

AJAX

목차

1. javascript
2. Web 2.0 / Symantic Web / RIA
3. AJAX 와 XMLHttpRequest
4. jQuery의 편리함과 생산성
Write less, do more
5. jQuery .ajax



- Jesse James Garrett
- Ajax: A New Approach to Web Applications
- Feb 18, 2005

■ 참고

- <http://www.adaptivepath.com/publications/essays/archives/000385.php>

Ajax 개요

❖ Ajax 개요

- Ajax는 자바스크립트처럼 특정한 프로그래밍 언어를 지칭하는 것이 아닌 구현하는 방식
- Ajax 방식으로 구현된 웹 사이트

그림 19-1 Ajax 예



Ajax 개요

❖ 검색 사이트에서 자동 완성 기능

■ Ajax의 가장 대표적인 예

- 기존 웹 페이지는 새로운 데이터를 보여주려면 항상 페이지 전환 필요

❖ 포털 사이트에 로그인할 때

■ 아이디와 비밀번호 입력

■ ‘로그인’ 버튼을 누르는 순간 데이터가 서버로 전송

■ 서버에서는 아이디와 비밀번호가 일치하는지 확인

- 일치하면 로그인 된 페이지 출력
- 일치하지 않으면 비밀번호 재입력을 요구하는 페이지 출력

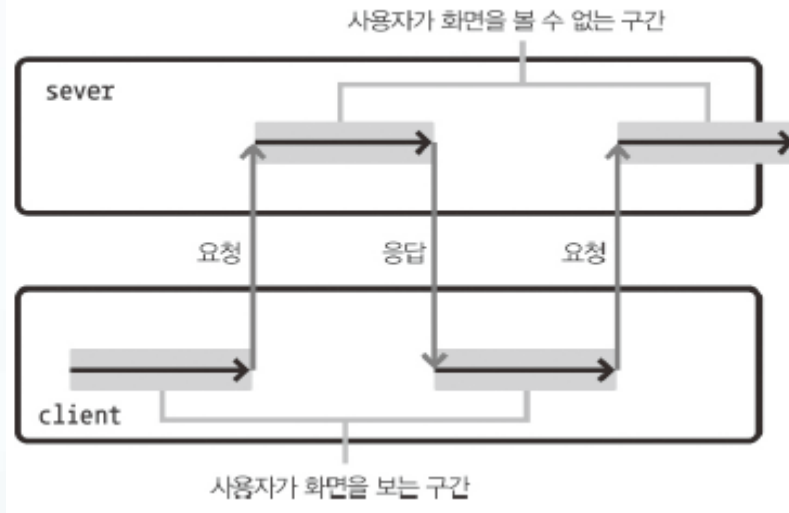
■ 모바일은 페이지가 전환되는 1~2초 동안 흰 페이지 출력

19.1 Ajax 개요

❖ Ajax 개요

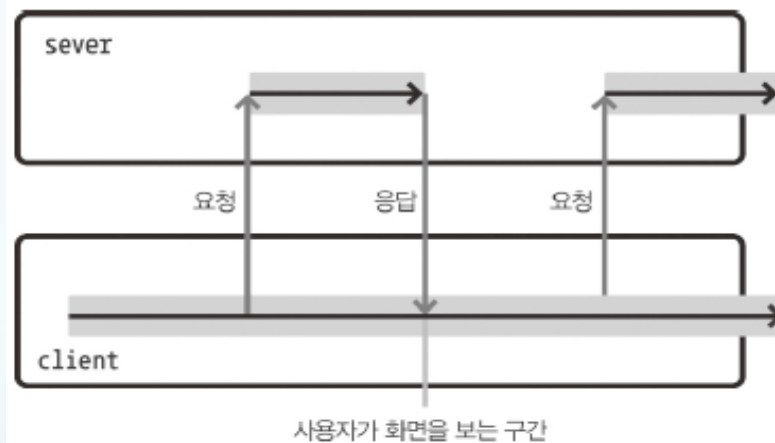
■ 기존의 방식

그림 19-2 기존의 방식



■ Ajax 방식

그림 19-3 Ajax 방식



19.1 Ajax 개요

❖ Ajax 개요

- 가장 간단한 예를 보면 페이스북이나 트위터 같은 SNS를 들 수 있음

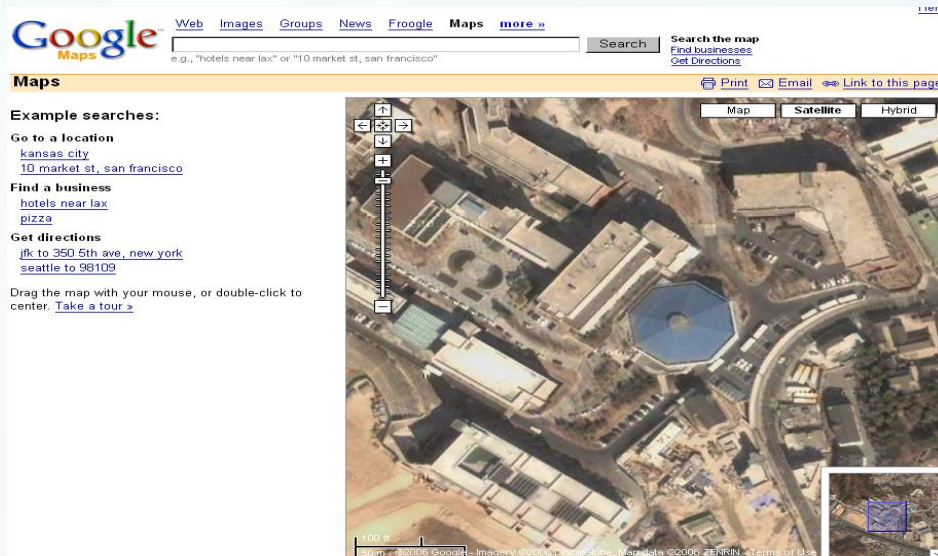
그림 19-4 페이스북



- AJAX 개요

- 비동기적 Javascript 언어와 XML 등의 기타 웹 표준을 종합적으로 사용하는 기술로서 Web 2.0의 핵심 기술.
- AJAX의 개별 기술은 신기술이 아니지만 이들을 종합적으로 활용하는 기술이 핵심임.(기존의 기술들을 효율적으로 조합해 사용)
- 구글맵과 더불어 2005년 부터 AJAX라는 용어의 등장과 함께 웹 분야에서 주목을 받고 있는 최신 기술임
- JavaScript, XML, HTML, DHTML, XmlHttpRequest, DOM, CSS등을 기반으로 작성됨

- AJAX 사용예(<http://maps.google.com>)
 - Ajax 로만 구현
 - 완벽한 멀티 플랫폼 서비스
 - 위성사진 제공
 - API공개로 각종 변종 서비스 제공
 - MSN Virtual Earth, A9 Maps 등 경쟁서비스 붐



- AJAX 사용예 (Google Suggest)

(<http://www.google.com/webhp?complete=1&hl=en>)

- 검색어에 대한 추천단어 서비스
- Ajax를 이용한 대표적 서비
