Ajax

1.Ajax란

1-1.Ajax 개념(Asynchronous Javascript + XML)

Ajax라는 용어는 Jesse James Garret 이 2005년 2월 'Ajax :A New Approach to Web Application란 에세이에서 처음으로 사용하였다.

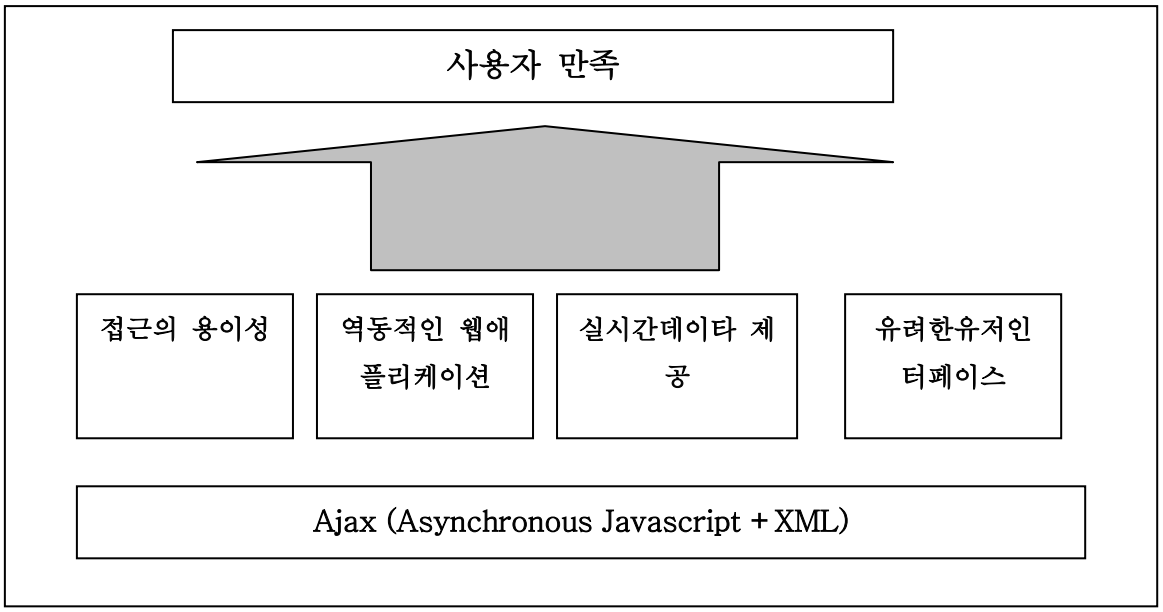
Jesse James Garret은 에세이에서 Google 사례를 제시하면서 웹애플리케이션의 새로운 접근을 소개하였다. 바로이단어하나가 새로운 웹애플리케이션의 패러다임으로 자리 잡아가고있다

Ajax가 혜성처럼 등장하긴 했지만 사실 이와 관련된기술들은 90년대 중반부터 존재했던 것으로 ,자바스크립트와 DOM (Document Object Model) 을 기반으로 한다.

DOM은 CSS,XHTML,XML 과 같은 일종의 웹표준 이며 이러한것들은 모두 Ajax 라는 말이 나오기 수년전부터 존재해왔던 기술이다.

새로운 점이라면 개념적으로 하나로묶어 개발방법으로 제시되었다는 점이다

1-2. Ajax 방식의 특징



① 접근의 용이성

Ajax 의 중심기술은 JavaScript,XHTML,CSS,DOM,XML,XMLHttpRequest 이다.

JavaScript는 웹애플리케이션을 개발해보았던 개발자는 누구나 알고있는 언어이다.

XML은 구조적인 개념만 가지고있으면 구사할수있다. 왜냐하면 XML은 애플리케이션

로직을 구현하는 것이 아니라 구조적으로 데이터를 만드는 것이 주된목적이기 때문이다

XHTML,CSS는 HTML 계열이며 HTML을 모르는 개발자는 없다고해도 과언이아니다

DOM은 DHTML의 비표준적인면을 보완하면서 유저인터페이스를 제공한다.

