

```

1 XMLHttpRequest Object
2
3 1. XHR을 지원하는 주요 Web Browser의 사양
4     1) Microsoft IE 4.0 +
5     2) FF 1.0 +
6     3) Opera 7.6 +
7     4) Safari 1.2 +
8     5) Netsape 7 +
9     6) Konqueror 3 +
10
11
12 2. XHR Programming 순서
13     1)XHR 객체 구하기
14     2)Web Server(WAS)에 요청 전송하기
15     3)Web Server(WAS)에서 응답이 도착하면 화면에 반영하기
16
17
18 3. XHR 객체 구하기
19     var xhr = new XMLHttpRequest();
20
21
22 4. Web Server(WAS)에 요청 전송하기
23     1)open()
24         -요청의 초기화
25         -GET/POST 선택
26         -접속할 URL 입력
27
28     2)send()
29         -Web Server(WAS)에 요청 전송
30
31     3)open()
32         -다음 3개의 인자
33         a. HTTP Method : "GET" | "POST"
34         b. 접속할 URL, 보안상 접속할 URL은 현재 Page와 같은 Domain
35         c. 동기/비동기 방식 지정
36
37
38 5. IE 7 이하용 XMLHttpRequest 객체 생성
39     [httpRequest.js]
40     function getXMLHttpRequest(){
41         if(window.ActiveXObject){ //IE7
42             try{
43                 return new ActiveXObject("Msxml2.XMLHTTP");
44             }catch(e){
45                 try{
46                     return new ActiveXObject("Microsoft.XMLHTTP");
47                 }catch(e){
48                     return null;
49                 }
50             }
51         }else if(window.XMLHttpRequest){ //IE8 higher, FF, Safari, Opear, Chrome
52             return new XMLHttpRequest();
53         } else{
54             return null;
55         }
56     }
57     var httpRequest = null;
58     function sendRequest(url, params, callback, method){
59         httpRequest = getXMLHttpRequest();
60         var httpMethod = method ? method : "get";
61         if(httpMethod.toLowerCase() != "get" && httpMethod.toLowerCase() != "post"){
62             httpMethod = "get";
63         }
64         var httpParams = (params == null || params == "") ? null : params;
65         var httpUrl = url;
66         if(httpMethod == "get" || httpParams != null){
67             httpUrl = httpUrl + "?" + httpParams;

```

```

68     }
69     httpRequest.open(httpMethod, httpUrl, true);
70     httpRequest.onreadystatechange = callback;
71     httpRequest.send(httpMethod == "post" ? httpParams : null);
72 }
73
74
75 6. Lab.
76 [data.txt]
77 <font color='red' size='7'><b>This is a Server Message</b></font>
78
79 [ajaxdemo.html]
80 <script>
81     function getData(url){
82         xmlObj = new XMLHttpRequest();
83         xmlObj.open("get", url);
84         xmlObj.onreadystatechange = myCallback;
85         xmlObj.send(null);
86     }
87     function myCallback(){
88         if(xmlObj.readyState == 4 && xmlObj.status == 200){
89             document.getElementById("targetDiv").innerHTML = xmlObj.responseText;
90         }
91     }
92 </script>
93 <body>
94     <form>
95         <input type="button" value="Display Message" onclick="getData('data.txt');" />
96     </form><br />
97     <div id="targetDiv">
98         <p>here</p>
99     </div>
100 </body>
101
102 [pizzas.txt]
103 <font size='6' color='blue'><i>Toppings : pepperoni, sausage, black olives.</i></font>
104
105 [sandwiches.txt]
106 <font size='7' color='red'></b>We offer too many sandwiches to list!</b></font>
107
108 [soups.txt]
109 <font color='green' size='5'><u>Soups : chicken, beef, or vegetable.</u></font>
110
111 [ajaxdemo1.html]
112 <script>
113     var xmlObj;
114     function getData(url){
115         xmlObj = new XMLHttpRequest();
116         xmlObj.open("get", url, true);
117         xmlObj.onreadystatechange = OnMouseOver;
118         xmlObj.send(null);
119     }
120     function OnMouseOver(){
121         if(xmlObj.readyState == 4 && xmlObj.status == 200){
122             document.getElementById("targetDiv").innerHTML = xmlObj.responseText;
123         }
124     }
125 </script>
126 </head>
127 <body>
128     <h1>Mouse Over Ajax Demo</h1>
129     
130     
131     <br />
132     <div id="targetDiv">
133         <p>Welcome to my restaurant!</p>
134     </div>

