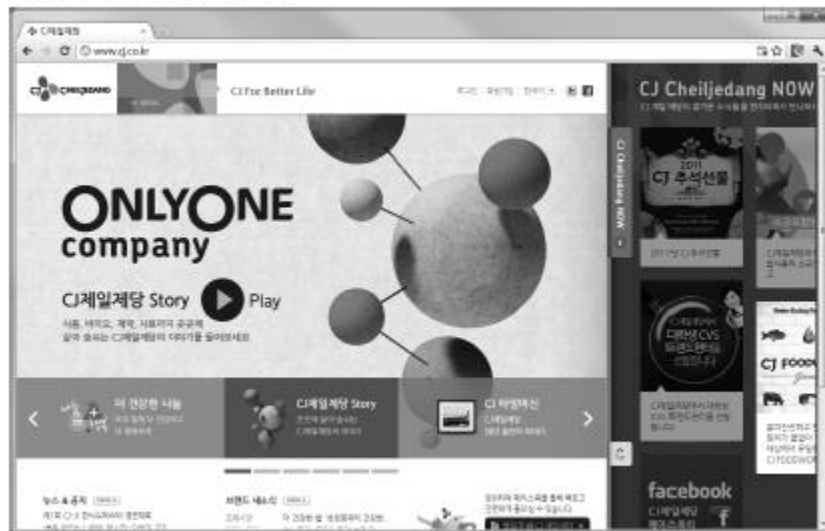


13.1 jQuery에 대해

❖ jQuery 사용의 장점

- 웹 표준만으로도 플래시와 실버라이트로 구현한 웹 사이트와 비슷한 수준의 시각적 효과 구현
 - 아이폰, 아이패드 같은 장치에서도 작동
 - Ex) CJ 제일제당 사이트

그림 13-1 CJ제일제당 사이트

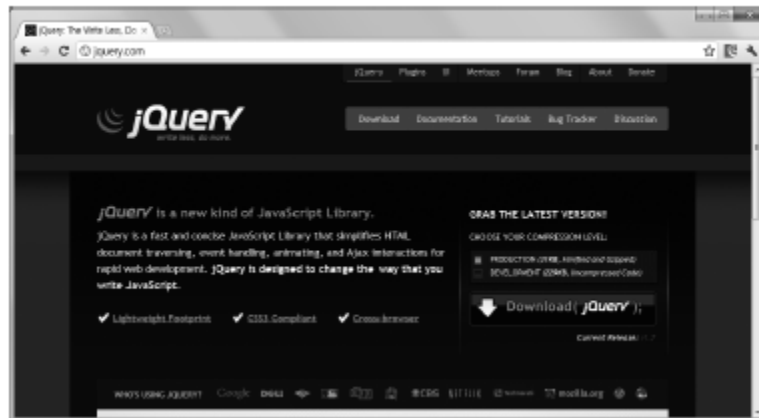


13.1 jQuery에 대해

❖ jQuery 다운로드와 CDN 방식

- 다운받으려면 <http://jquery.com> 접속
- 메인 화면에서 곧바로 jQuery 다운 가능

그림 13-2 <http://jquery.com/>



13.1 jQuery에 대해

❖ jQuery 사용법

- 첫 번째 방법은 CDN 호스트 사용하는 방법
- 두 번째는 직접 다운받아 사용하는 방법

그림 13-3 jQuery의 CDN 호스트 주소

- Google Ajax API CDN (Also supports SSL via HTTPS)
 - <http://ajax.googleapis.com/ajax/libs/jquery/1.7.0/jquery.min.js>
 - [Google Ajax CDN Documentation](#)
- Microsoft CDN (Also supports SSL via HTTPS)
 - <http://ajax.aspnetcdn.com/ajax/jquery/jquery-1.7.min.js>
 - [Ajax CDN Announcement](#), [Microsoft Ajax CDN Documentation](#)
- jQuery CDN (via Media Temple)
 - <http://code.jquery.com/jquery-1.7.min.js> Minified version
 - <http://code.jquery.com/jquery-1.7.js> Source version



❖ CDN 이란?

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



❖ jQuery CDN 호스트 사용해 이용

■ HTML 페이지 구성

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

  </script>
</head>
<body>

</body>
</html>
```

■ script 태그의 src 속성에 제공되는 CDN 호스트 입력

- <http://code.jquery.com/jquery-1.7.js>
- <http://code.jquery.com/jquery-1.7.min.js>
- <http://ajax.googleapis.com/ajax/libs/jquery/1.7/jquery.min.js>
- <http://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.7.js>
- <http://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.7.min.js>



❖ 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



13.1 jQuery에 대해

❖ jQuery 보조 기능

- 마이크로소프트의 Visual Studio 제품군
 - jQuery를 위한 보조 기능 제공
 - Visual Web Developer 2010 Express 도 포함

❖ 보조기능 사용예제

- HTML 페이지 구성
 - <http://Ajax.aspnetcdn.com/Ajax/jQuery/jquery-1.7-vsdoc.js>
 - 마이크로소프트에서 제공하는 jQuery 보조 파일
 - <http://www.asp.net/Ajaxlibrary/cdn.ashx> 참고

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

  </script>
</head>
<body>

</body>
</html>
```



13.1 jQuery에 대해

❖ 보조기능 사용예제

- 페이지 구성 후 보조기능 이용

그림 13-4 jQuery 메서드

jQuery(

jQuery jQuery(String selector, jQuery context)

1: Accepts a string containing a CSS selector which is then used to match a set of elements.

