

3. JSP 페이지의 내장 변수

- JSP 페이지의 내장 변수(implicit variable) : JSP 페이지 안에 선언을 하지 않고도 사용할 수 있는 변수

```
<%@page contentType= "text/html; charset=euc-kr" %>
<HTML>
  <HEAD><TITLE>정수를 순서대로</TITLE></HEAD>
  <BODY>
    <H3>정수를 순서대로</H3>
    <%
      String str = request.getParameter( "MAX " );
      int max = Integer.parseInt(str);
      for (int cnt = 1; cnt <= max; cnt++)
        out.println(cnt + "<BR> ");
    %>
  </BODY>
</HTML>
```

내장 변수

내장 변수

[그림 3-15] JSP 페이지의 내장 변수의 예

- request 내장 변수는 서블릿 클래스의 doGet, doPost 메서드의 첫 번째 파라미터와 동일한 역할을 한다.
- out 내장 변수는 서블릿 클래스에서 getWriter 메서드를 호출해서 얻은 PrintWriter 객체와 마찬가지로의 역할을 한다.



3. JSP 페이지의 내장 변수

- JSP 페이지 안에서 내장 변수를 사용할 수 있는 이유는 웹 컨테이너가 JSP 페이지를 서블릿 클래스로 변환할 때 자동으로 내장 변수를 선언하기 때문이다.
- JSP 페이지에서 사용할 수 있는 내장 변수들

변수 이름	제공하는 기능/변수의 역할	변수 타입	다루는 장
request	doGet, doPost 메서드의 첫 번째 파라미터와 동일한 역할	javax.servlet.http.HttpServletRequest	3장
response	doGet, doPost 메서드의 두 번째 파라미터와 동일한 역할	javax.servlet.http.HttpServletResponse	3장
out	웹 브라우저로 HTML 코드를 출력하는 기능	javax.servlet.jsp.JspWriter	3장
application	JSP 페이지가 속하는 웹 애플리케이션에 관련된 기능	javax.servlet.ServletContext	3장, 6장
config	JSP 페이지의 구성 정보를 가져오는 기능	javax.servlet.ServletConfig	6장
pageContext	JSP 페이지 범위 내에서 사용할 수 있는 데이터 저장 기능 등	javax.servlet.jsp.PageContext	7장
session	세션에 관련된 기능	javax.servlet.http.HttpSession	4장
page	JSP 페이지로부터 생성된 서블릿	java.lang.Object	다루지 않음
exception	익셉션 객체	java.lang.Throwable	5장



3. JSP 페이지의 내장 변수

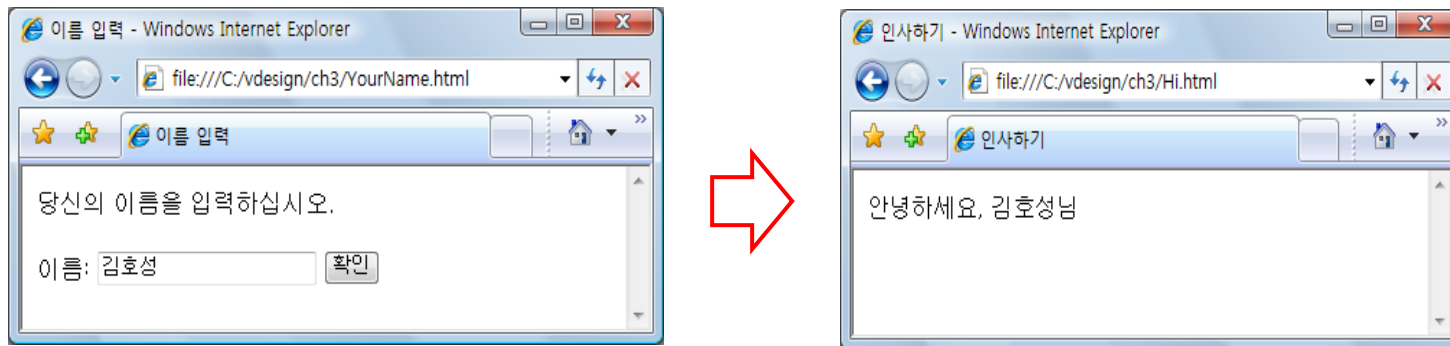
❖ request 내장 변수

- request 내장 변수는 서블릿 클래스에 있는 doGet, doPost 메서드의 첫 번째 파라미터와 동일한 역할을 하고, 타입도 동일하게 javax.servlet.http.HttpServletRequest이다.

```
String str = request.getParameter( "NAME " );
```

↑
데이터 이름

- 웹 브라우저를 통해 입력된 데이터를 가져다가 처리하는 애플리케이션을 작성해보자.