- 국내 포탈도 동일서비스 제공



[Web](#) [Images](#) [Groups](#) [News](#) [Froogle](#) [Maps](#) [more »](#)

As you type, G

kookmin university	
kookmin bank	64,900 results
kookmin university	23,000 results
kookmin bank korea	31,200 results
kookmin bank seoul	12,900 results
kookmin bank new zealand	
kookmin bank swift code	
kookmin bank english	8,340 results
kookmin bank new york	
kookmin bank routing number	119 results
kookmin card	13,500 results

[Advanced Search](#)
[Preferences](#)
[Language Tools](#)

ts. [Learn more](#)

• AJAX 사용 예(DAUM 주소록)

- 주소정렬, 검색, 편집 시 DOM과 Ajax 사용
- 주소를 두 번 클릭하면 간단하게 정보를 수정.
- 메일 검색 시에 순차적으로 검색결과를 가져온다

The screenshot displays the DAUM address book interface. At the top, there are tabs for sorting and filtering: 전체, ㄱ, ㄴ, ㄷ, ㄹ, ㅁ, ㅂ, ㅅ, ㅇ, ㅈ, ㅊ, ㅋ, ㅌ, ㅍ, ㅎ, A-Z, and 0-9. Below these, there are buttons for '선택한 주소를' (Selected address), '그룹선택' (Group select), '로' (to), '복사' (Copy), '이동' (Move), '삭제' (Delete), '상세보기' (View details), '메일보내기' (Send email), '문자보내기' (Send message), and '개인추가' (Add personal).

The main area is divided into three columns: '그룹(42)' (Group), '이름' (Name), '이메일' (Email), and '연락처' (Contact). The '그룹' column shows a tree view of folders, including '전체 (811)', '미분류 (123)', 'CJ Project (8)', 'CJ Project PL (6)', 'CJ Project PMO (6)', 'EDUTEC (5)', 'HP교육센터-J.. (7)', 'Hunter (14)', 'KH교육원 (1)', 'SKTelink MD (15)', and 'SK폰빌프로젝트 (11)'. The '이름' column lists contacts with checkboxes, and the '이메일' column shows their email addresses. The '연락처' column has a '추가' (Add) button.