XMLHttpRequest이다. 이것또한 쉽게 접근할수있다.

② 역동적인 웹애플리케이션

사용자가 웹페이지에서 전송버튼을 클릭한후,서버가 처리를 완료할때까지 멀끔히 웹페이지를 쳐다보고 기다려야하는 모습에서 벗어날수있다.Ajax는 웹페이지전체가 서버에서 수행한후 결과를 브라우져에 보내는것이아니라 데이터 처리만 서버에서 수행하고 사용자와 직접적인 관계가있는 유저인터페이스는 클라이언트에서 수행한다.

서버에서 웹페이지를 해석하고 HTML을 만들어 브라우져에 보내는 일련의 과정을 수행하지 않아도 된다.따라서 웹애플리케이션 처리속도가 향상되는 것은 당연하며,클라이언트에서 수행하게되므로 역동적인 애플리케이션을 실현할수있다.

③ 실시간데이타제공

Ajax는 비동기 방법으로 데이터를 처리하는 것이 기본적인방법이다.

동기방법으로 처리할수도있지만 특별한경우를 제외하고는 비동기방법으로 처리한다.

서버가 데이터를 처리하기위해서는 웹페이지를 해석하여 데이터를 추출해야한다.

하지만 Ajax는 처음부터 데이터만 서버로 보내고 가져온다.

따라서 서버에서 데이터를 추출할필요가 없으므로 그만큼 데이터 처리 속도가 향상된다.

④수려한유저인터페이스

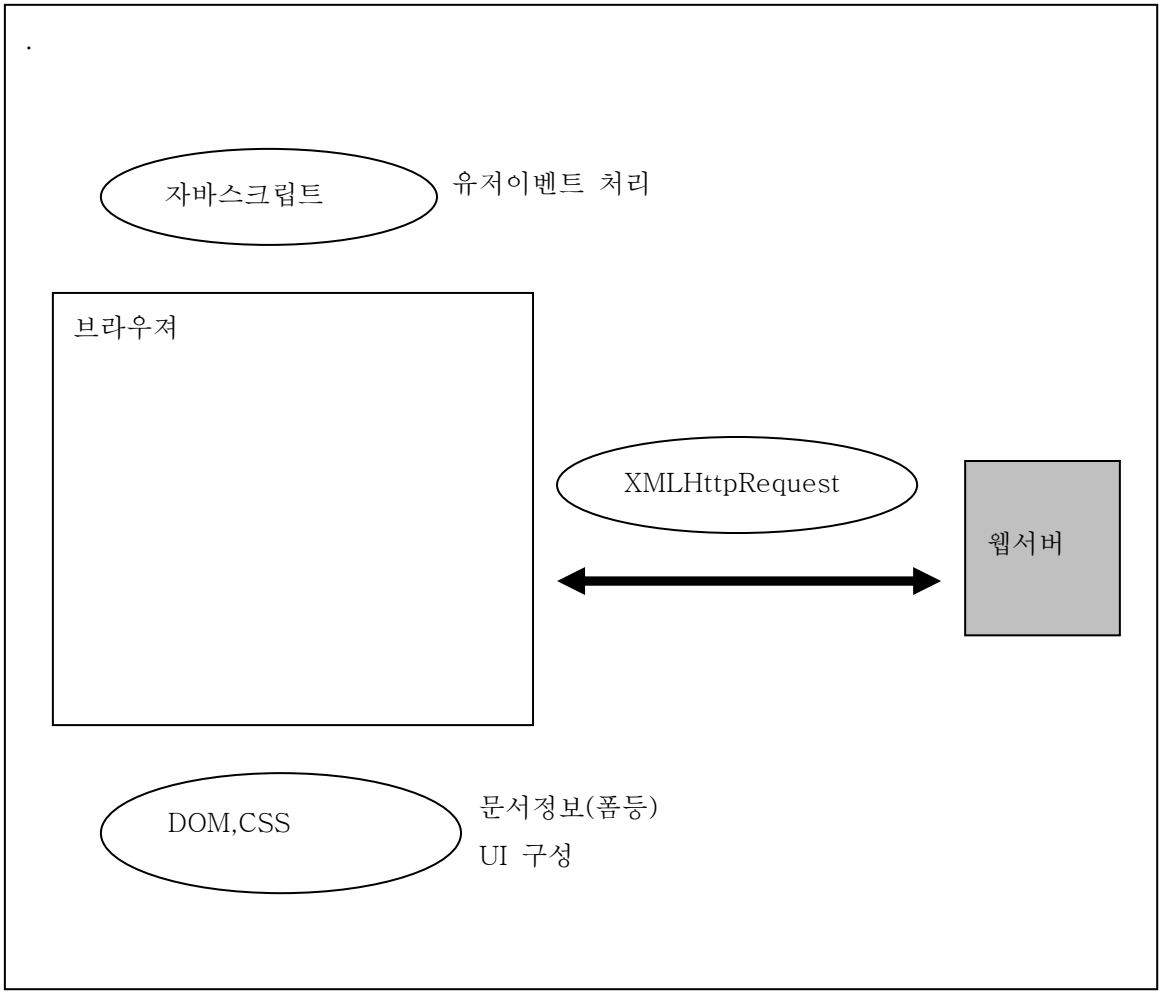
Ajax는 깨끗하도 편리한 유저인터페이스를 제공할수있다.둔탁한 웹페이지를 미려한 웹페이지로 만들수있다. 데스크탑 애플리케이션 에서 구현할 수 있는 유저 인터페이스를 제공할수있으며 웹이갖는 다양한 유저인터페이스를 제공할수있다.