```

```

135 </body>
136
137
138 7. Data 전달방법
139 1)GET 방식 전달
140     var xhr = new XMLHttpRequest();
141     xhr.open("GET", "test.jsp?id=javaexpert&pw=1234", true);
142     xhr.send(null);
143
144 2)POST 방식 전달
145     xhr = new XMLHttpRequest();
146     xhr.open("POST", "test.jsp", true);
147     xhr.setRequestHeader("Content-Type","application/x-www-form-urlencoded;charset=utf-8");
148     xhr.send("id=javaexpert&pw=1234");
149
150 3)JSON 전달
151     xhr = new XMLHttpRequest();
152     xhr.onload = mycallback;
153     var data = {
154         "username" : document.getElementById("username").value,
155         "userage" : document.getElementById("userage").value,
156         "userphone" : document.getElementById("userphone").value
157     };
158     xhr.open("POST", "getData.jsp");
159     //반드시 POST 방식에서는 setRequestHeader()를 설정해야 한다.
160     xhr.setRequestHeader("Content-Type", "application/json");
161     //POST 방식에서는 반드시 send(data)를 넣을 것.
162     xhr.send(JSON.stringify(data));
163
164 [jsonajax.html]
165 <script>
166     window.addEventListener('load', setup);
167     function setup(){
168         document.getElementById("btnGetdata").addEventListener('click', myclick);
169     }
170     var xhr;
171     function myclick(){
172         xhr = new XMLHttpRequest(); //1. 단계
173         xhr.onload = mycallback; //2. 단계
174         var data = {
175             "username" : document.getElementById("username").value,
176             "userage" : document.getElementById("userage").value,
177             "userphone" : document.getElementById("userphone").value
178         };
179         xhr.open("POST", "getData1.jsp");
180         //반드시 POST 방식에서는 setRequestHeader()를 설정해야 한다.
181         xhr.setRequestHeader("Content-Type", "application/json");
182         //POST 방식에서는 반드시 send(data)를 넣을 것.
183         xhr.send(JSON.stringify(data));
184     }
185     function mycallback(){
186         if(xhr.status == 200 || xhr.status == 201){
187             document.getElementById("result").innerHTML = xhr.responseText;
188         }
189     }
190 </script>
191 <body>
192     <h1>User Info : JSON</h1>
193     <ul>
194         <li>Name : <input type="text" id="username" /></li>
195         <li>Age : <input type="number" id="userage" /></li>
196         <li>Phone : <input type="tel" id="userphone" /></li>
197         <li><button type="button" id='btnGetdata'>Get Data</button></li>
198     </ul>
199     Result : <span id="result"></span>
200 </body>
201

```

```

202 -json-simple-1.1.1.jar 필요
203 [getData.jsp]
204 <%@ page language="java" contentType="text/html; charset=UTF-8"
205     pageEncoding="UTF-8"%>
206 <%@ page import="org.json.simple.JSONObject, org.json.simple.parser.JSONParser" %>
207 <%@ page import="java.io.BufferedReader" %>
208 <%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
209
210 <fmt:requestEncoding value="utf-8" />
211 <%
212     BufferedReader reader = request.getReader();
213     String line = null, str = "";
214     while((line = reader.readLine()) != null){
215         str += line + "\n";
216     }
217
218     JSONParser parser = new JSONParser();
219     JSONObject obj = (JSONObject)parser.parse(str);
220     String username = (String)obj.get("username");
221     int userage = Integer.parseInt((String)obj.get("userage"));
222     String userphone = (String)obj.get("userphone");
223 %>
224
225 <span style='color:blue;font-weight:bold;font-size:2em'><%=username %></span>
226 <span style='font-style:italic'><%=userage %></span>
227 <span style='text-decoration:underline'><%=userphone %></span>
228

```

4)FormData 전달

-XMLHttpRequest Level2에서 POST방식의 Data를 Server로 전송하는데 편의를 도모하기 위해 새롭게 FormData API가 추가되었다.

-Page전환없이 Form Data를 제출하고 싶을 때 사용한다.

-ex

```

233 var formData = new FormData();
234 formData.append('username', '한지민');
235 formData.append('userage', '24');
236 formData.append('userphone', '010-1234-5678');
237

```

-FormData 생성자

--FormData([form])

a. FormData 객체를 생성한다.

b. Option으로 <form> 객체를 넣을 수 있는데 현재 <form>을 구성하고 있는 요소들의 name과 value를 자동으로 FormData에 저장한다.