그룹(42)	이름	이메일	연락처
<input type="checkbox"/>			<input type="text"/> <input type="button" value="추가"/>
<input type="checkbox"/>	물류	logi@cjproject.co.kr	<input type="text"/>
<input type="checkbox"/>	생산1	fac1@cjproject.co.kr	<input type="text"/>
<input type="checkbox"/>	생산2	fac2@cjproject.co.kr	<input type="text"/>
<input type="checkbox"/>	영업	sale@cjproject.co.kr	<input type="text"/>
<input type="checkbox"/>	인사	hrm@cjproject.co.kr	<input type="text"/>
<input type="checkbox"/>	회계	acct@cjproject.co.kr	<input type="text"/>

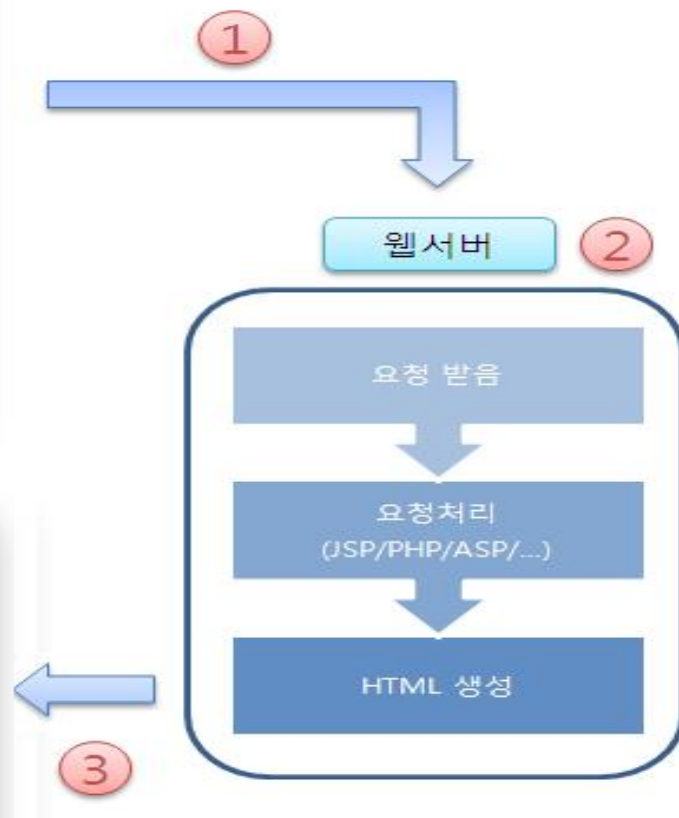
- 장점

- 페이지 이동 없이 빠르게 화면을 전환
- 서버처리를 기다리지 않고 비동기 요청이 가능
- 수신하는 데이터 양을 줄일 수 있음
- **ActiveX**나 플래시 등의 플러그인 없이도 **Interactive**한 웹 페이지 작성
- **Reload** [새로고침]가 필요 없는 웹 페이지 작성 가능.
- **Ajax** 는 서버 기술을 의식하지 않습니다.