즉 데스크탑 애플리케이션의 유저 인터페이스+ 웹의 유저인터페이스를 구현할 수 있다 이는 Ajax가 데스크탑에서 실행 되기 때문에 가능하다.

1-3 Ajax 방식의 싸이트

http://map.google.com

2.Ajax의 주요구성요소



Ajax 를 구성하는 요소들은 각자 맡은 역할이 존재하는데 그역할은 다음과 같다.

- XMLHttpRequest – 웹서버와의 통신을 담당한다.사용자의요청을 웹서버에 전송하고 웹서버로부터 받은 결과를 웹브라우져에 전달한다.
- DOM – 문서의 구조를 나타낸다. 폼등의 정보나 화면구성을 조작할 때 사용된다.
- CSS – 글자색 ,배경색 위치 ,투명도 등 UI 와 관련된 부분을 담당한다.
- 자바스크립트 – 사용자가 마우스를 드래그하거나 버튼을 클릭하면 ,XMLHttpRequest를 사용해서 웹서버에 요청을 전송한다.
  또한 , XMLHttpRequest객체로부터 응답이 오면 DOM,CSS 등을 사용해서 화면을 조작한다.

3.XMLHttpRequest 객체

3-1.XMLHttpRequest 객체를 사용한 데이터송수신

① XMLHttpRequest 객체 구하기

XMLHttpRequest 객체를 구하는방식은 IE 와 나머지 브라우저가 서로다르다. IE 같은 경우는 ActiveX 컴포넌트로 제공하고있으며 기타 나머지 부라우져 의경우는 XMLHttpRequest클래스를 기본적으로 제공하고있다

```
httpRequest= new ActiveXObject("Msxml2.XMLHTTP");
httpRequest.open("GET", url, true);
httpRequest.send(null);
```

②웹서버에 요청전송하기

XMLHttpRequest 객체를 생성한 다음에는 ,생성한 XMLHttpRequest객체를 사용해서 웹서버에 요청을 전송할 수 있다. 웹서버에 요청을 전송할 때 사용되는 함수는 다음과 같다.

- **open() 함수 : 요청의 초기화,GET/POST선택,접속할 URL입력**

  open(*arg1,arg2,arg3*)

  arg1 : HTTP 메서드를 지정한다.(GET or POST)

  arg1 : 접속할 URL을 입력한다.

  arg3 : 동기/비동기 방식을 지정한다.

