



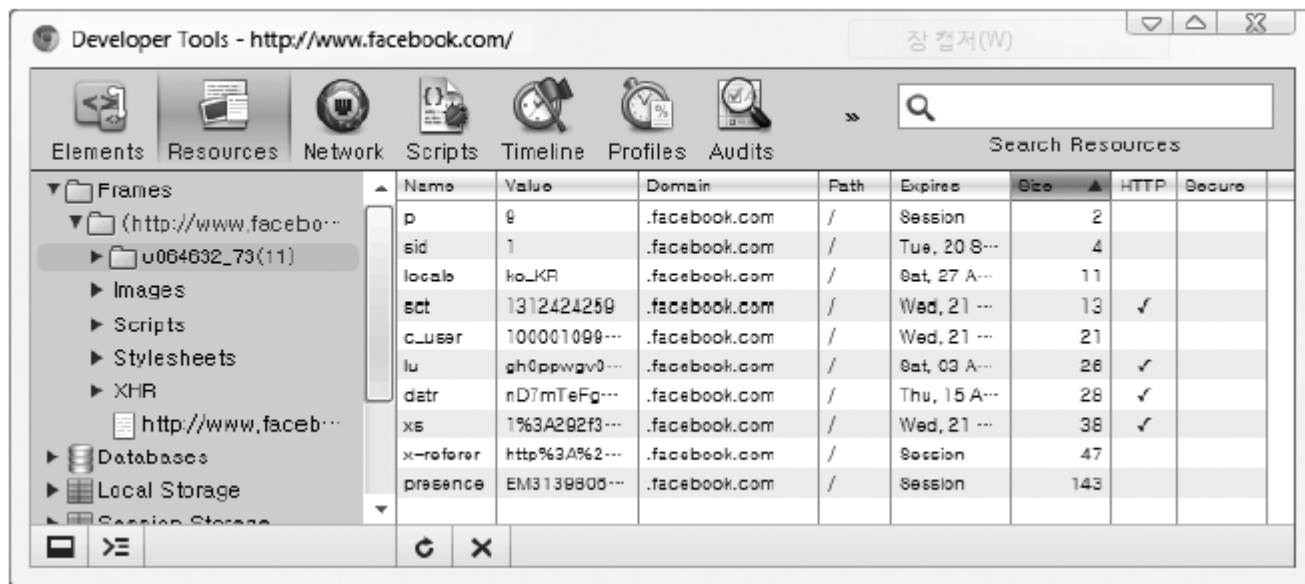
C 쿠키와 저장소

모던 웹을 위한 Javascript jQuery 입문

❖ 쿠키란?

- 키와 값이 들어 있는 작은 데이터 조각
 - 이름, 값, 파기 날짜와 경로 정보 포함
 - 서버와 클라이언트에서 모두 저장, 사용 가능
 - 일정 기간 동안 데이터 저장 가능
 - 로그인 유지하는 페이스북 같은 웹 사이트에서 사용
- 크롬으로 접속한 후 요소 검사의 Resources 탭 -> 좌측 탭 Cookies

그림 C-1 크롬 요소 검사를 사용한 쿠키 검사



❖ 쿠키의 속성

- 지정할 수 있는 속성은 Name, Value, Domain, Path, Expires, Secure
- document 객체의 cookie 속성에 입력해 생성
 - `document.cookie = 'Name=Value;Expires=날짜;Domain=도메인;Path=경로;Secure';`
- Name, Value 속성
 - 데이터를 저장, 읽는데 사용, 쿠키 사용시 필수 지정
- Expires 속성
 - 쿠키의 파기 날짜 지정하는 속성 (GMT 형식이나 UTC 형식)
 - 입력하지 않으면 브라우저가 종료될 때 쿠키 삭제
- Secure 속성
 - 쿠키가 보안 연결을 사용해 전송돼야 하는지 결정
 - Secure 속성을 지정하면 해당 쿠키는 SSL을 사용해서만 요청 가능



❖ 쿠키의 속성

■ Domain 속성과 Path 속성

- 입력하지 않으면 현재 도메인의 현재 경로로 자동 입력
- 페이지 요청과 비교해 도메인의 경로와 Domain 속성이 일치하지 않으면?
 - 쿠키에 접근하는 것을 막으므로 **Domain 속성은 건드리지 않음**
 - **Path 속성은 일부러 지정하는 경우가 많음**



❖ 예제 코드 C-1]