[그림 3-17] 인사말을 출력하는 웹 애플리케이션의 화면 설계

3. JSP 페이지의 내장 변수

❖ request 내장 변수

- 둘 이상의 화면으로 구성된 애플리케이션은 먼저 URL을 정한 뒤 각 URL에 해당하는 코드를 작성하는 것이 좋다.

http://localhost:8080/brain03/YourName.html

← (그림3-17) 왼쪽 화면 URL

http://localhost:8080/brain03/Hi.jsp

← (그림3-17) 오른쪽 화면 URL

[예제3-6] 웹 브라우저로부터 이름을 입력받는 HTML 문서

```
<HTML>
  <HEAD>
    <META http-equiv= "Content-Type" content= "text/html; charset=euc-kr" >
    <TITLE>이름 입력</TITLE>
  </HEAD>
  <BODY>
    당신의 이름을 입력하세요.
    <FORM ACTION=/brain03/Hi.jsp METHOD=GET>
      이름: <INPUT TYPE=TEXT NAME=YOURNAME>
      <INPUT TYPE=SUBMIT VALUE= '확인' >
    </FORM>
  </BODY>
</HTML>
```

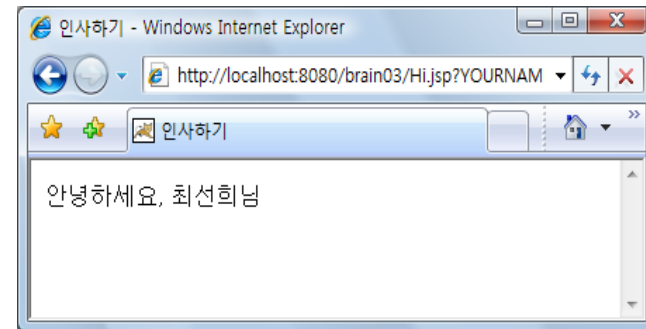
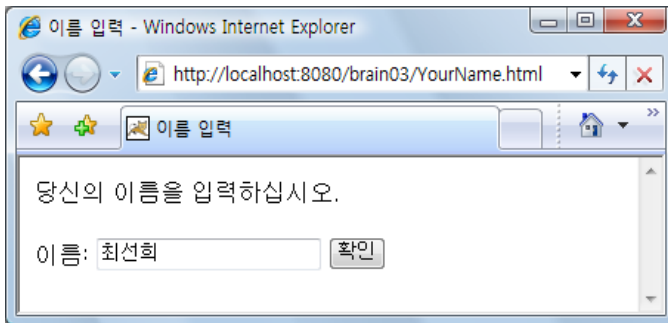


3. JSP 페이지의 내장 변수

❖ request 내장 변수

[예제3-7] 입력된 이름을 가지고 인사말을 출력하는 JSP 페이지

```
<%@page contentType="text/html; charset=euc-kr"%>
<HTML>
  <HEAD><TITLE>인사하기</TITLE></HEAD>
  <BODY>
    안녕하세요, <%= request.getParameter("YOURNAME") %>님
  </BODY>
</HTML>
```

