

## 2 1. Ajax 방식

#### 4 2) 기존 방식

6 -Web Server는 JSP/PHP/ASP등의 Server측 application을 사용해서 사용자의 request를 처리한 뒤, 처리 결과를 HTML로 생성해서 Web Browser에 Response한다.

8

10 -<http://map.google.com>

12 -<http://www.naver.com>

#### 14 -몇가지 특징들

- 19

```
21 <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
```

23 <html lang="en">

25 &lt;meta charset="UTF-8"&gt;

27 &lt;script&gt;

```
29 function setup(){
```

31 }

```
33 if(document.getElementById("num1").value != "" &&
```

```
35 document.forms[0].action="noajax.jsp";
```

```
37 document.forms[0].submit();
```

39 }

41 &lt;/head&gt;

43 

```
<input type="text" id="num2" name="num2" size="10" value="\${param.num2}" /> =
```

/>&nbsp;   

48 &lt;/form&gt;

50 &lt;/html&gt;

51

53  $\leq 0\%$ 

```
int num2 = Integer.parseInt(request.getParameter("num2"));
```

57 %>

58

```
50 <!DOCTYPE html>
```

51 &lt;html&gt;

52 &lt;head&gt;

53 &lt;meta charset="UTF-8"&gt;

54 &lt;title&gt;Insert title here&lt;/title&gt;

55 &lt;script&gt;

[illegible]

## 2. Ajax 의 주요 구성 요소

### 1)XMLHttpRequest

- Web Server와 통신을 담당한다.
- 사용자의 요청을 Web Server(WAS)에 전송하고, Web Server(WAS)로부터 받은 결과를 Web Broser에 전달한다.

## 2)DOM

- 문서의 구조를 나타낸다.
- Form 등의 정보나 화면 구성을 조작할 때 사용된다.

### 3)CSS

- 글자색, 배경색, 위치, 투명도 등 UI와 관련된 부분을 담당한다.

#### 4)JavaScript

- 사용자가 mouse를 drag하거나 button을 click하면, XHR객체를 사용해서 Web Server에 요청을 전송한다.
- 또한 XHR 객체로부터 응답이 오면 DOM, CSS등을 사용해서 화면을 조작한다.

### 5)순서

- a. 사용자가 event를 발생시키면 JavaScript는 DOM을 사용해서 필요한 정보를 구한 뒤, XHR객체를 통해서 Web Server(WAS)에 요청을 전달한다.
- b. Web Server(WAS)는 XHR로부터 요청을 알맞게 처리한 뒤, 그 결과를 XML이나 단순 Text로 생성해서 XHR에 전송한다.
- c. Server로부터 응답이 도착하면 XHR객체는 JavaScript에 도착 사실을 알리게 되고, JavaScript는 응답 data와 DOM을 조작해서 사용자 화면에 반영한다.

### 6) Ajax 가 기존의 방식과 다른 차이점

- Web Browser 가 아닌 XHR객체가 Web Server와 통신한다.
- Web Server의 응답 결과가 HTML이 아닌 XML 또는 단순 Text이다.
- Page 이동없이 결과가 화면에 반영된다.

[date.jsp]

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
```

```
<%@ page import="java.util.Calendar" %>
```

< 0%

```
response.setHeader("cache-control", "no-cache"); //HTTP 1.1
```

```
response.setHeader("cache-control", "no-store"); //FF
```

```
response.setDateHeader("expires", 0);
```

```
response.setHeader("pragma", "no-cache"); //HTTP1.0
```

```
Calendar now = Calendar.getInstance();
```

```
int hour = now.get(Calendar.HOUR_OF_DAY);
```

```

130     int minute = now.get(Calendar.MINUTE);
131     int second = now.get(Calendar.SECOND);
132     out.println("'" + hour + " : " + minute + " : " + second);
133     %>
134
135 [test.html]
136 <!DOCTYPE html>
137 <html>
138 <head>
139     <meta charset="UTF-8" />
140     <title>Insert title here</title>
141     <script>
142         window.addEventListener("load", setup, false);
143         function setup(){
144             document.getElementById('mytext').addEventListener('change', startAjax, false);
145         }
146         function startAjax(){
147             var xmlObj;
148             xmlObj = new XMLHttpRequest();
149             xmlObj.onreadystatechange = function(){
150                 if(xmlObj.readyState == 4){
151                     document.getElementById("mytime").value = xmlObj.responseText;
152                     //alert(xmlObj.responseText);
153                 }
154             }
155             xmlObj.open("get", "date.jsp", true);
156             xmlObj.send(null);
157         }
158     </script>
159 </head>
160 <body>
161     <form>
162         Name : <input type="text" id="mytext" /> <br /> <br />
163         <input type="text" id="mytime" size="30"/>
164     </form>
165 </body>
166 </html>

```