-기존 Form이 있는 경우는

```

244 var formData = new FormData(document.getElementById('Form ID'));

```

--이렇게 선택하면 알아서 Form의 내용들의 formData 객체 안에 들어간다.

-append로 넣을 수만 있는 게 아니라 내용물을 확인할 수도 있다.

```

248 formData.has('item');

```

```

249 formData.has('money');

```

```

250 formData.get('item'); // orange

```

```

251 formData.getAll('item'); // ['orange', 'melon']

```

--has() method는 해당하는 key가 존재하는 지 확인할 수 있다.

--get() method로 직접 가져올 수 있다.

--유의할 점은 get()은 처음 저장한 값 하나만 불러온다.

--위에서 item key에 값을 orange와 melon 두 개를 넣었다면 get만 하면 orange 밖에 뜨지 않는다.

--이 때 getAll() method가 있다.

--해당 key에 매칭되는 값들을 전부 배열로 반환한다.

-반복자 사용

```

260 var keys = formData.keys();

```

```

261 keys.next(); // { done: false, value: 'name' }

```

```

262 keys.next(); // { done: false, value: 'item' }

```

```

263 keys.next(); // { done: false, value: 'item' }

```

```

264 keys.next(); // { done: true, value: undefined }

```

```

265 var values = formData.values();

```

```

266 values.next(); // { done: false, value: 'zerocho' }

```

```

267 values.next(); // { done: false, value: 'orange' }
268 values.next(); // { done: false, value: 'melon' }
269 values.next(); // { done: true, value: undefined }
270 var entries = formData.entries();
271 entries.next(); // { done: false, value: ['name', 'zerocho'] }
272 entries.next(); // { done: false, value: ['item', 'orange'] }
273 entries.next(); // { done: false, value: ['item', 'melon'] }
274 entries.next(); // { done: true, value: undefined }
275 --iterator는 상황에 따라 매우 유용할 수 있다.
276 --FormData의 key나 값, 또는 key와 값 모두를 쉽게 보여준다.
277 --반복문에 사용하면 좋다.
278
279 -다양한 예
280 formData.append('test', ['hi', 'zero']);
281 formData.get('test'); // hi, zero
282 formData.delete('test');
283 formData.get('test'); // null
284 formData.set('item', 'apple');
285 formData.getAll('item'); // ['apple']
286 --formData에 이제 값을 지우려면 delete() method를 사용한다.
287 --append()와 비슷한 set() method도 있다.
288 --set()과 append()의 차이점은 set()도 추가를 해주기는 하지만, 기존 key가 있으면 그 값을 모두 덮어쓴다.
289 --위의 item에 대한 값이 orange와 melon이었는데 apple을 set하는 순간 item 값이 apple 하나로 덮어씌워졌다.
290
291 -FormData에 이미지를 담고 싶으면
292 --formData.append('img', document.getElementById('file input').files[0])
293 --위와 같이 넣어주고 Server로 formData를 전송한다.
294 --file이 여러 개면 반복문으로 append()를 하면 된다.
295 --주의할 점은 여러 개를 append()할 때 항상 key값(위에서는 img)은 같아야 여러 file이 같은 key로 upload된다.
296
297 [com.example.MemberServlet.java]
298 @MultipartConfig //FormData를 Servlet에서 사용하기 위해서 필요
299 @WebServlet("/MemberServlet")
300 public class MemberServlet extends HttpServlet {
301     protected void doPost(HttpServletRequest req, HttpServletResponse res) throws
302         ServletException, IOException {
303         req.setCharacterEncoding("utf-8");
304         res.setContentType("text/html;charset=utf-8");
305         String username = req.getParameter("username");
306         int userage = Integer.parseInt(req.getParameter("userage"));
307         String userphone = req.getParameter("userphone");
308         String str = "<span style='color:blue;font-weight:bold;font-size:1.5em'>" + username +
309             "</span>" +
310             "<span style='font-style:italic'>" + userage + "</span>" +
311             "<span style='text-decoration:underline'>" + userphone + "</span>";
312         PrintWriter out = res.getWriter();
313         out.println(str);
314     }
315 }
316
317 [formDataDemo.html]
318 <script>
319     window.onload = function(){
320         document.getElementById("getdata").addEventListener("click", myclick);
321     }
322     function myclick(){
323         var xhr = new XMLHttpRequest();
324         var formData = new FormData(document.myform);
325         /* var formData = new FormData();
326         formData.append("username", document.getElementById("username"));
327         formData.append("userage", document.getElementById("userage"));
328         formData.append("userphone", document.getElementById("userphone")); */
329         xhr.onload = function(){
330             if(xhr.status == 200 || xhr.status == 201)
331                 document.getElementById('result').innerHTML = xhr.responseText;
332         }
333         xhr.open("POST", "MemberServlet");