- **send() 함수 : 웹서버에 요청 전송**

```
httpRequest= new ActiveXObject("Msxml2.XMLHTTP");
httpRequest.open("GET", "helloAjax.jsp", true);
httpRequest.send(null);
```

■ GET 방식전달

```
httpRequest= new ActiveXObject("Msxml2.XMLHTTP");
httpRequest.open("GET",
                 "helloAjax.jsp?id=guard&pw=1234",
                 true);
httpRequest.send(null);
```

■ POST 방식전달

```
httpRequest= new ActiveXObject("Msxml2.XMLHTTP");
httpRequest.open("POST",
                 "helloAjax.jsp",
                 true);
httpRequest.send("id=guard&pw=1234");
```

③ 서버응답처리하기

open함수와 send 함수를 사용하여 웹서버 에 데이터를 전송한 다음에 할작업은 서버로부터 응답이 도착하면 알맞게 처리하는것이다.

ⓐ. XMLHttpRequest 객체는 웹서버로부터 응답이 도착하면 특정자바스크립트를 호출하는 기능이 있는데 이때사용되는 프로퍼티가 onreadystatechange 이다.

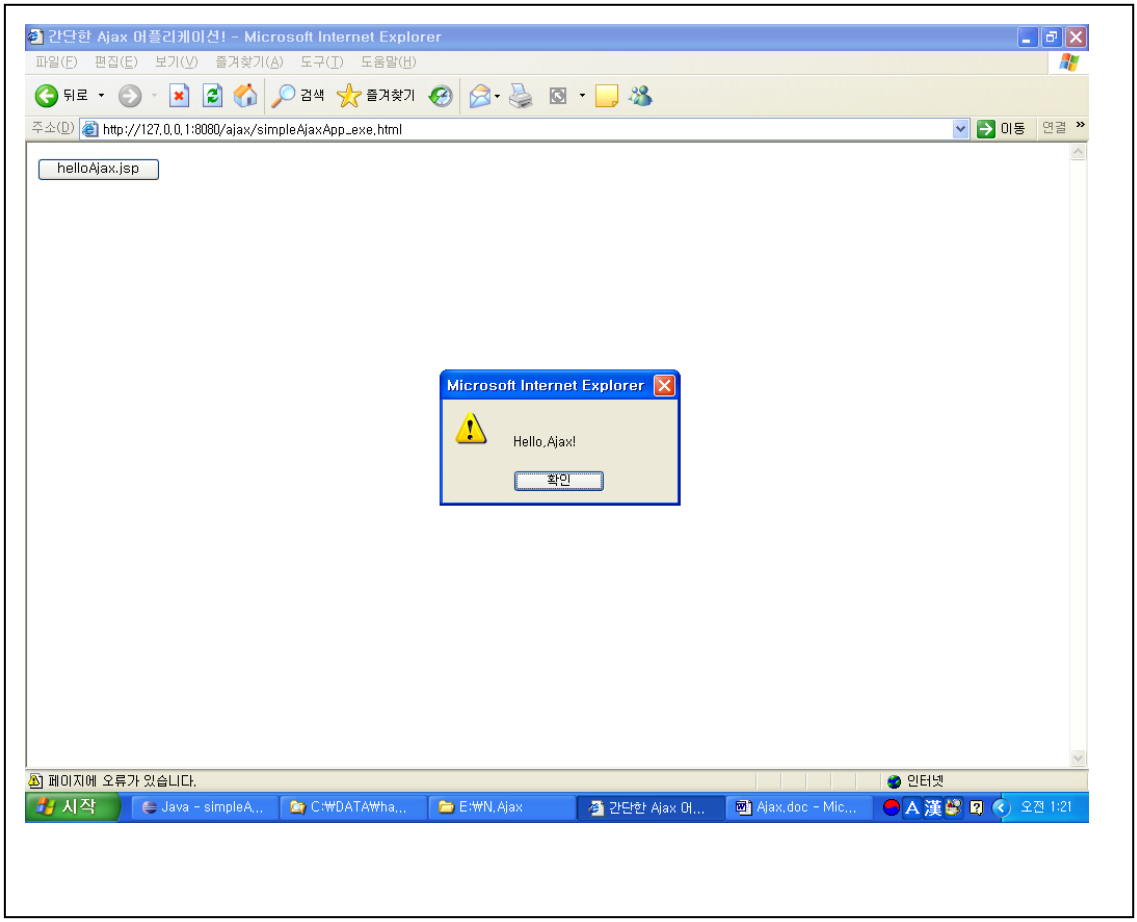
XMLHttpRequest 객체의 onreadystatechange 프로퍼티는 웹서버로부터 응답이 도착할 때 호출될 함수를 지정할 때 사용되는데,코드는 다음과같다.

```
httpRequest= new ActiveXObject("Msxml2.XMLHTTP");
httpRequest.open("POST",
                 "helloAjax.jsp",
                 true);
httpRequest.onreadystatechange=viewMessage;
httpRequest.send("id=guard&pw=1234");
.
.
.

function viewMessage(){
      //서버로부터 응답이 왔을 때 실행되는 메쏘드
}
```

