

JSP의 기초

뇌를 자극하는 JSP & Servlet



Contents

❖ 학습목표

JSP 기술에서는 웹 애플리케이션을 JSP 페이지 형태로 구현한다. 이 장에서는 JSP 페이지를 작성하는데 필요한 여러 가지 기초 지식을 배워보자.

❖ 내용

- JSP 페이지란?
- JSP 의 기초 문법
- JSP 페이지의 내장 변수
- 파일 입출력
- 다른 JSP 페이지 호출하기

1. JSP 페이지란?

- JSP 기술에서 웹 애플리케이션을 구연할 때 작성하는 코드를 JSP 페이지라고 한다.
- JSP 페이지는 HTML 문서의 사이에 JSP 문법의 코드가 삽입되는 형태로 작성된다.

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

      <HTML>

      <HEAD><TITLE>1부터 100까지의 합
      TITLE></HEAD>

      <BODY>
      (%)

      int total = 0;
      for (int cnt = 1; cnt <= 100; cnt++)</td>

      total += cnt;
      HTML 문서의 사이사이에 JSP 문법의 코드가 삽입되니다

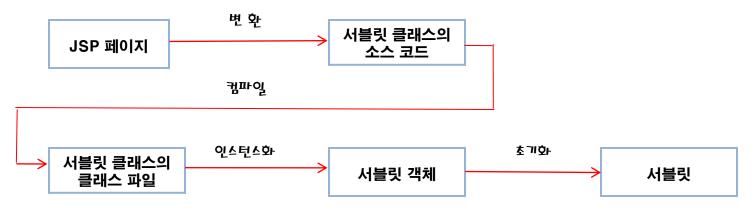
      </BODY>
      (/BODY>

      </HTML>
      [그림 3-1] JSP 페이지의 형태
```

JSP 페이지에 있는 HTML 코드는 웹 브라우저로 그대로 전송되지만, JSP 문법의 코드는 웹 컨테이너 쪽에서 실행되고 그 결과만 웹 브라우저로 전송된다.

1. JSP 페이지란?

- 웹 컨테이너는 JSP 페이지 전체를 서블릿 클래스의 소스 코드로 변환한 다음에, 그 소스 코드를 컴파일해서 그 결과를 가지고 서블릿 객체를 만들고, 그 서블릿 객체를 초기화해 서 서블릿을 만든다.
- 웹 브라우저로부터 URL이 왔을 때 실행되는 것은 서블릿이다.



[그림 3-2] JSP 페이지가 실행되기까지의 과정

❖ JSP의 다양한 문법

- JSP의 문법에는 세가지 형태가 있다.
 - <%로 시작해서 %>로 끝나는 형태
 - \${로 시작해서}로 끝나는 형태
 - 〈jsp:forward〉 나 〈C:if〉와 같은 XML 태그 형태
- 지시자와 스크립팅 요소
 - 〈%로 시작해서 %〉로 끝난다.

[그림 3-3] JSP의 문법 - 지시자와 스크립팅 요소

**IAIX*(directive)

❖ JSP의 다양한 문법

- 익스프레션 언어(expression language): 자바 코드를 대신할 새로운 언어
 - \${로 시작해서} 로 끝나고, 그 안에 데이터 이름이나 간단한 식을 쓸 수 있다.
- 액션(action)
 - XML 태그 형태로 기술되며, 자바의 if 문, for 문, switch 문에 해당하는 태그들이 있다.



❖ JSP의 다양한 문법



```
JSP 페이지

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

<HEAD><TITLE>1부터 100까지의 합</TITLE></HEAD>
<BODY>

1부터 100까지의 합은? ${result}
</BODY>
</HTML>
```

[그림 3-4] JSP 문법 - 익스프레션 언어

이스프레션 어어

(expression language)

❖ JSP의 다양한 문법



```
SP 페이지

<%@page contentType= "text/html; charset=euc-kr "%>
<%@taglib prefix= "c" uri= "http://java.sun.com/jsp/jstl/core " %>
<HTML>