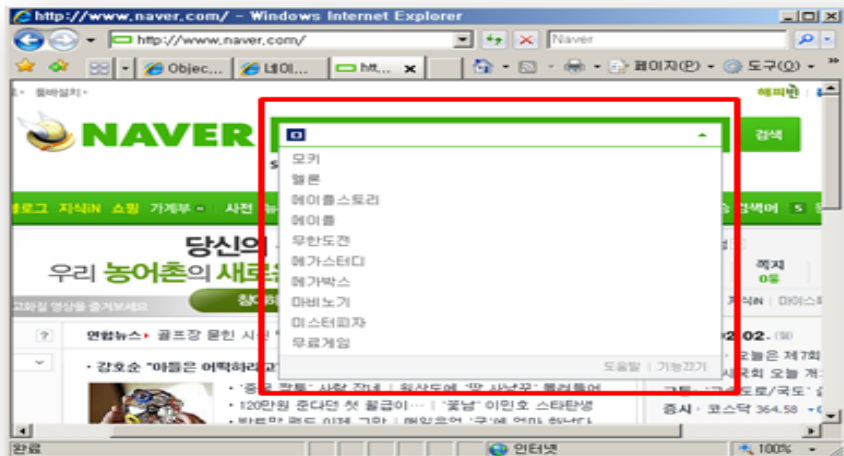
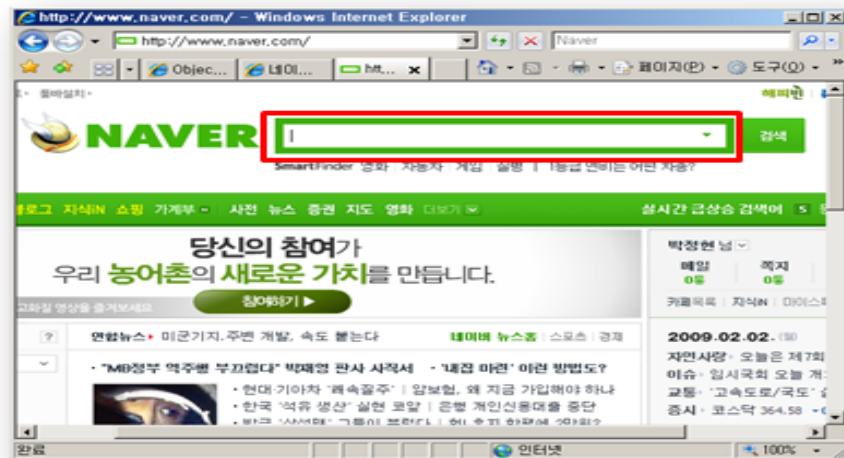
- 일반 Web Application 의 특징
 - Click, Wait, and Refresh
 - 동기식 통신 모델
 - request / response
 - Page 단위로 갱신 ➔ Page-driven
- 일반 Web Application의 문제점
 - 느린 응답시간과 화면 refresh
 - page 갱신 동안 이전 내용의 분실
 - 과도한 서버의 load로 인한 bandwidth의 소비
 - ➔ Ajax가 등장하게 된 동기

- 일반 Web의 통신방법

기존 웹 페이지



- Ajax 의 통신방법



XMLHttpRequest

웹 서버

요청 받음

요청 처리

XML 생성
(또는 텍스트)

- XMLHttpRequest의 개요
 - MS(마이크로 소프트)는 Internet Explorer(IE)가 자바스크립트로부터 XML을 불러올 수 있도록 XMLHttpRequest를 설계.
 - XMLHttpRequest객체는 현재 대부분의 브라우저에 내장되어 있는 객체이다.
 - W3C 의 표준이 아니므로, IE에서는 ActiveX Component 형식으로 구현되었고, 다른 브라우저들(FireFox, Safari, Opera)은 native javascript 객체로 구현.
 - XMLHttpRequest는 실제로 자바스크립트에 대한 일반적인 HTTP 클라이언트 이기 때문에 GET과 POST HTTP 요청을 서버에 할 수 있다.
 - 클라이언트와 서버간의 통신을 담당하는 객체.

- XMLHttpRequest의 개요
 - XMLHttpRequest는 간단한 API와 몇 가지 메서드와 속성들을 제공한다. 그러나 브라우저간의 차이점이 존재하기 때문에 “Cross-Browser”(브라우저의 종류에 관계없이 동작) 부분을 생각할 필요가 있다.
 - 내용의 변경 없이 새로운 콘텐츠를 불러올 수 있다.
 - JavaScript로 동기 방식의 호출을 할 수 있다.

- XMLHttpRequest 객체 생성
 - Internet Explorer ActiveXObject
 - IE는 ActiveXObject를 사용해서 서버와 통신하며 2가지 객체가 있다
 - Msxml2.XMLHTTP
 - Microsoft.XMLHTTP
 - Msxml2.XMLHTTP는 IE 5.0 이후의 버전이고 Microsoft.XMLHTTP는 IE 5.0이전 버전이다.
 - 가능한 두 종류를 보장해서 XMLHttpRequest를 정의할 필요가 있다. 왜냐하면 “Microsoft.XMLHTTP”를 사용하지 않는다는 보장을 할 수 없기 때문이다.

- XMLHttpRequest 객체 생성
 - Internet Explorer ActiveXObject
 - IE는 ActiveXObject를 사용해서 서버와 통신하며 2가지 객체가 있다
 - Msxml2.XMLHTTP
 - Microsoft.XMLHTTP
 - Msxml2.XMLHTTP는 IE 5.0 이후의 버전이고 Microsoft.XMLHTTP는 IE 5.0이전 버전이다.
 - 가능한 두 종류를 보장해서 XMLHttpRequest를 정의할 필요가 있다. 왜냐하면 “Microsoft.XMLHTTP”를 사용하지 않는다는 보장을 할 수 없기 때문이다.