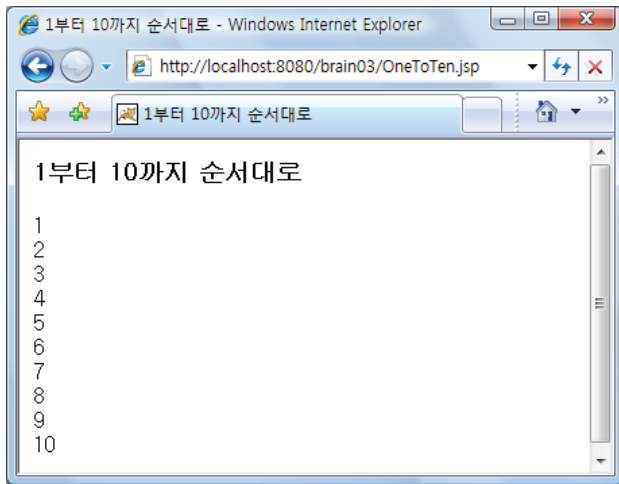


[그림 3-18] 예제 3-6, 예제 3-7의 실행 결과

3. JSP 페이지의 내장 변수

❖ out 내장 변수

- JSP 페이지에서는 HTML 코드와 익스프레션만 가지고도 원하는 HTML 문서를 만들어서 출력할 수 있기 때문에, 서블릿 클래스의 경우처럼 `println`, `print`, `printf` 메서드를 굳이 호출해야 할 필요가 없다.



[그림 3-19] 1부터 10까지의 정수를 순서대로 출력하는 웹 페이지

[예제3-7] 입력된 이름을 가지고 인사말을 출력하는 JSP 페이지

```
<%@page contentType= "text/html; charset=euc-kr"%>

<HTML>

    <HEAD><TITLE>1부터 10까지 순서대로</TITLE></HEAD>

    <BODY>

        <H3>1부터 10까지 순서대로</H3>

        <% for (int cnt = 1; cnt <= 10; cnt++) { %>

            <%= cnt %> <BR>

        <% } %>

    </BODY>

</HTML>
```



3. JSP 페이지의 내장 변수

❖ out 내장 변수

- out 내장 변수는 서블릿 클래스에서 `getWriter` 메서드를 호출해서 얻은 `PrintWriter` 객체와 비슷한 역할을 한다.

```
out.print("<FONT SIZE=1>안녕하세요!</FONT> ");
```

```
out.println("<BR> ");
```

[예제3-8] out 내장 변수를 사용하는 JSP 페이지

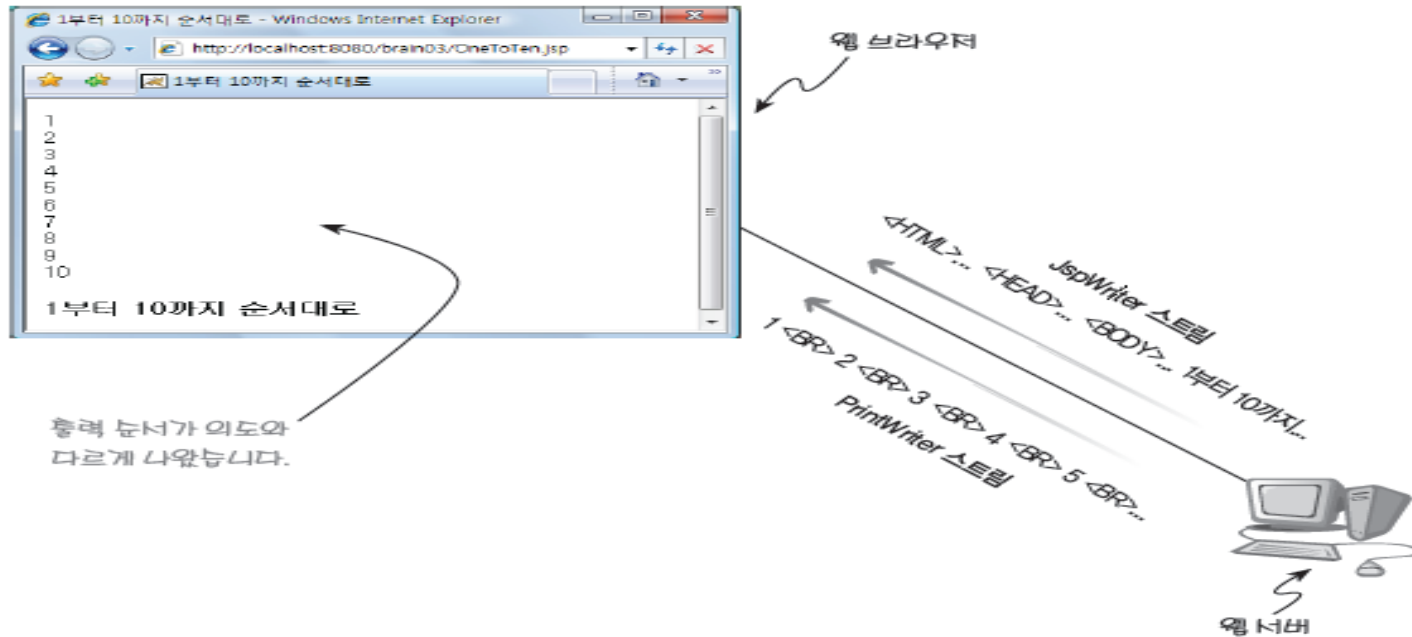
```
<%@page contentType="text/html; charset=euc-kr"%>
<HTML>
  <HEAD><TITLE>1부터 10까지 순서대로</TITLE></HEAD>
  <BODY>
    <H3>1부터 10까지 순서대로</H3>
    <%
      for (int cnt = 1; cnt <= 10; cnt++)
        out.println(cnt + "<BR> ");
    %>
  </BODY>
</HTML>
```