<HEAD><TITLE>복권 당첨 변호</TITLE></HEAD>
<BODY>

<c:forEach var= "num " items= "${ARR} ">

${num} < BR>

</c:forEach>
</HTML>
```

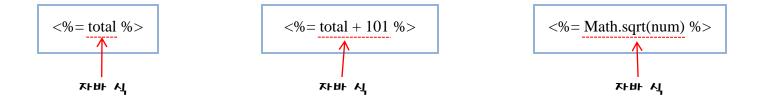
스크립팅 요소의 문법

- 스크립팅 요소(scripting elements)란 다음 세가지 문법을 말한다.
 - 스크립틀릿(scriptlet)
 - 익스프레션(expression)
 - 선언부(declaration)
- 스크립틀릿(scriptlet)은 〈%로 시작해서 %〉로 끝나고, 그 사이 자바 명령문이 들어갈수 있다.

• 이 명령문은 웹 브라우저로 전송되는 것이 아니라 웹 서버 쪽에 실행 된다.

스크립팅 요소의 문법

- 익스프레션(expression)은 <%=로 시작해서 %>로 끝나고 그 사이에 자바 식이 들어갈
 수 있다.
 - 이 식은 상수나 변수 이름 하나로 구성될 수도 있고, 연산자를 포함할 수도 있으며, 리턴 값이 있는 메서드 호출식이 될 수도 있다.



• 이 식은 웹 서버 쪽에서 실행되고 그 결과만 웹 브라우저로 전송된다.



❖ 스크립팅 요소의 문법

```
      서블릿 클래스의 코드

      out.println( "<HTML>");

      out.println( "<HEAD><TITLE>1부터 100까지의 합</TITLE></HEAD> ");

      out.println( "<BODY> ");

      int total = 0;

      for (int cnt = 1; cnt <= 100; cnt++)</td>

      total += cnt;

      out.print( "1부터 100까지 더한 값은? " );

      out.println(total);

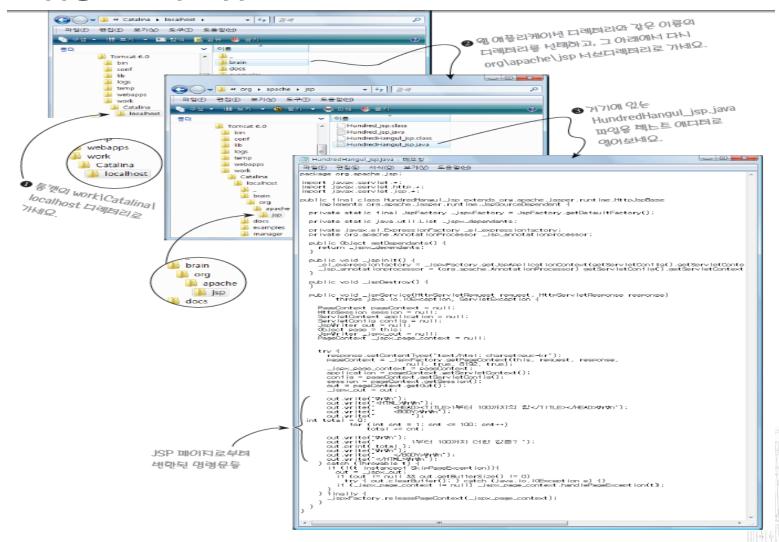
      out.println("</BODY> ");

      out.println("</HTML>");
```

스크립틀릿 안에 있던 자바 명령문

[그림 3-6] JSP 페이지의 코드가 서블릿 클래스의 코드로 변환되는 방법

소크립팅 요소의 문법



[그림 3-7] JSP 페이지로부터 변환된 서블릿 클래스의 소스 코드

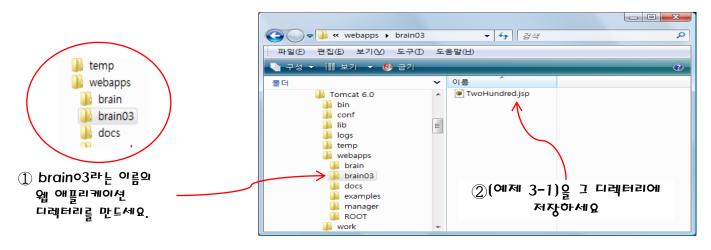
스크립팅 요소의 문법

JSP 페이지의 스크립틀릿은 모두 _jspService 메서드 안에 들어가는 명령문이 되므로,
 한 스크립틀릿 안에서 선언한 변수를 그 뒤에 나오는 다른 스크립틀릿 안에 사용하는 것이 가능하다.