helloAjax.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ko"
lang="ko">
<head>
<meta http-equiv="content-type" content="text/html;
charset=euc-kr" />
<title>간단한 Ajax 어플리케이션!</title>
<script type="text/javascript">
      var httpRequest = null;
      function process(url) {
            httpRequest= new
ActiveXObject("Msxml2.XMLHTTP");
            httpRequest.open("GET", url, true);
            httpRequest.onreadystatechange = viewMessage;
            httpRequest.send(null);
      }
      function viewMessage() {
            alert(httpRequest.responseText);
      }
</script>
</head>
<body>
<input type="button" value="helloAjax.jsp"
onclick="process('helloAjax_ok.jsp')" />
</body>
</html>

</body>
</html>
```

실행 결과


간단한 Ajax 어플리케이션! - Microsoft Internet Explorer
파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)
뒤로 검색 즐겨찾기
주소(D) http://127.0.0.1:8080/ajax/simpleAjaxApp_exe.html
이동 연결
helloAjax.jsp
Microsoft Internet Explorer
Hello,Ajax!
확인
페이지에 오류가 있습니다.
인터넷
시작
Java - simpleA...
C:₩DATA₩ha...
E:₩N.Ajax
간단한 Ajax 어...
Ajax.doc - Mic...
오전 1:21

ⓑ XMLHttpRequest 객체의 상태: readyState

XMLHttpRequest객체의 onreadystatechange 프로퍼티에 콜백함수를지정하면웹서버로부터 응답이 도착했을 때 콜백함수가 호출된다.그런데 실제로 onreadystatechange 에서 명시한 콜백함수는 readyState 라는 프로퍼티의 값이 변경될때마다 호출된다.

readyState 프로퍼티의 값은 다섯개가 존재한다.

보통 readyState 값이 4인경우만 원하는 기능을 수행한다.

값	의미	설명
0	UNINITIALIZED	객체만 생성되고 아직 초기화 되지않은상태 (open 메써드 호출안됨)
1	LOADING	Open 메써드가 호출되고 아직 send 메써드가 부리지 않은상태
2	LOADED	send 메써드는 불리었지만 status 와 헤더는 도착하지 않은 상태
3	INTERACTIVE	데이터의 일부를 받은상태
4	COMPLETE	데이터의 전부를 받은상태,완전한 데이터의 이용가능

```
httpRequest= new ActiveXObject("Msxml2.XMLHTTP");
httpRequest.open("POST",
                 "helloAjax.jsp",
                 true);
httpRequest.onreadystatechange=viewMessage;
httpRequest.send("id=guard&pw=1234");
..