```

```

332         xhr.send(formData);
333     }
334 </script>
335 </head>
336 <body>
337     <form name="myform">
338         <ul>
339             <li>User name : <input type="text" name="username" id="username"/> </li>
340             <li>User age : <input type="text" name="userage" id="userage"/> </li>
341             <li>User phone : <input type="text" name="userphone" id="userphone"/> </li>
342             <li><button type='button' id='getdata'>Get Data</button>
343         </ul>
344     </form>
345     <div id='result'></div>
346 </body>

```

349 8. Server 응답 처리하기

350 1)onreadystatechange

351 -Web Server(WAS)로 부터 응답이 도착하면 특정한 JavaScript 함수를 호출하는 기능의 property

352 2)ex

353 xhr.onreadystatechange = callbackFunction;

356 9. readyState property

357 1)0

358 -UNINITIALIZED

359 -객체만 생성되고 아직 초기화되지 않는 상태(open () 가 호출되지 않음)

361 2)1

362 -LOADING

363 -open() 호출되고 아직 send() 호출 전

365 3)2

366 -LOADED

367 -send() 호출 후, Server로부터 status 와 Header가 도착하지 않은 상태

369 4)3

370 -INTERACTIVE

371 -Data의 일부를 받은 상태

373 5)4

374 -COMPLETED

375 -Data의 전부를 받은 상태.

376 -완전한 Data의 이용 가능

379 10. Server로부터의 응답 상태

380 -status / statusText

381 1)Web Server로부터 응답이 도착하면 Web Server에서 처리가 올바르게 수행되었는지 확인해야 한다.

382 2)Web Server가 올바르게 작업을 실행했는지, 처리 도중에 Error가 발생했는지 여부 등의 상태 Code

383 3)HTTP 주요 상태 Code

384 -200

385 --OK

386 --요청 성공

387 -403

388 --Forbidden

389 --접근 거부

390 -404

391 --Not Found

392 --Page Not Found

393 -500

394 -Internal Server Error

395 -Server 오류 발생

397 function callbackFunction(){

398 if(httpRequest.readyState == 4) {

```

399         if(httpRequest.status == 200) {
400             // 정상적으로 수행됨
401         } else {
402             alert("문제 발생 : " + httpRequest.status);
403         }
404     }
405 }

```

4)statusText

-status property의 값을 설명하는 문장

```

409
410
411 11. 응답 Data 사용하기
412 -responseText
413 1)Server는 단순 Text 또는 XML 두 가지 형식으로 Data를 전송하는데, 단순 Text를 받을 때 사용

```

```

414
415     function callbackFunction(){
416         if(httpRequest.readyState == 4) {
417             if(httpRequest.status == 200) {
418                 var txt = httpRequest.responseText;
419             } else {
420                 alert("문제 발생 : " + httpRequest.status);
421             }
422         }
423     }

```

```

424
425
426 12. 동기 방식과 비동기 방식의 실행 차이
427 1)XMLHttpRequest 객체의 open()의 마지막 세번째 인자는 동기 / 비동기 여부를 표시
428 2>true : 비동기, false : 동기

```

```

429
430     httpRequest.open("get", "/text.txt", true);
431
432 3)비동기 방식으로 실행한 경우, send() 호출된 뒤 바로 다음의 Code가 실행된다.
433 4)반면, 동기 방식은 send() 호출된 뒤, Server와의 통신이 완전히 완료된 다음, send() 이후의 함수를 Code가 실행된다.

```

```

434
435
436 13. Parameter의 한글 처리 방법
437 1)JavaScript 는 문자열을 utf-8로 encoding해주는 encodeURIComponent() 함수를 제공한다.