```
[예제3-1] 여러 개의 스크립틀릿이 있는 JSP 페이지
<%@page contentType= "text/html; charset=euc-kr "%>
                                                                               total 변수를 선언한다
<HTML>
    <HEAD><TITLE>1부터 200까지의 합</TITLE></HEAD>
    <BODY>
        <%
            int total = 0:
            for (int cnt = 1; cnt <= 100; cnt++)
               total += cnt;
        %>
        1부터 100까지의 합 = <%= total %> <BR>
        <%
            for (int cnt = 101; cnt <= 200; cnt++)
               total += cnt;
        %>
                                                                              total 변수를 사용한다
        1부터 200까지의 합 = <%= total %> <BR>
   </BODY>
</HTML>
```

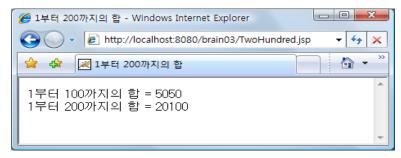
스크립팅 요소의 문법

■ JSP 페이지를 설치하는 방법



[그림 3-8] 예제 3-1의 설치 방법

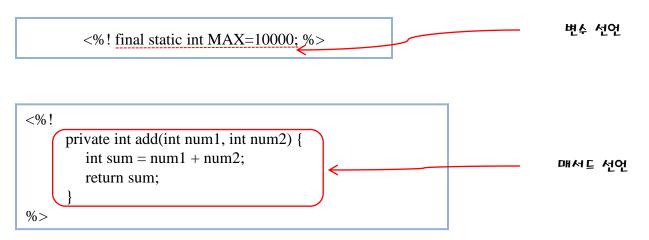
■ JSP 페이지를 실행한 결과



[그림 3-9] 예제 3-1의 실행 결과

❖ 스크립팅 요소의 문법

- 선언부(declaration)는 <%!로 시작해서 %>로 끝나고, 그 사이에 변수 선언문이나 메서
 드 선언문을 쓸 수 있다.
- 여기에 선언한 변수, 메서드는 JSP 페이지로부터 변환된 서블릿 클래스의 멤버가 되므로, final, public, private, protected, static 등의 키워드를 붙이는 것도 가능하다.

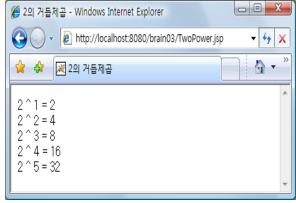


- 선언부 안에 변수를 선언할 때 주의해야 할 점
 - JSP 페이지로부터 변환된 서블릿 클래스는 기본적으로 멀티 스래드 모델로 작동하고, 멀티-스레드 모델로 작동하는 서블릿 클래스 안에는 인스턴스 변수를 선언하면 안 된다.

스크립팅 요소의 문법

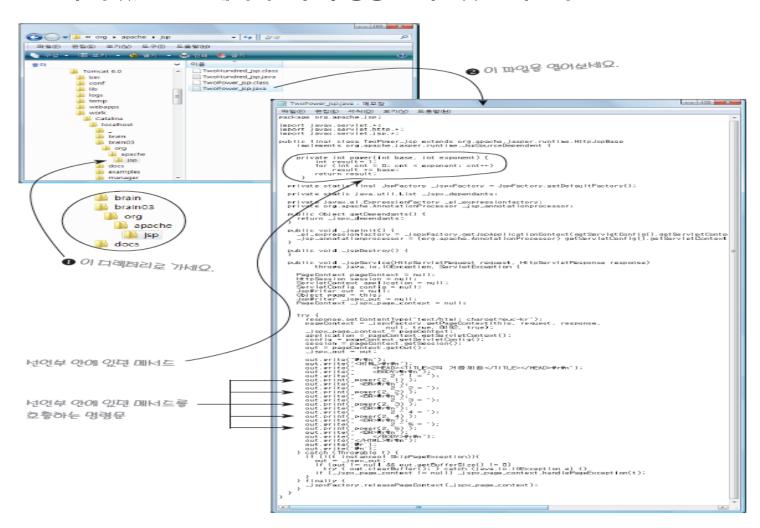
선언부의 사용 예를 보여주는 JSP 페이지