1.1 - \$(selector, context)

1.2 - \$(element)

1.3 - \$(elementArray)

1.4 - \$(jQuery object)

1.5 - \$()

2: Creates DOM elements on the fly from the provided string of raw HTML.

2.1 - \$(html, ownerDocument)

2.2 - \$(html, props)

3: Binds a function to be executed when the DOM has finished loading.

3.1 - \$(callback)

selector: A string containing a selector expression



13.2 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
- jQuery 이벤트 메서드는 이벤트로 여러 개의 함수 연결 가능한 장점

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>
```



13.2 jQuery(document).ready()

❖ \$(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>
```



13.3 jQuery 기본 선택자

❖ jQuery 메서드의 가장 기본적인 형태

- 문서 객체를 다룰 때 사용하는 형태
- jQuery에서 가장 많이 사용하는 형태

그림 13-5 jQuery 메서드의 가장 기본 형태

```
jQuery ('h1').css('color', 'red');
```

jQuery

선택자

메서드



❖ 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>
```



❖ 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>
```



❖ 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>
```



13.3 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>
```



13.3 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>
```



13.3 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>
```



13.3 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>
```



13.4 jQuery 자식 선택자와 후손 선택자

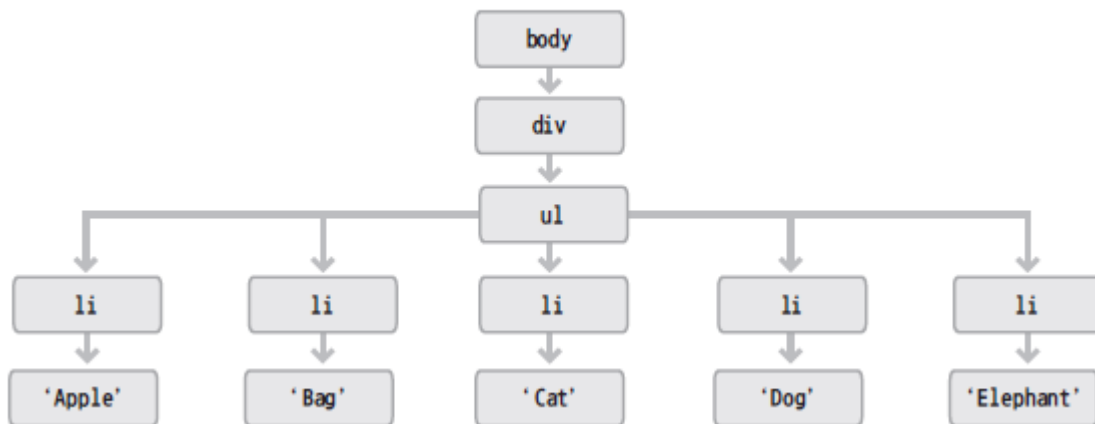
❖ jQuery 자식 선택자와 후손 선택자

- 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



❖ 자식 선택자

- 자식을 선택하는 선택자
- '부모 > 자식'의 형태로 사용
- 예제 코드 13-16
 - body 태그의 자식으로 범위 한정해 전체 선택

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>
```



13.4 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>
```



13.5 jQuery 속성 선택자

❖ 속성 선택자

- 기본 선택자 뒤에 붙여 사용
- jQuery는 표 13-1의 속성 선택자 지원

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.



13.5 jQuery 속성 선택자

❖ 속성 선택자 사용 예제 코드

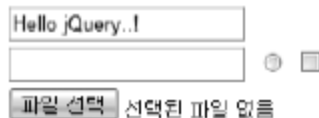
코드 13-18 body 태그 구성

```
<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



13.6 jQuery 입력 양식 필터 선택자

❖ 속성 선택자 사용

- input 태그의 type 속성 따라 문서 객체 선택 가능
- 표 13-2의 필터 선택자 사용
 - 필터 선택자는 기본 선택자 뒤에 사용
 - 'Input:button'의 형태로 선택

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.



13.6 jQuery 입력 양식 필터 선택자

❖ 속성 선택자 사용

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.



13.6 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



13.7 jQuery 필터 선택자

❖ 위치와 관련된 필터 선택자

- 선택자 사용해 문서 객체 선택
- 표 13-4의 위치 지정

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.



13.7 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



13.7 jQuery 필터 선택자

❖ 함수 형태의 필터 선택자

- 표 13-5는 jQuery가 제공하는 함수 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,).