function viewMessage(){
        //서버로부터 응답이 왔을 때 실행되는 메쏘드
        if(httpRequest.readyState==1 ||
          httpRequest.readyState==2 ||
          httpRequest.readyState==3){
          //화면에 서버에서 작업중이라는 메시지 출력
          }else if(httpRequest.readyState==4){
          //서버에서 완전한 응답이 도착한경우
          //응답결과에 따라 알맞은 작업수행
          }
}
```

helloAjax.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ko"
lang="ko">
<head>
<meta http-equiv="content-type" content="text/html;
charset=euc-kr" />
<title>간단한 Ajax 어플리케이션!</title>
<script type="text/javascript">
      var httpRequest = null;
      function process(url) {
            httpRequest= new ActiveXObject("Msxml2.XMLHTTP");
            httpRequest.open("GET", url, true);
            httpRequest.onreadystatechange = viewMessage;
            httpRequest.send(null);
      }
      function viewMessage() {
            //서버로부터 응답이 왔을 때 실행되는 메쏘드
        if(httpRequest.readyState==1 ||
            httpRequest.readyState==2 ||
            httpRequest.readyState==3){
                  //화면에 서버에서 작업중이라는 메시지 출력
            alert("작업중….");
            httpRequest.readySate="+httpRequest.readyState);
            }else if(httpRequest.readyState==4){
                  //서버에서 완전한 응답이 도착한경우
                  //응답결과에 따라 알맞은 작업수행
            alert("정상작업수행...
            httpRequest.readySate="+httpRequest.readyState);
            }
      }
</script></head><body>
<input type="button" value="helloAjax.jsp"
      onclick="process('helloAjax_ok.jsp')" />
</body>
</html>
```

ⓒ서버로부터 응답상태: status,statusText

XMLHttpRequest 객체는 웹서버가 전달한 상태코드를 status 프로퍼티에 저장한다.status 상태코드에 저장되는 주요 상태코드는 다음과 같다.

값	택스트	설명
200	OK	요청성공
403	Frobidden	접근거부
404	Not Found	페이지없음
500	Internal Server Error	서버오류발생

따라서 서버로부터 응답이 도착하면 status 프로퍼티를 사용해서 요청이 성공적으로 수행되었는지 확인해야한다.

```
httpRequest= new ActiveXObject("Msxml2.XMLHTTP");
httpRequest.open("POST",
                 "helloAjax.jsp",
                 true);
httpRequest.onreadystatechange=viewMessage;
httpRequest.send("id=guard&pw=1234");

function viewMessage(){
        //서버로부터 응답이 왔을 때 실행되는 메쏘드
        if(httpRequest.readyState==1 ||
          httpRequest.readyState==2 ||
          httpRequest.readyState==3){
                //화면에 서버에서 작업중이라는 메시지 출력
          }else if(httpRequest.readyState==4){
                if(httpRequest.status==200){
                  //정상적으로 수행
                }else{
                  alert("문제발생:"+ httpRequest.status);
                }
          }
}
```

helloAjax.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ko"
lang="ko"><head>
<meta http-equiv="content-type" content="text/html;
charset=euc-kr" />
<title>간단한 Ajax 어플리케이션!</title>
<script type="text/javascript">
       var httpRequest = null;
       function process(url) {
              httpRequest= new ActiveXObject("Msxml2.XMLHTTP");
              httpRequest.open("GET", url, true);
              httpRequest.onreadystatechange = viewMessage;
              httpRequest.send(null);
       }
       function viewMessage() {
        if(httpRequest.readyState==1 ||
        httpRequest.readyState==2 ||
               httpRequest.readyState==3){
                     //화면에 서버에서 작업중이라는 메시지 출력
                     alert("작업중….
httpRequest.readySate="+httpRequest.readyState);
              }else if(httpRequest.readyState==4){
                     if(httpRequest.status==200){
                            //정상적으로 수행
                            alert("정상수행...
httpRequest.readySate="+httpRequest.readyState);
                     }else{
                            alert("문제발생:"+httpRequest.status);
                     }
              }
        }
</script>
</head>
<body><input type="button" value="helloAjax.jsp"
       onclick="process('helloAjax.jsp')" />
</body>
</html>
```

ⓓ서버로부터 받은 응답데이타 사용하기: responseText

서버로부터응답이 도착한 것을 확인하고 readyState 프로퍼티의 값이 4이고 서버가 요청을 올바르게 수행했다면 status 프로퍼티값이 200 이고 이제 남은 작업은 서버가 전송한 데이터를 사용하는것이다.

웹서버는 단순텍스트 또는 XML 의 두가지 형식으로 데이터를 전송할수 있다

웹서버가 생성한 단순 응답테스트를 참조하려면 XMLHttpRequest객체의 resoponseText 프로퍼티를 사용하면 된다.