3. JSP 페이지의 내장 변수

❖ out 내장 변수

- JspWriter 나 PrintWriter처럼 스트림 형태로 데이터를 출력하는 클래스는 송신측과 수신측 사이에 가상의 통로를 만든다.
- JSP 페이지에서 PrintWriter 객체를 새로 만들면 기존의 out 내장 변수가 관리하던 통로와 더불어 두 개의 통로가 공존하게 된다.



[그림 3-20] response.getWriter 메서드를 호출하면 안 되는 이유

3. JSP 페이지의 내장 변수

❖ out 내장 변수

- out.print, out.println 메서드를 통해 출력되는 내용뿐만 아니라, JSP 페이지 안에 있는 HTML 코드와 익스프레션을 통해 출력되는 내용도 모두 out 내장 변수를 통해 웹 브라우저로 출력된다.
- page 지시자의 buffer 애트리뷰트를 이용하면 출력 버퍼의 크기를 바꿀 수 있다. buffer 애트리뷰트에는 버퍼의 크기를 킬로바이트 단위의 정수로 써야 하며, 뒤에 kb라는 단위 표시를 붙여 써야 한다.

```
<%@page buffer="2kb" %>
```

출력 버퍼의 크기

- 버퍼의 실제 크기를 알고 싶을 경우에는 out 내장 변수에 대해 getBufferSize라는 메서드를 호출하면 된다.

```
int bsize = out.getBufferSize();
```

출력 버퍼의 크기를 바이트 단위로 리턴하는 메서드



3. JSP 페이지의 내장 변수

❖ out 내장 변수

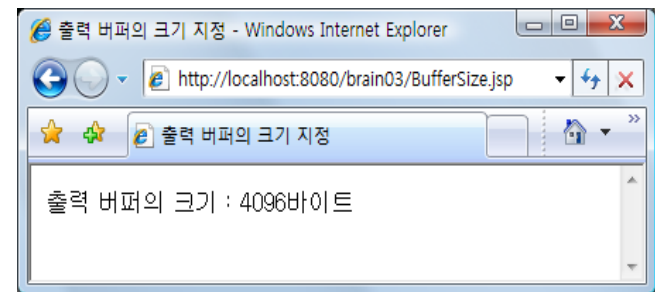
- page 지시자의 buffer 애트리뷰트에 none이라는 값을 지정하면 out 내장 변수를 통해 출력되는 내용이 출력 버퍼를 거치지 않고 웹 브라우저로 바로 전송되도록 만들 수 있다.

```
<%@page buffer="none" %>
```

출력 버퍼를 사용하지 않겠다는 표시

[예제3-10] 출력 버퍼의 크기를 바꾸는 JSP 페이지

```
<%@page contentType="text/html; charset=euc-kr" %>
<%@page buffer="4kb" %>
<HTML>
  <HEAD><TITLE>출력 버퍼의 크기 지정</TITLE></HEAD>
  <BODY>
    출력 버퍼의 크기 : <%= out.getBufferSize() %>바이트
  </BODY>
</HTML>
```