- 쿠키를 생성하는 가장 기본적인 방법
- document 객체의 cookie 속성에 문자열을 넣으면?
 - 문자열의 쿠키 정보를 별도로 뽑아 저장
 - Name 속성은 CookieName, Value 속성은 Value
 - Expires 속성은 현재 날짜에서 7일을 더한 날로 설정

코드 C-1 쿠키 생성

```
<script>
// 변수를 선언합니다.
var date = new Date();
date.setDate(date.getDate() + 7);

var willCookie = '';
willCookie += 'CookieName=Value;';
willCookie += 'expires=' + date.toUTCString();

// 쿠키에 넣습니다.
document.cookie = willCookie;
</script>
```

그림 C-3 인터넷 익스플로러의 쿠키 검사 결과

NAME	CookieName
VALUE	Value
DOMAIN	localhost
PATH	/Example/
EXPIRES	2011-08-29 오후 1:20:52



❖ 예제 코드 C-1]

- Path 속성
 - 현재 폴더 또는 현재 폴더의 하위에서 접근 가능
 - Path 속성을 /로 설정하면 도메인 내의 모든 곳에서 접근 가능
 - 쿠키의 접근 제한 범위를 좁게 잡을수록 보안에 좋음

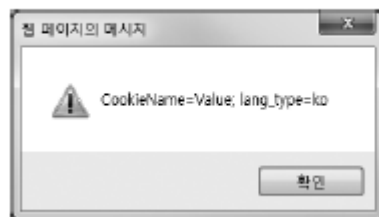
❖ 예제 코드 C-2]

- 예제 C-1에서 내용을 지우고 출력해도 값은 유지됨

코드 C-2 document 객체의 cookie 속성

```
<script>  
    // 출력합니다.  
    alert(document.cookie);  
</script>
```

그림 C-4 실행 결과



❖ 쿠키를 제거하는 방법

- Expires 속성을 현재 이전의 날짜로 설정하면 간단함

코드 C-3 쿠키 제거

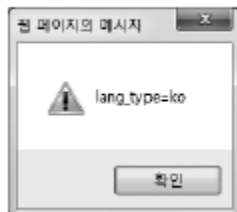
```
<script>
    // 변수를 선언합니다.
    var date = new Date();
    date.setDate(date.getDate() - 1);

    var willCookie = '';
    willCookie += 'CookieName=Value;';
    willCookie += 'expires=' + date.toUTCString();

    // 쿠키에 집어넣습니다.
    document.cookie = willCookie;

    // 출력합니다.
    alert(document.cookie);
</script>
```

그림 C-5 쿠키가 삭제되었습니다.



❖ 개별적인 쿠키를 읽는 방법

- 예제 코드 C-4처럼 코드를 작성해 쿠키 입력

코드 C-4 쿠키 다섯 개 생성

```
<script>
    // 변수를 선언합니다.
    var date = new Date();
    date.setDate(date.getDate() + 7);

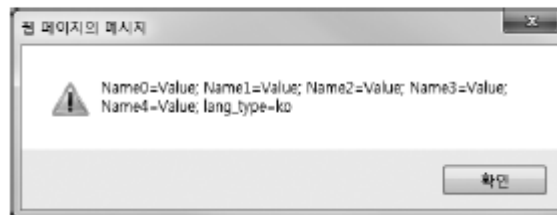
    // 5번 반복합니다.

    for (var i = 0; i < 5; i++) {
        var willCookie = '';
        willCookie += 'Name' + i + '=Value;';
        willCookie += 'expires=' + date.toUTCString();

        // 쿠키에 집어넣습니다.
        document.cookie = willCookie;
    }

    // 출력합니다.
    alert(document.cookie);
</script>
```

그림 C-6 document 객체의 cookie 속성



❖ 개별적인 쿠키를 읽는 방법 (2)

- Name 속성이 Name0인 쿠키의 Value 속성 추출
- 예제 코드 C-5
 - Name 속성이 Name0인 쿠키의 Value 속성 추출
 - 문자열을 세미콜론(;) 사용해 분해
 - “Name=Value” 형태의 문자열
 - 첫 위치와 마지막 위치의 공백을 제거해 깔끔히 출력
 - » `cookies[i].replace(name + '=', '').replace(/^\\s\\s*/, '').replace(/\\s\\s*$/, '')`

코드 C-5 쿠키 정보 출력

```
<script>
    var name = 'Name0';
    var cookies = document.cookie.split(';');
    alert(cookies);
    // 쿠키를 추출합니다.
    for (var i in cookies) {
        if (cookies[i].search(name) != -1) {
            alert(cookies[i].replace(name + '=', ''));
        }
    }
</script>
```