■ ActiveXObject 객체 생성 예

```
if (window.ActiveXObject) {      //IE
    try {
        return new ActiveXObject("Msxml2.XMLHTTP");
    } catch(e) {
        try {
            return new ActiveXObject("Microsoft.XMLHTTP");
        } catch(e) {
            return null;
        }
    }
}
```

- Internet Explorer ActiveXObject 이외
 - IE 이외 브라우저를 사용하여 애플리케이션을 실행하면 XMLHttpRequest 객체가 생성되지만, IE 브라우저로 애플리케이션을 실행하면 XMLHttpRequest 객체가 생성되지 않는다.
- XMLHttpRequest 객체 생성 예

```
if (window.XMLHttpRequest) {      //IE 이외(FireFox, Opera등)
    try {
        return new XMLHttpRequest()
    } catch(e) {
        return null;
    }
}
```



- Cross-Browsing
 - 일반적으로 많이 사용되는 브라우저는 IE이다. 그러나 IE 브라우저를 많이 사용하는 것이지 브라우저가 IE만 있는 것은 아니다.
 - Firefox, Netscape, Safari, Opera, Konqueror
 - 브라우저마다 XMLHttpRequest 객체의 명칭이 다르다.
 - 각기 다른 객체를 통합하여 하나로 만들어야 사용자가 어떤 브라우저를 사용하더라도 이를 대응할 수 있다.
 - 이렇게 브라우저에 상관없이 동작하는 것을 크로스 브라우징(Cross-Browsing)이라고 한다.

- Cross-Browsing 예

```
function createXMLHttpRequest(){
    var reqHttp;
    if (window.ActiveXObject) {
        try { reqHttp = new ActiveXObject("Msxml2.XMLHTTP");
        } catch(e) {
            try {
                reqHttp = new ActiveXObject("Microsoft.XMLHTTP");
            } catch(e1) {
                reqHttp = null;
            }
        }
    } else if (window.XMLHttpRequest) {
        try{
            reqHttp = new XMLHttpRequest()
        } catch(e1) {
            reqHttp = null
        }
    } else {
        reqHttp = null;
    }
}
```


■ crossbrowser.html

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3  ["http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4  <html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
5  <head>
6  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
7  <title>XMLHttpRequest 검사</title>
8  <link rel="stylesheet" href="commcss.css" type="text/css" />
9  <script language="javascript" type="text/javascript" >
10 // XMLHttpRequest 생성
11 function createXMLHttpRequest() {
12     var xmlhttp;
13     if (window.ActiveXObject) { // IE
14         try {
15             xmlhttp = new ActiveXObject("Msxml2.XMLHTTP");
16         } catch (e) {
17             try {
18                 xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
19             } catch (e1) {
20                 xmlhttp = null;
21             }
22         }
23     } else if (window.XMLHttpRequest) { // IE 이외
24         try {
25             xmlhttp = new XMLHttpRequest();
26         } catch (e) {
27             xmlhttp = null;
28         }
29     }
30     if (xmlhttp == null) errorMessage(); //XMLHttpRequest 생성 실패
31     return xmlhttp;
32 }
33
34 // 지원할 수 없는 브라우저 사용
35 function errorMessage() {
36     alert("지원할 수 없는 브라우저입니다.");
37 }
38
```

■ crossbrowser.html

```
39 // XMLHttpRequest 생성 체크
40 function openSendStatus() {
41     var xmlHttp = createXMLHttpRequest();    // XMLHttpRequest 생성
42     if (xmlHttp != null) {                    // XMLHttpRequest 생성 성공
43         document.getElementById("rcvData").innerHTML = "XMLHttpRequest가 생성되었습니다.";
44     }
45     if (xmlHttp == null) {                    // XMLHttpRequest 생성 실패
46         document.getElementById("rcvData").innerHTML = "지원할 수 없는 브라우저입니다.";
47     }
48 }
49 </script>
50 <!--
51 <script language="javascript" type="text/javascript" src="createXMLHttpRequest.js"></script>
52 -->
53 </head>
54
55 <body>
56 <h1>XMLHttpRequest 검사</h1>
57
58 <div id="btnarea">
59     <input type="button" id="btnclick" name="btnclick" value="XMLHttpRequest 생성"
60         onclick="openSendStatus()" />
61 </div>
62
63 <div id="datashow">
64     <div id="rcvData"></div>
65 </div>
66
67 </body>
68 </html>
```

- XHR(XMLHttpRequest) 속성
 - onreadystatechange
 - 자바스크립트 콜백함수(function pointer)를 지정한다.
 - 콜백함수는 readyState 값이 변할 때 마다 호출된다.
 - 요청이 서버로 보내지면 readyState 는 5가지 숫자 값으로 계속 변화가 일어나게 된다.
 - readyState
 - 요청의 상태를 의미한다.

값	의 미	설 명
0	uninitialized	객체만 생성되고 아직 호출되지 않음
1	loading	open 메서드가 호출되고 아직 send메서드가 호출되지 않음
2	loaded	send 메서드 호출, status와 헤더는 도착하지 않음
3	interactive	데이터의 일부를 받은 상태
4	complete	데이터를 전부 받음



– responseText

- 서버의 응답을 string 형식으로 나타낸다. 단순 text 를 innerHTML 속성으로 표현하기에는 알맞지만 논리적으로 파싱하거나 동적으로 페이지 콘텐츠를 생성하기는 힘들다.

– responseXML

- 서버의 응답을 XML 로 나타낸다. 이 속성은 DOM 객체로 파싱할 수 있다.

– status

- 서버로부터의 HTTP 상태코드이다.(예 200(OK), 404(NOT Found), 202(결과 값이 없을 때)등등)

– statusText

- HTTP 상태코드에 대한 텍스트 값이다.(예 OK, NOT Found 등등)

■ createXMLHttpRequest.js