```
[예제3-2] 선언부를 포함한 JSP 페이지
<%@page contentType="text/html; charset=euc-kr"%>
<HTML>
    <HEAD><TITLE>2의 거듭제곱</TITLE></HEAD>
    <BODY>
         2 \land 1 = <\% = power(2, 1) \% > <BR >
         2 \land 2 = <\% = power(2, 2) \% > <BR >
         2 \land 3 = <\% = power(2, 3) \% > <BR >
         2 \land 4 = <\% = power(2, 4) \% > <BR >
         2 \land 5 = <\% = power(2, 5) \% > <BR >
    </BODY>
</HTML>
<%!
      private int power(int base, int exponent) {
         int result= 1:
         for (int cnt = 0; cnt < exponent; cnt++)
                                                                       선언부
            result *= base;
        return result;
%>
```



❖ 스크립팅 요소의 문법

■ 선언부가 있는 JSP 페이지로부터 생성된 서블릿 클래스의 소스 코드



자시자의 문법

- 지시자(directive)는 JSP의 다른 문법들(스크립팅 요소, 익스프레션 언어, 액션)과는 다른 목적으로 사용된다.
- 웹 브라우저로부터의 요청을 처리하는 것이 아니라, 웹 컨테이너가 JSP 페이지를 서블 릿 클래스로 변환할 때 필요한 정보들을 기술하는 역할을 한다.
- JSP 페이지에 사용할 수 있는 지시자의 종류
 - page ストノト
 - include 지시자
 - taglib 지시자

❖ 지시자의 문법

- 모든 지시자는 〈%@으로 시작하고 %〉로 끝나야 한다.
- <%@ 바로 다음에는 지시자 이름이 와야 하고, 지시자 이름 다음에는 여러 가지 애트리 뷰트가 올 수 있다.



■ page 지시자는 JSP 페이지 전체에 적용되는 정보를 기술하기 위해 사용된다.



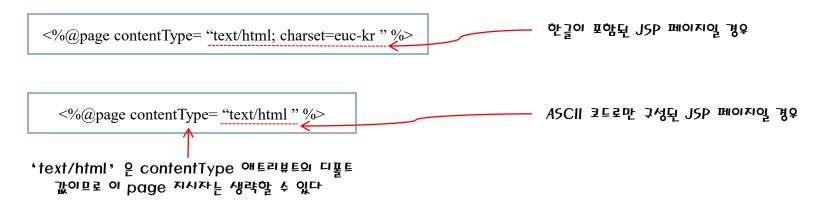
❖ 지시자의 문법

애트리뷰트 이름	기술하는 정보/애트리뷰트의 역할	다루는 장
contentType	JSP 페이지가 생성하는 문서의 종류와 그 문서를 웹 브라우저로 전송할 때 사용되는 인코딩 타입	3 장
import	스크립팅 요소 안에서 사용할 자바 클래스와 인터페이스를 임포트하기 위해 사용하는 애트리뷰트	3 장
buffer	출력 버퍼의 크기	3 장
autoFlush	출력 버퍼가 모두 찼을 때의 동작	3 장
isThreadSafe	JSP 페이지가 싱글-스레드 모드로 작동하도록 만들기 위해 필요한 애트리뷰트	3 장
session	JSP 페이지의 세션 참여 여부	4 장
errorPage	에러를 처리할 JSP 페이지의 URL	5 장
isErrorPage	에러를 처리하는 JSP 페이지인지 여부	부록 A
isELIgnored	익스프레션 언어의 무시/처리 여부	다루지 않음
pageEncoding	JSP 페이지의 인코딩 타입	다루지 않음
info	JSP 페이지에 대한 설명	다루지 않음
extends	JSP 페이지로부터 생성되는 서블릿 클래스의 슈퍼클래스	다루지 않음
language	스크립팅 요소 안에서 사용할 프로그래밍 언어. 현재는 'java'라는 값만 지정할 수 있음	다루지 않음
${\bf deferred Syntax Allowed As Literal}$	익스프레션 언어의 예약 문자열인 '#{'를 사용했을 때의 에러 발생 여부	다루지 않음
trimDirectiveWhitespaces	지시자 바로 다음에 있는 공백 문자를 제거하기 위해 사용하는 애트리뷰트	다루지 않음

[표 3-1] page 지시자의 애트리뷰트

❖ 지시자의 문법

contenType 애트리뷰트는 JSP 페이지가 생성하는 문서의 종류와 그 문서를 웹 브라우저로 전송할 때 사용할 인코딩 방식을 지정하기 위해 사용한다. 이 두 값은 세미콜론(;)으로 구분해서 써야 한다.



■ JSP 페이지를 이용하면 HTML 문서뿐만 아니라 XML 문서도 생성할 수 있다. 그럴 때는 contentType 애트리뷰트에 문서의 종류를 text/xml로 써야 한다.

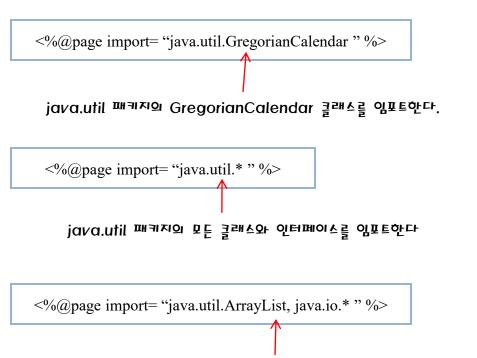
❖ 지시자의 문법

■ JSP 페이지가 서블릿 클래스로 변환될 때 contentType 애트리뷰트의 값은 setContentType 메서드 호출문의 파라미터가 된다.