❖ 쿠키 사용 예제 코드

코드 C-6 쿠키 관련 함수

```
<!DOCTYPE html>
<html>
<head>
  <script>
    function setCookie(name, value, day) {
      // 변수를 선언합니다.
      var date = new Date();
      date.setDate(date.getDate() + day);

      var willCookie = '';
      willCookie += name + '=' + encodeURIComponent(value) + ';';
      willCookie += 'expires=' + date.toUTCString() + ';';

      // 쿠키에 집어넣습니다.
      document.cookie = willCookie;
    }

    function getCookie(name) {
      // 변수를 선언합니다.
      var cookies = document.cookie.split(';');

      // 쿠키를 추출합니다.
      for (var i in cookies) {
        if (cookies[i].search(name) != -1) {
          return decodeURIComponent(cookies[i].replace(name + '=', ''));
        }
      }
    }
  </script>
</head>
</html>
```



❖ 쿠키 사용 예제 코드 (2)

```
    }  
  }  
  
  function removeCookie(name) {  
    // 변수를 선언합니다.  
    var date = new Date();  
    date.setDate(date.getDate() - 1);  
  
    var willCookie = '';  
    willCookie += 'CookieName=Value;';  
    willCookie += 'expires=' + date.toUTCString();  
  
    // 쿠키에 집어넣습니다.  
    document.cookie = willCookie;  
  }  
</script>  
<script>  
  alert(getCookie('Cookie'));  
  setCookie('Cookie', 'Cheese', 7);  
  alert(getCookie('Cookie'));  
</script>  
</head>  
<body>  
  
</body>  
</html>
```



❖ 쿠키 사용 예제 코드 (2)

- 처음 코드 실행
 - 첫 경고창에는 Cookie에 해당하는 데이터가 없으므로 undefined 출력
 - 두 번째 경고창에는 Cheese 출력
- 한번 더 코드 실행
 - 두 경고창 모두에서 Cheese 출력



❖ jQuery Cookie 플러그인

- 편리하게 쿠키 관련 메서드 제공
- <https://github.com/carhartl/jquery-cookie> 에서 다운
- jQuery Cookie 플러그인을 사용하려면?
 - 예제 코드 C-7처럼 head 태그 구성

그림 C-7 폴더 구성



코드 C-7 head 태그 구성

```
<head>
  <script src="http://code.jquery.com/jquery-1.7.js"X/script>
  <script src="jquery.cookie.js"X/script>
  <script>
    $(document).ready(function () {

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



❖ jQuery Cookie 플러그인

- 핵심 메서드는 \$.cookie()
 - cookie() 메서드의 사용방법
 - 순서대로 쿠키를 저장하고 읽고 삭제
 - 매개 변수에 따라 기능이 다르므로 주의 !!!

코드 C-8 쿠키의 저장과 삭제

```
<script>
$(document).ready(function () {
    // 쿠키를 저장합니다.
    $.cookie('MyKey', 'Hello');

    // 쿠키를 읽습니다.
    alert($.cookie('MyKey'));

    // 쿠키를 삭제합니다.
    $.cookie('MyKey', null);

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



❖ 옵션 객체에 입력할 수 있는 속성

- 쿠키를 저장하는 형태의 메서드
 - 옵션 객체는 세 번째 매개 변수

표 C-1 jQuery Cookie 플러그인의 옵션 속성

옵션 속성 이름	설명
expires	쿠키를 유지하고자 하는 일수를 지정합니다.
path	쿠키의 대상 경로를 지정합니다.
domain	쿠키의 대상 도메인을 지정합니다.
secure	쿠키의 보안 유무를 지정합니다.
raw	인코딩과 관련된 속성입니다. true로 설정하면 인코딩하지 않습니다.



❖ 옵션 객체의 사용 예

- raw 속성은 거의 사용하지 않음
 - 인코딩할 것이냐 안 할 것인지에 대한 속성

코드 C-9

```
<script>
$(document).ready(function () {
    // 쿠키를 저장합니다.
    $.cookie('MyKey', '안녕', {
        expires: 7,
        path: '/',
        raw: true
    });
    // 출력합니다.
    $.cookie('MyKey');
});
</script>
```



❖ 저장소 기능의 필요

- 기존의 자바스크립트는 쿠키 이외에도 다른 방법으로 데이터 저장
- 플래시 저장소를 사용
- 각각의 브라우저에 존재하는 특수 저장소 사용
- HTML5부터는 저장소 기능 기본 제공
 - 저장소 기능은 다음 브라우저 버전부터 사용 가능
 - 인터넷 익스플로러 8
 - 파이어폭스 3.5
 - 사파리 4.0
 - 크롬 4.0
 - 오페라 10.5