```
function createXMLHttpRequest(){
    var reqHttp;
    if (window.ActiveXObject) {
        try { reqHttp = new ActiveXObject("Msxml2.XMLHTTP"); }
        catch(e) {
            try {
                reqHttp = new ActiveXObject("Microsoft.XMLHTTP");
            } catch(e1) {
                reqHttp = null;
            }
        }
    } else if (window.XMLHttpRequest) {
        try{
            reqHttp = new XMLHttpRequest()
        } catch(e1) {
            reqHttp = null
        }
    } else {
        reqHttp = null;
    }
    if (reqHttp == null) alert("지원할 수 없는 브라우저!");
    return reqHttp;
}
```

- simpleRequest.html

```
<html xmlns="http://www.w3.org/1999/xhtml">
<script type="text/javascript" src="createXMLHttpRequest.js"></script>
<script type="text/javascript">
var xmlHttp;
function startRequest() {
    xmlHttp = createXMLHttpRequest();
    xmlHttp.onreadystatechange = handleStateChange;
    xmlHttp.open("GET", "simpleResponse.xml", true);
    xmlHttp.send(null);
}
function handleStateChange() {
    if(xmlHttp.readyState == 4) {
        if(xmlHttp.status == 200) {
            alert("The server replied with: " + xmlHttp.responseText);
        }
    }
}
</script>
<body>
    <form action="#">
        <input type="button" value="Start Basic Asynchronous Request" onclick="startRequest();"/>
    </form>
</body>
</html>
```

- simpleResponse.xml