[그림 3-21] 예제 3-10의 실행 결과

3. JSP 페이지의 내장 변수

❖ response 내장 변수

- response 내장 변수는 서블릿 클래스에 있는 doGet, doPost 메서드의 두 번째 파라미터와 동일한 역할을 한다. 이 변수는 javax.servlet.http.HttpServletResponse 타입이기 때문에 이 인터페이스에 속하는 여러 가지 메서드들을 호출 할 수 있다.

```
response.sendRedirect( "http://www.hanb.co.kr/ ");
```

호출할 웹 자원의 URL

- sendRedirect 메서드를 호출할 때 주의할 점: 이 메서드를 호출하기 전과 후에 웹 브라우저로 데이터를 출력하면 안 된다.

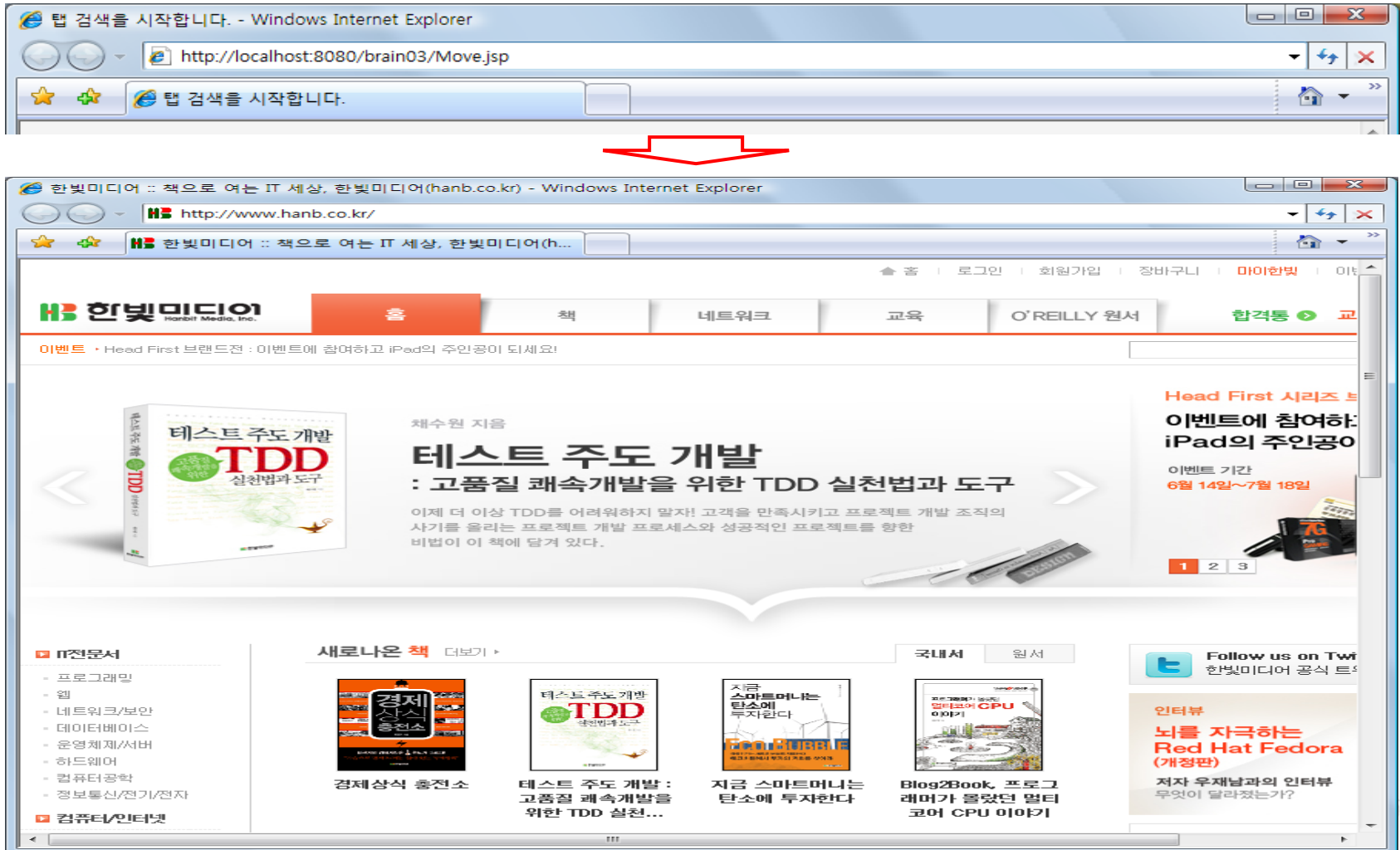
[예제3-11] 다른 웹 페이지로 이동하는 JSP 페이지

```
<% response.sendRedirect( "http://www.hanb.co.kr/ "); %>
```