❖ 저장소

- 로컬 저장소와 세션 저장소 두 가지
- 두 저장소는 사용 방법은 같음
 - 로컬 저장소: 웹 브라우저를 삭제하지 않는 이상 데이터를 영구적으로 저장
 - 세션 저장소: 웹 브라우저가 종료될 때까지만 데이터를 저장
- 두 저장소 모두 window 객체 안에 들어 있음
 - jQuery와 저장소와는 관계가 없으나 편하게 사용 가능하게 도와줌

코드 C-10 head 태그 구성

```
<head>
  <script src="http://code.jquery.com/jquery-1.7.js"X/script>
  <script>
    $(document).ready(function () {
      // 로컬 저장소를 사용합니다.
      if (window.localStorage) {

      } else {
        alert('로컬 저장소를 사용할 수 없는 브라우저입니다.');      }
    });
  </script>
</head>
```



❖ body 태그 구성

- 데이터 입력후 데이터를 확인할 수 있게 작성

코드 C-11 body 태그 구성

```
<body>
  <input type="text" id="key" />
  <input type="text" id="value" />
  <input type="button" value="저장" />
  <div id="output">

  </div>
</body>
```



❖ printLocalStorage() 함수

- localStorage 객체 내부에 있는 데이터를 div 태그에 추가
- 로컬 저장소
 - 일반 객체처럼 객체처럼 쉽게 데이터를 뽑고 추가 가능

코드 C-12 로컬 저장소

```
<script>
    $(document).ready(function () {
        // 로컬 저장소의 내용을 출력합니다.
        function printLocalStorage() {
            // div 태그를 비웁니다.
            $('div').empty();

            // div 태그에 로컬 저장소의 내용을 출력합니다.
            for (var key in localStorage) {
                $('<h1>X/h1>').html(key + ': ' + localStorage[key]).appendTo('div');
            }
        }

        // 로컬 저장소를 사용합니다.
        if (window.localStorage) {
        } else {
            alert('로컬 저장소를 사용할 수 없는 브라우저입니다.');
```



❖ localStorage 객체의 메서드

- 데이터를 생성하고 읽고 제거할 수 가능
- 숫자나 불리언을 저장해도 저장소에는 무조건 문자열로 저장

표 C-2 localStorage 객체의 메서드

메서드 이름	설명
setItem(key, value)	로컬 저장소에 데이터를 생성합니다.
getItem(key)	로컬 저장소의 데이터를 읽습니다.
removeItem(key)	로컬 저장소의 데이터를 삭제합니다.
clear()	로컬 저장소의 모든 데이터를 삭제합니다.
key(number)	로컬 저장소의 특정 위치의 데이터를 읽습니다.



❖ 예제 코드 C-13]

- 버튼 누르면 입력 양식에서 키와 값 읽어 로컬 저장소에 추가
- 추가 후 로컬 저장소 내용 출력

코드 C-13 이벤트 연결

```
// 로컬 저장소를 사용합니다.
if (window.localStorage) {
    printLocalStorage();
    $('input:button').click(function () {
        // 변수를 선언합니다.
        var key = $('#key').val();
        var value = $('#value').val();

        // 데이터를 추가합니다.
        localStorage.setItem(key,value);

        // 데이터를 출력합니다.
        printLocalStorage();
    });
} else {
    alert('로컬 저장소를 사용할 수 없는 브라우저입니다.');
```

```
}
```



❖ 예제 코드 C-13]

- 코드를 실행하고 값을 계속 입력
 - 그림 C-8처럼 데이터 계속 추가
 - 브라우저 종료 후 다시 실행

– 로컬 저장소에 데이터를 저장했으므로 데이터 영구 저장

그림 C-8 로컬 저장소의 사용



Language JavaScript 저장

MyValue: HelloLocalStorage

Language: JavaScript



❖ 세션 저장소

- 같은 메서드를 가지지만 브라우저를 종료할 때 데이터가 모두 사라짐

코드 C-14 세션 저장소

```
<script>
$(document).ready(function () {
    // 세션 저장소를 사용합니다.
    if (window.sessionStorage) {

    } else {
        alert('세션 저장소를 사용할 수 없는 브라우저입니다.');
```





Thank You !

모던 웹을 위한 Javascript jQuery 입문