```
한글 코드 디니다가 있을 때
try {
 response.setContentType("text/html; charset=euc-kr");
 pageContext = _jspxFactory.getPageContext(this, request, response,
                   null, true, 8192, true);
  _jspx_page_context = pageContext;
 application = pageContext.getServletContext();
 config = pageContext.getServletConfig();
 session = pageContext.getSession();
 out = pageContext.getOut();
                                                               하글이 데대로 표시된니다.
  _jspx_out = out;
 out.write("#r#n");
out.write("<HTML>#r#n");
                (HEAD><TITLE>1부터 100까지의 합에/TITLE></HEAD>#r#n");
 out.write(
                <BODY>#r#n")
 out.write(
 out.write('
                                                                 한글 코드 디니다가 없을 때
 response.setContentType("text/html");
 pageContext = _jspxFactory.getPageContext(this, request, response,
                   null, true, 8192, true);
  _jspx_page_context = pageContext;
 application = pageContext.getServletContext();
 config = pageContext.getServletConfig();
 session = pageContext.getSession();
                                                               하글이 깨뎌서 표시된니다.
 out = pageContext.getOut();
  _ispx_out = out;
 out.write("#r#n");
out.write("<HTML>#r#n");
 out.write("
                out.write(
                <BODY>#r#n");
 out.write(
```

자시자의 문법

page 지시자의 import 애트리뷰트는 자바의 import 문과 마찬가지로 다른 패키지에 속하는 클래스나 인터페이스를 임포트하는 역할을 한다.



java.util.ArrayList 클래스와 java.io 패키지의 모든 클래스, 인터페이스를 임포트한다

❖ 지시자의 문법

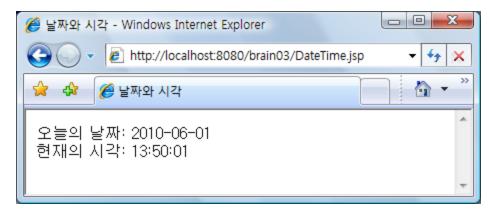
■ import 애트리뷰트의 사용 예를 보여주는 JSP 페이지

```
[예제3-3] page 지시자의 import 애트리뷰트 사용 예

<%@page contentType= "text/html; charset=euc-kr"%>
<%@page import= "java.util.GregorianCalendar"%>
<HTML>

<HEAD><TITLE>날짜와 시각</TITLE></HEAD>
<BODY>
<%

GregorianCalendar now = new GregorianCalendar();
String date = String.format( "%TF", now);
String time = String.format( "%TT", now);
%>
오늘의 날짜: <% = date %> <BR>
현재의 시각: <% = time %> <BR>
</BODY>
</HTML>
```



자시자의 문법

■ page 지시자에 여러 종류의 애트리뷰트를 함께 쓸 수도 있다.

```
<%@page contentType= "text/html; charset=euc-kr" import= "java.util.GregorianCalendar" %>

여러 중류의 애트리뷰트를 함께 쓸 수 있다
```

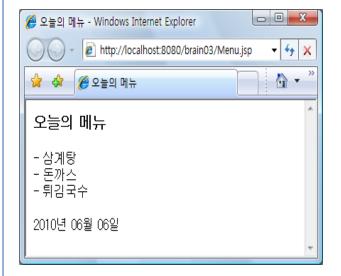
- include 지시자는 다른 JSP 페이지 또는 HTML 문서를 불러다가 현재 JSP 페이지의 일부로 만들기 위해 사용한다.
- 불러올 대상은 file 애트리뷰트를 이용해서 지정할 수 있으며, 이 애트리뷰트의 값은 지시자가 속하는 JSP 페이지를 기준으로 한 상대적인 URL로 해석된다.



❖ 지시자의 문법

■ include 지시자의 사용 예를 보여주는 JSP **페이지**