3. JSP 페이지의 내장 변수

❖ response 내장 변수

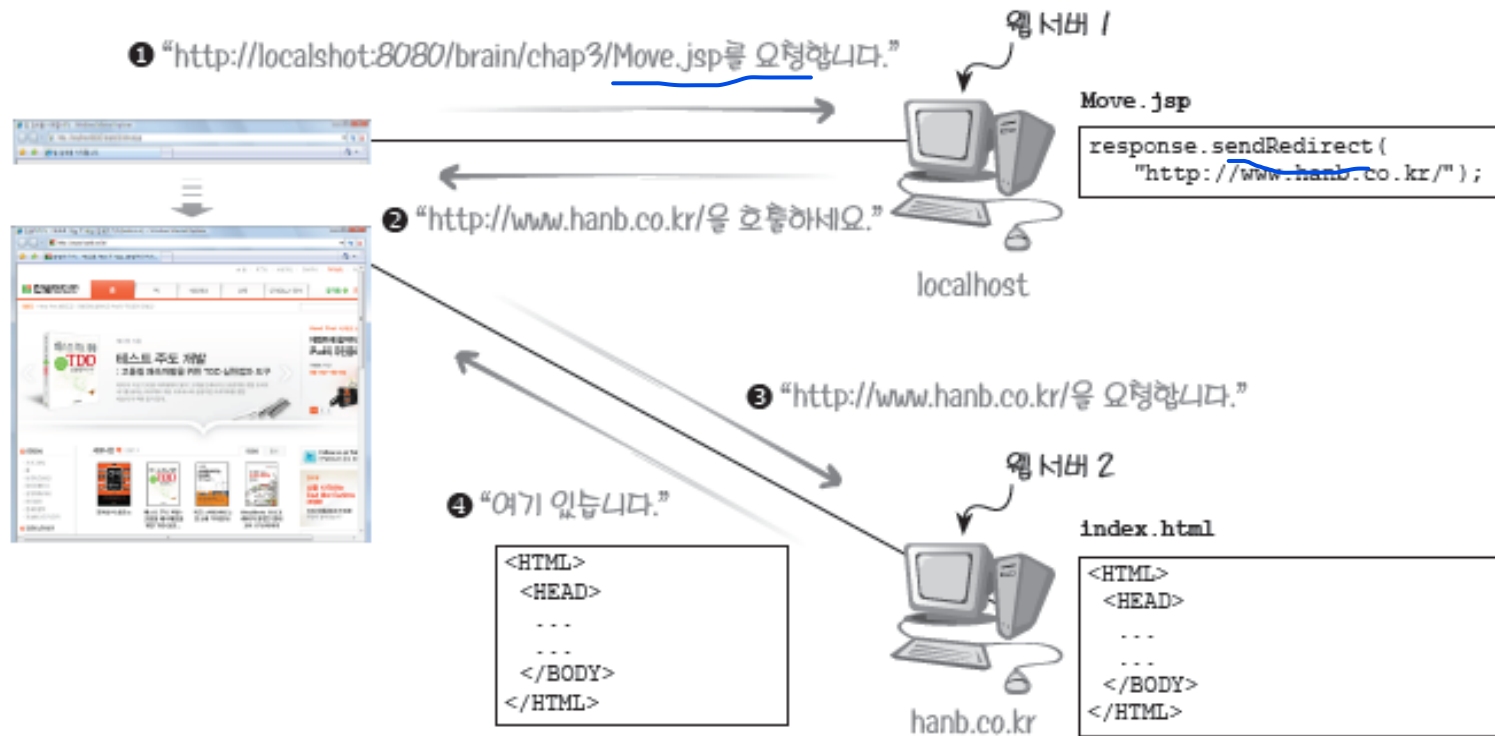


[그림 3-22] 예제 3-11의 실행 결과

3. JSP 페이지의 내장 변수

❖ response 내장 변수

- sendRedirect 메서드는 파라미터로 지정한 URL을 직접 호출하는 것이 아니라 그 URL을 이용해서 다시 웹 자원을 호출하라는 메시지를 웹 브라우저로 보낼 뿐이다.



[그림 3-23] sendRedirect 메서드의 작동 원리



3. JSP 페이지의 내장 변수

❖ application 내장 변수

- application 내장 변수는 웹 애플리케이션에 관련된 여러 가지 기능을 제공한다.
- application 내장 변수에 대해 호출할 수 있는 `getContextPath` 메서드는 웹 애플리케이션의 URL 경로명을 리턴하는 메서드이다.

```
String appPath = application.getContextPath();
```

웹 애플리케이션의 URL 경로명을 리턴하는 메서드

목표지 저장소

- application 내장 변수에 대해 호출할 수 있는 `getRealPath` 메서드는 웹 애플리케이션 내에서의 파일 경로명을 파일시스템 전체에 대한 절대 경로명으로 바꾸는 메서드이다.

```