❖ 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



13.8 jQuery를 사용한 배열 관리 (1)

❖ jQuery 사용한 배열 관리

- each() 메서드 사용
 - 매개 변수로 입력한 함수 사용
 - for in 반복문처럼 객체나 배열의 요소 검사하는 메서드
- each() 메서드의 형태
 - 1 \$.each(object, function(index, item){ })
 - 2 \$(selector).each(function(index, item){ })

❖ 예제 코드 13-25]

- 자바스크립트 배열에 들어 있는 내용 HTML 페이지에 표시

코드 13-25 배열 선언

```
<script>
$(document).ready(function () {
    // 변수를 선언합니다.
    var array = [
        { name: 'Hanbit Media', link: 'http://hanb.co.kr' },
        { name: 'Naver', link: 'http://naver.com' },
        { name: 'Daum', link: 'http://daum.net' },
        { name: 'Paran', link: 'http://paran.com' }
    ];
});
</script>
```



13.8 jQuery를 사용한 배열 관리 (1)

❖ \$.each() 메서드

- 첫 번째 매개 변수에는 배열 넣음
- 두 번째 매개 변수는 매개 변수로 index와 item 갖는 함수 넣음
 - Index - 배열의 인덱스 또는 객체의 키 의미
 - 매개 변수 item - 해당 인덱스나 키가 가진 값 의미

코드 13-26 \$.each() 메서드

```
<script>
$(document).ready(function () {
    // 변수를 선언합니다.
    var array = [
        { name: 'Hanbit Media', link: 'http://hanb.co.kr' },
        { name: 'Naver', link: 'http://naver.com' },
        { name: 'Daum', link: 'http://daum.net' },
        { name: 'Paran', link: 'http://paran.com' }
    ];

    // $.each() 메서드를 사용합니다.
    $.each(array, function (index, item) {

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



13.8 jQuery를 사용한 배열 관리 (1)

❖ \$.each() 메서드의 콜백 함수

- item 객체 안에 들어 있는 name, link 속성 사용
 - 링크를 만들어 body 태그 뒷부분에 넣음

코드 13-27 \$.each() 메서드의 콜백 함수

```
// $.each() 메서드를 사용합니다.
$.each(array, function (index, item) {
    // 변수를 선언합니다.
    var output = '';

    // 문자열을 만듭니다.
    output += '<a href="' + item.link + '">';
    output += '    <h1>' + item.name + '</h1>';
    output += '</a>';

    // 집어넣습니다.
    document.body.innerHTML += output;
});
```

그림 13-16 배열로부터 동적으로 생성한 문서 객체

Hanbit Media

Naver

Daum

Paran



13.9 jQuery를 사용한 배열 관리 (2)

❖ jQuery의 배열 객체

- 선택자 사용해 여러 개의 문서 객체 선택할 때 생성
- style 태그에서 high_light 클래스의 background 속성 지정

코드 13-28 HTML 페이지 구성

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

      });
  </script>
</head>
<body>
  <h1>item - 0</h1>
  <h1>item - 1</h1>
  <h1>item - 2</h1>
  <h1>item - 3</h1>
  <h1>item - 4</h1>
</body>
</html>
```



13.9 jQuery를 사용한 배열 관리 (2)

❖ addClass() 메서드

- 문서 객체에 class 속성 추가하는 메서드

코드 13-29 addClass() 메서드

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

❖ removeClass() 메서드

- 문서 객체의 class 속성 제거하는 메서드



13.9 jQuery를 사용한 배열 관리 (2)

❖ each() 메서드

- \$() 메서드 사용해 h1 객체 선택
- body 태그 안에 h1 태그가 다섯 개
 - 다섯 개의 문서 객체 가져옴
- 각 객체에 다르게 설정하고 싶을 때 each() 메서드 사용

코드 13-30 \$(selector).each() 메서드

```
<script>
    $(document).ready(function () {
        $('h1').each(function (index, item) {

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



13.9 jQuery를 사용한 배열 관리 (2)

❖ each() 메서드

- 각 문서 객체에 다른 클래스 적용하는 간단한 예제

코드 13-31 style 태그 구성

```
<style>
    .high_light_0 { background:Yellow; }
    .high_light_1 { background:Orange; }
    .high_light_2 { background:Blue; }
    .high_light_3 { background:Green; }
    .high_light_4 { background:Red; }
</style>
```

코드 13-32 each() 메서드를 사용한 서로 다른 클래스 적용

```
<script>
    $(document).ready(function () {
        $('h1').each(function (index, item) {
            $(item).addClass('high_light_' + index);
        });
    });
</script>
```

코드 13-33 each() 메서드에서의 this 키워드

```
<script>
    $(document).ready(function () {
        $('h1').each(function (index, item) {
            $(this).addClass('high_light_' + index);
        });
    });
</script>
```

그림 13-17 각 클래스가 적용돼 색상이 나타납니다.

item - 0

item - 1

item - 2

item - 3

item - 4



13.9 jQuery를 사용한 배열 관리 (2)

❖ addClass() 메서드의 매개 변수 활용

- 코드 13-34처럼 함수도 입력 가능

코드 13-34 addClass() 메서드의 특수한 사용 방법

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



13.10 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>
```



❖ \$.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



13.11 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>
```





Thank You !

모던 웹을 위한 Javascript jQuery 입문