```
httpRequest= new ActiveXObject("Msxml2.XMLHTTP");
httpRequest.open("POST",
                 "helloAjax.jsp",
                 true);
httpRequest.onreadystatechange=viewMessage;
httpRequest.send("id=guard&pw=1234");

function viewMessage(){
      //서버로부터 응답이 왔을 때 실행되는 메쏘드
      if(httpRequest.readyState==1 ||
        httpRequest.readyState==2 ||
        httpRequest.readyState==3){
            //화면에 서버에서 작업중이라는 메시지 출력
        }else if(httpRequest.readyState==4){
            if(httpRequest.status==200){
              var txt=httpRequest.responseText;
              //txt를 사용해서 알맞은 작업수행
            }else{
              alert("문제발생:"+ httpRequest.status);
            }
        }
}
```

helloAjax.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ko"
lang="ko">
<head><meta http-equiv="content-type" content="text/html;
charset=euc-kr" />
<title>간단한 Ajax 어플리케이션!</title>
<script type="text/javascript">
       var httpRequest = null;
       function process(url) {
              httpRequest= new ActiveXObject("Msxml2.XMLHTTP");
              httpRequest.open("GET", url, true);
              httpRequest.onreadystatechange = viewMessage;
              httpRequest.send(null);
       }
       function viewMessage() {
        if(httpRequest.readyState==1 ||
        httpRequest.readyState==2 ||
               httpRequest.readyState==3){
                     //화면에 서버에서 작업중이라는 메시지 출력
alert("작업중…. httpRequest.readySate="+httpRequest.readyState);
              }else if(httpRequest.readyState==4){
                     //서버에서 완전한 응답이 도착한경우
                     //응답결과에 따라 알맞은 작업수행
                     if(httpRequest.status==200){
  alert("정상수행. httpRequest.readySate="+httpRequest.readyState);
alert("서버응답텍스트데이타:"+httpRequest.responseText);
                     }else{
                            alert("문제발생:"+httpRequest.status);
                     }
              }
        }
</script></head><body>
<input type="button" value="helloAjax.jsp"
       onclick="process('helloAjax.jsp')" />
<input type="text" id="result"/>
</body>
</html>
```

XMLHTTPRequest모듈만들기

httpRequest.js

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

function sendRequest(url, params, callback, method) {
	httpRequest = getXMLHttpRequest();
	var httpMethod = method ? method : 'GET';
	if (httpMethod != 'GET' && httpMethod != 'POST') {
		httpMethod = 'GET';
	}
	var httpParams = (params == null || params == '') ? null :
params;
	var httpUrl = url;
	if (httpMethod == 'GET' && httpParams != null) {
		httpUrl = httpUrl + "?" + httpParams;
	}
	httpRequest.open(httpMethod, httpUrl, true);
	httpRequest.setRequestHeader(
		'Content-Type', 'application/x-www-form-urlencoded');
	httpRequest.onreadystatechange = callback;
	httpRequest.send(httpMethod == 'POST' ? httpParams : null);
}
```

helloApp.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ko" lang="ko">
<head>
        <meta http-equiv="content-type" content="text/html; charset=euc-kr"
/>
        <title>안녕하세요!</title>
        <script type="text/javascript" src="httpRequest.js"></script>
        <script type="text/javascript">
        function helloToServer() {
                var                                params                                =
"name="+ encodeURIComponent(document.f.name.value);
                sendRequest("hello.jsp", params, helloFromServer, "POST");
        }
        function helloFromServer() {
                if (httpRequest.readyState == 4) {
                        if (httpRequest.status == 200) {
                                alert("서버응답:"+ httpRequest.responseText);
                        }
                }
        }
        </script>
</head>
<body>
<form name="f">
<input type="text" name="name" />
<input type="button" value="입력" onclick="helloToServer()" />
</form>
</body>
</html>
```