```

```

438
439     var param = "name=" + encodeURIComponent('조성모');
440     httpRequest.open("get", "hello.jsp?" + param, true);
441
442 [idcheck.jsp]
443 <%@ page language="java" contentType="text/html; charset=UTF-8"
444 pageEncoding="UTF-8"%>
445 <%
446 request.setCharacterEncoding("utf-8");
447 String irum = request.getParameter("irum");
448 irum = new String(irum.getBytes("ISO8859_1"), "utf-8");
449 out.println("당신의 이름은 " + irum + " 이군요.");
450 %>
451
452 [ajaxdemo3.html]
453 <!DOCTYPE html>
454 <html lang="ko">
455 <head>
456 <meta charset="UTF-8">
457 <title>Insert title here</title>
458 <script type="text/javascript" src="js/httpRequest.js"></script>
459 <script type="text/javascript">
460     window.onload=setup;
461     function setup(){
462         document.getElementById("btn").onclick=Click;
463     }
464     function Click(){
465         var aaa = encodeURIComponent(document.getElementById("irum").value);

```

[illegible]

483 14. XML 문서 처리하기

```

484 -responseXML
485 [test.xml]
486 <?xml version="1.0" encoding="UTF-8"?>
487 <root>XML 데이터입니다</root>
488
489 [ajaxdemo4.html]
490 <!DOCTYPE html>
491 <html lang="ko">
492 <head>
493 <meta charset="UTF-8">
494 <title>Insert title here</title>
495 <script type="text/javascript" src="js/httpRequest.js"></script>
496 <script type="text/javascript">
497     window.onload=setup;
498     function setup(){
499         document.getElementById("btn").onclick=Click;
500     }
501     function Click(){
502         sendRequest('test.xml', "", MyCallback, "get");
503     }
504     function MyCallback(){
505         if(httpRequest.readyState == 4 && httpRequest.status == 200){
506             var xmlDoc = httpRequest.responseXML;
507             var node = xmlDoc.getElementsByTagName("root");
508             alert(node[0].firstChild.nodeValue);
509         }
510     }
511 </script>
512 </head>
513 <body>
514     <input type="button" value="Click Me!" id="btn" />
515 </body>
516 </html>

```

519 15. Ajax and Servlet

```

520 [web.xml]
521 <?xml version="1.0" encoding="UTF-8"?>
522 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="
http://java.sun.com/xml/ns/javaee" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID" version="3.0">
523     <display-name>AjaxDemo</display-name>
524     <servlet>
525         <servlet-name>ParameterGet</servlet-name>
526         <servlet-class>kr.co.javaexpert.libs.ParameterGetServlet</servlet-class>
527     </servlet>
528     <servlet>
529         <servlet-name>ParameterPost</servlet-name>
530         <servlet-class>kr.co.javaexpert.libs.ParameterPostServlet</servlet-class>

```



```

531     </servlet>
532
533     <servlet-mapping>
534         <servlet-name>ParameterGet</servlet-name>
535         <url-pattern>/servlets/ParameterGet</url-pattern>
536     </servlet-mapping>
537     <servlet-mapping>
538         <servlet-name>ParameterPost</servlet-name>
539         <url-pattern>/servlets/ParameterPost</url-pattern>
540     </servlet-mapping>
541 </web-app>
542
543 [ParameterGetServlet.java]
544 package kr.co.javaexpert.libs;
545
546 import java.io.*;
547 import javax.servlet.*;
548 import javax.servlet.http.*;
549
550 public class ParameterGetServlet extends HttpServlet{
551     public void doGet(HttpServletRequest request, HttpServletResponse response)
552         throws ServletException, IOException{
553         requestProcess(request, response);
554     }
555
556     public void doPost(HttpServletRequest request, HttpServletResponse response)
557         throws ServletException, IOException{
558         requestProcess(request, response);
559     }
560
561     public void requestProcess(HttpServletRequest request, HttpServletResponse response)
562         throws ServletException, IOException{
563         response.setContentType("text/html;charset=utf-8");
564         PrintWriter out = response.getWriter();
565
566         String name = request.getParameter("name");
567         name = entoutf(name);
568         String age = request.getParameter("age");
569         String mobilePhone = request.getParameter("mobilePhone");
570
571         String str = "<b>" + name + "</b>님의 나이는 " + age + "이고, 휴대폰 번호는 " + mobilePhone +
572             " 입니다.";
573         out.println(str);
574         out.close();
575     }
576     private String entoutf(String en){
577         String utf8 = null;
578         try{
579             utf8 = new String(en.getBytes("ISO8859_1"), "utf-8");
580         }catch(java.io.UnsupportedEncodingException e){}
581         return utf8;
582     }
583 }
584
585 [parameterGet.html]
586 <!doctype html>
587 <html lang="ko">
588     <head>
589         <meta charset="utf-8" />
590         <title>GET방식으로 요청파라미터 보내기</title>
591         <script>
592             var xhrObj;
593             window.onload = function(){
594                 document.getElementById("getSend").addEventListener("click", Click, false);
595             }
596             function Click(evt){
597                 xhrObj = new XMLHttpRequest();