String absolutePath = application.getRealPath( "/sub1/Intro.html ");
```

웹 애플리케이션 내에서의 파일의 경로명

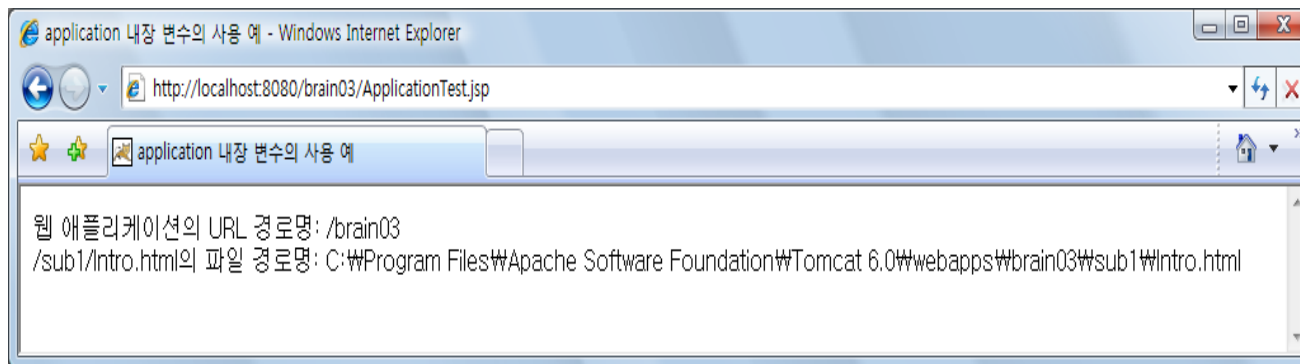


3. JSP 페이지의 내장 변수

❖ Application 내장 변수

[예제3-12] application 내장 변수의 사용 예

```
<%@page contentType= "text/html; charset=euc-kr" %>
<HTML>
  <HEAD><TITLE>application 내장 변수의 사용 예</TITLE></HEAD>
  <BODY>
    <%
      String appPath = application.getContextPath();
      String filePath = application.getRealPath( "/sub1/Intro.html " );
    %>
    웹 애플리케이션의 URL 경로명: <%= appPath %> <BR>
    /sub1/Intro.html의 파일 경로명: <%= filePath %> <BR>
  </BODY>
</HTML>
```



[그림 3-24] 예제 3-12의 실행 결과





Thank You !

뇌를 자극하는 JSP & Servlet