```
<고객>  
<이름>홍길동</이름>  
<나이>30</나이>  
</고객>
```

첫째, 사용자가 Start Basic Asynchronous Request 버튼을 클릭하면 이벤트가 발생해서 startRequest() 메소드가 실행된다.

둘째, XHR 객체가 생성되고 handleStateChange 콜백함수가 XHR 객체의 onreadystatechange 속성에 저장된다.

셋째, GET 방식의 비동기로 서버에 요청을 보낸는데, 이때 서버의 simpleResponse.xml 파일을 요구한다.

넷째, 서버는 Ajax 클라이언트의 요청 url 인 simpleResponse.xml 을 찾아서 읽은 후에 string 형식으로 XHR 객체로 보낸다.

다섯째, 콜백메소드는 XHR 의 state 가 변할때 실행되므로 readyState=4 이고 stat=200 일때 결과값을 브라우저에 보낸다.



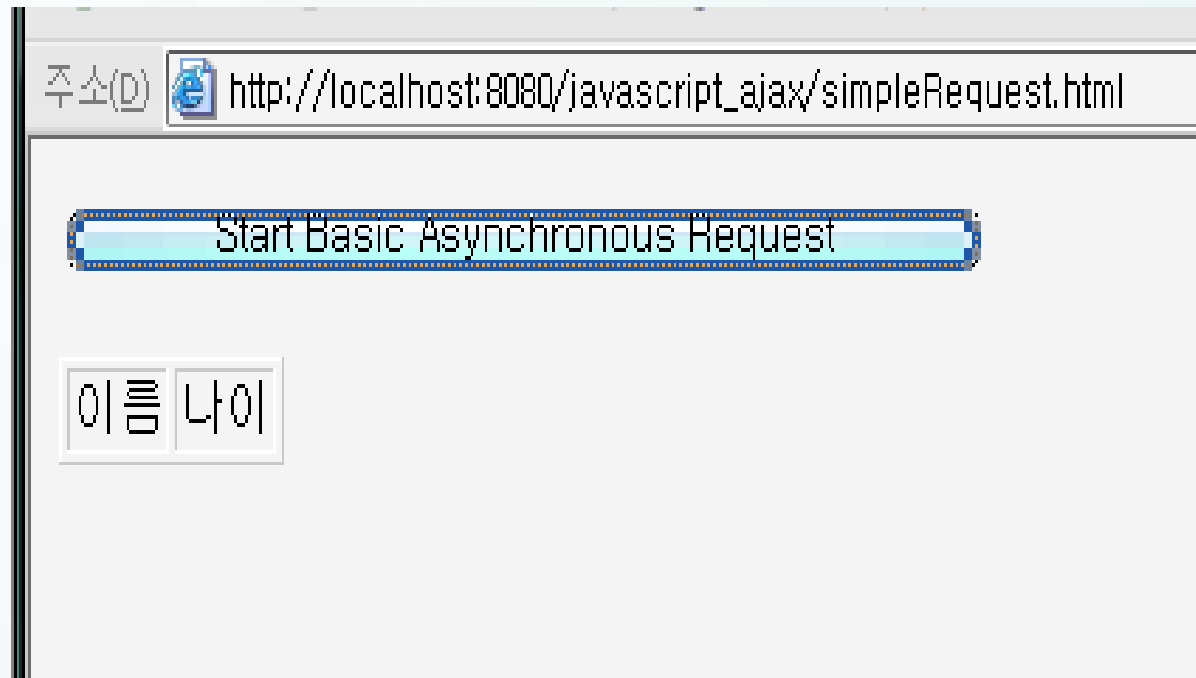
- innerHTML이란?
 - 특정 html 태그 안에 들어갈 html을 말한다.
 - document.getElementById()함수를 이용해서 원하는 html 태그에 구조적으로 접근하여 html 태그를 삽입 또는 추출할 수 있다.
 - 간단하고 단순한 문자열을 처리하는데 적합한 형태이다.

- simpleRequest.html

```
<html xmlns="http://www.w3.org/1999/xhtml">
<script type="text/javascript" src="createXMLHttpRequest.js"></script>
<script type="text/javascript">
var xmlHttp;
function startRequest() {
    xmlHttp = createXMLHttpRequest();
    xmlHttp.onreadystatechange = handleStateChange;
    xmlHttp.open("GET", "simpleResponse.html", true);
    xmlHttp.send(null);
}
function handleStateChange() {
    if(xmlHttp.readyState == 4) {
        if(xmlHttp.status == 200) {
            //alert("The server replied with: " + xmlHttp.responseText);
            document.getElementById("results").innerHTML = xmlHttp.responseText;
        }
    }
}
</script>
<body>
    <form action="#">
        <input type="button" value="Start Basic Asynchronous Request" onclick="startRequest();"/>
    </form>
    <div id="results"></div>
</body>
</html>
```

- simpleResponse.xml