```

```

597         var name = document.getElementById("name").value;
598         var age = document.getElementById("age").value;
599         var mobilePhone = document.getElementById("mobilePhone").value;
600         var msg = "name=" + name + "&age=" + age + "&mobilePhone=" + mobilePhone;
601         var url = "servlets/ParameterGet?" + msg;
602         xhrObj.open("get", url, true);
603         xhrObj.onreadystatechange = resultProcess;
604         xhrObj.send(null);
605     }
606     function resultProcess(){
607         if(xhrObj.readyState == 4 && xhrObj.status == 200){
608             document.getElementById("resultDisplay").innerHTML = xhrObj.responseText;
609         }
610     }
611 </script>
612 </head>
613 <body>
614     <form>
615         <h3>GET방식으로 요청파라미터 보내기</h3>
616         이름: <input id="name" type="text" /><br/>
617         나이: <input id="age" type="text" /><br/>
618         휴대폰: <input id="mobilePhone" type="text" /><br/>
619         <input id="getSend" type="button" value="GET방식으로 요청파라미터 보내기" /><br/>
620     </form>
621     <h3>응답 결과</h3>
622     <div id="resultDisplay"></div>
623 </body>
624 </html>
625
626 [ParameterPost.java]
627 package kr.co.javaexpert.libs;
628
629 import java.io.*;
630 import javax.servlet.*;
631 import javax.servlet.http.*;
632
633 public class ParameterPostServlet extends HttpServlet{
634     public void doGet(HttpServletRequest request, HttpServletResponse response)
635         throws ServletException, IOException{
636         requestProcess(request, response);
637     }
638
639     public void doPost(HttpServletRequest request, HttpServletResponse response)
640         throws ServletException, IOException{
641         requestProcess(request, response);
642     }
643
644     public void requestProcess(HttpServletRequest request, HttpServletResponse response)
645         throws ServletException, IOException{
646         request.setCharacterEncoding("utf-8");
647         response.setContentType("text/html;charset=utf-8");
648         PrintWriter out = response.getWriter();
649
650         String name = request.getParameter("name");
651         String age = request.getParameter("age");
652         String mobilePhone = request.getParameter("mobilePhone");
653         String email = request.getParameter("email");
654
655         String str = "<b>" + name + "</b>님의 나이는 " + age + "입니다. <br/>";
656         str += "휴대폰 번호는 " + mobilePhone + " 이고, 이메일은 " + email + " 입니다.";
657         out.println(str);
658         out.close();
659     }
660 }
661
662 [parameterPost.html]
663 <!doctype html>

```

```

664 <html lang="ko">
665 <head>
666 <meta charset="utf-8" />
667 <title>POST방식으로 요청파라미터 보내기</title>
668 <script>
669     var xhrObj;
670     window.addEventListener("load", setup, false);
671     function setup(evt){
672         document.getElementById("postSend").addEventListener("click", Click, false);
673     }
674     function Click(evt){
675         xhrObj = new XMLHttpRequest();
676         var name = document.getElementById("name").value;
677         var age = document.getElementById("age").value;
678         var mobilePhone = document.getElementById("mobilePhone").value;
679         var email = document.getElementById("email").value;
680
681         var str = "name=" + name + "&age=" + age + "&mobilePhone=" + mobilePhone +
682             "&email=" + email;
683         var url = "servlets/ParameterPost" ;
684         xhrObj.onreadystatechange = resultProcess;
685         xhrObj.open("Post", url, true);
686
687         xhrObj.setRequestHeader("Content-Type","application/x-www-form-urlencoded;charset=
688             utf-8");
689         //POST방식의 경우 반드시 설정
690         xhrObj.send(str);
691     }
692     function resultProcess(){
693         if(xhrObj.readyState == 4 && xhrObj.status == 200){
694             document.getElementById("resultDisplay").innerHTML = xhrObj.responseText;
695         }
696     }
697 </script>
698 </head>
699 <body>
700 <h3>POST방식으로 요청파라미터 보내기</h3>
701 <form>
702     이름: <input id="name" type="text" /><br/>
703     나이: <input id="age" type="text" /><br/>
704     휴대폰: <input id="mobilePhone" type="text" /><br/>
705     이메일: <input id="email" type="text" /><br/>
706     <input id="postSend" type="button" value="POST방식으로 요청파라미터 보내기" /><br/>
707 </form>
708 <h3>응답 결과</h3>
709 <div id="resultDisplay"></div>
710 </body>
711 </html>

```