```
<table border="1">  
<tr>  
<td>이름</td>  
<td>나이</td>  
</tr>  
</table>
```





JQuery



- 개요
 - 2006년 초, John Resig가 만든 JavaScript Library
 - JavaScript™와 Asynchronous JavaScript + XML (Ajax) 프로그래밍을 단순화 함
 - DOM 스크립팅과 Ajax의 반복성을 단순하게
 - 코드를 단순하고 간결하게 유지한다.
 - 많은 반복 루프와 DOM 스크립팅 라이브러리 호출을 작성할 필요가 없다.

- JavaScript Library 종류



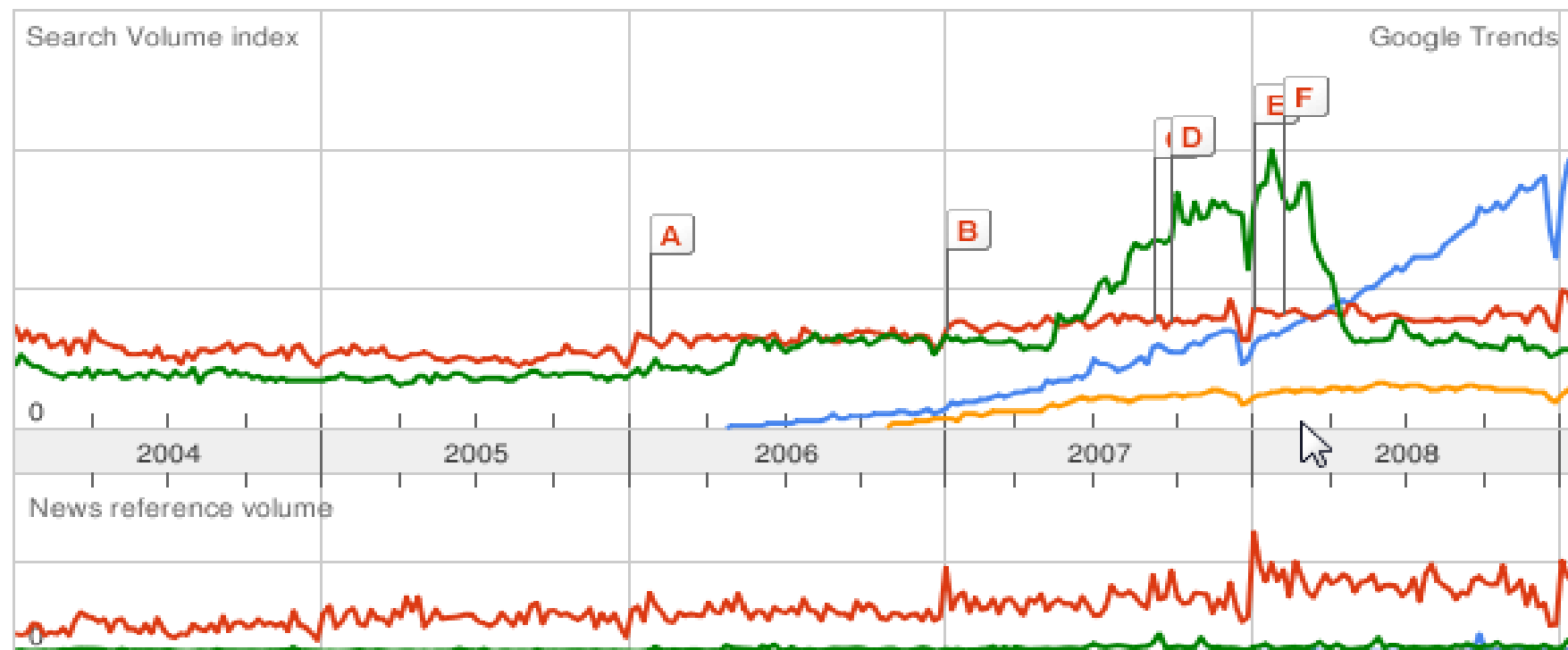
jquery, prototype, mootools, dojo

Search Trends

Tip: Use commas to compare multiple search terms.

Searches Websites

● jquery ● prototype ● mootools ● dojo



- 다운로드

HTTP://JQUERY.COM

jQuery: The Write Less, Do More, JavaScript Library - Microsoft Internet Explorer

파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)

주소(D) <http://jquery.com/>

jQuery Plugins UI Blog About Donate

jQuery
write less, do more.

Download Documentation Tutorials Bug Tracker Discussion

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.**

✓ [Lightweight Footprint](#) ✓ [CSS3 Compliant](#) ✓ [Cross-browser](#)

GRAB THE LATEST VERSION!

CHOOSE YOUR COMPRESSION LEVEL:

☒ PRODUCTION (**19KB**, Minified and Gzipped)

☐ DEVELOPMENT (**120KB**, Uncompressed Code)

[Download \(jQuery\);](#)

Current Release: **v.1.3.2**

WHO'S USING JQUERY? Google DELL IBM NBC CBS NETFLIX Technorati mozilla.org

LEARN JQUERY NOW!

What does jQuery code look like? Here's the quick and dirty:

```
$("#p.neat").addClass("obnoxious").show("slow");
```

[RUN CODE](#)

JQUERY RESOURCES

Getting Started With jQuery

- ◆ [How jQuery Works](#)
- ◆ [Tutorials](#)
- ◆ [Using jQuery with other libraries](#)
- ◆ [jQuery Documentation](#)

Developer Resources

- ◆ [Mailing List](#)
- ◆ [Source code / SVN](#)
- ◆ [Plugin Authoring](#)
- ◆ [Submit a New Bug Report](#)

완료 인터넷



- 특징

CSS Selector

도큐먼트의 구조를 명료하면서도 읽기 쉬운 형태로 표현 가능

Plug-in Architecture

feature creep를 피하고 창의적 산출물을 공유

Method Chaining

여러 동작을 한 줄에 나열이 가능하며, 임시 변수 사용을 최소화 하거나 불필요한 반복을 피할 수 있음