```
[예제3-4] include 지시자의 사용 예
<%@page contentType= "text/html; charset=euc-kr"%>
<HTML>
   <HEAD><TITLE>오늘의 메뉴</TITLE></HEAD>
   <BODY>
       <H3>오늘의 메뉴</H3>
       - 삼계탕 <BR>
       - 돈까스 <BR>
       - 튀김국수 <BR><BR>
       <%@include file= "Today.jsp "%>
   </BODY>
</HTML>
                               Today.jsp를 include한다
   <%@page contentType="text/html; charset=euc-kr"%>
   <%@page import="java.util.*"%>
   <% GregorianCalendar now = new GregorianCalendar(); %>
   <%= String.format("%TY년 %Tm월 %Td일", now, now, now) %>
```





❖ 지시자의 문법

- taglib 지사자는 JSP 문법 중 하나인 액션(action)을 사용할 때 필요하다.
- taglib 지시자는 액션이 속한 라이브러리를 설치해야만 사용할 수 있다.

<%@taglib prefix= "c" uri= "http://java.sun.com/jsp/jstl/core" %>

애션이 속하는 라이브러리를 지정하는 지시자

❖ 주석을 기술하는 방법

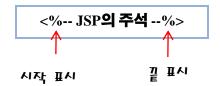
- JSP 페이지에 주석을 다는 방법은 다양하다.
- JSP 페이지의 HTML 코드 부분 : <!--로 시작해서 -->로 끝나는 HTML 주석을 쓸 수 있다.



JSP 페이지의 스크립팅 요소 안 : 자바 문법을 따르는 주석을 쓸 수 있다.



■ JSP 고유의 주석을 사용할 수 있다.





❖ 주석을 기술하는 방법

• 여러 가지 주석의 사용 예를 보여주는 JSP 페이지

```
[예제3-5] 여러 가지 주석을 포함하는 JSP 페이지
<%@page contentType= "text/html; charset=euc-kr"%>
<HTML>
    <HEAD><TITLE>1부터 10까지의 곱</TITLE></HEAD>
    <!-- 이것은 JSP에 의해 생성된 HTML 문서입니다. -->
    <BODY>
       <%-- 다음은 데이터를 처리하는 스크립플릿입니다. --%>
       <% int result = 1: // 곱을 저장할 변수
          /* 1부터 10까지 곱하는 반복문 */
          for (int cnt = 1; cnt \leq 10; cnt++)
             result *= cnt:
       %>
       1부터 10까지 곱한 값은? <%= result %>
    </BODY>
</HTML>
```

```
TenMultiply_jsp.java - 메모장
파일(E) 편집(E) 서식(O) 보기(V) 도움말(H)
 public void _jspService(HttpServletRequest request, HttpServletResponse response)
       throws java.io.IOException, ServletException {
   PageContext pageContext = null;
   HttpSession session = null;
   ServietContext application = null;
   ServletConfig config = null;
   Jsp₩riter out = null;
   Object page = this;
   Jsp₩riter _jspx_out = null;
   PageContext _jspx_page_context = null;
     response.setContentType("text/html; charset=euc-kr");
     pageContext = _jspxFactory.getPageContext(this, request, response,
                        null, true, 8192, true);
      _jspx_page_context = pageContext;
     application = pageContext.getServletContext();
                                                                        HTML 주석
     config = pageContext.getServletConfig();
     session = pageContext.getSession();
     out = pageContext.getOut();
     _ispx_out = out;
     out.write("₩r₩n");
     out.write("<HTML>#r#n");
                     "~\#EAD><fiTLE>1부터 10까지의 곱</TITLE></HEAD>#r#n"); ✔
<!---이것은 JSP에 의해 생성된 HTML 문서입니다. -->#r#n");
<BOOY;*#r#n");
     out.write("
     out.write(
     out.write(
     out.write(
     out.write("\r\n");
     out.write("
          write(
t = 1; // 곱을 저장할 변수
/* 1부터 10까지 곱하는 반복문 */
for (int cnt = 1; cnt <= 10; cnt++)
int result = 1;
                                                                          JAVA 주석
               result *= cnt;
     out.write("
                         1부터 10DN지 곱한 값은? ");
     out.print( result );
     out.write(" \r\n"
out.write(" </
                     </BODY>#r#n");
     out.write("</HTML>\r\n");
   } catch (Throwable t)
     if (!(t instanceof SkipPageException)){
       if (out != null && out.getBufferSize() != 0)
         try { out.clearBuffer(); } catch (java.io.IOException e) {}
       if (_jspx_page_context != null) _jspx_page_context.handlePageException(t);
      _jspxFactory.releasePageContext(_jspx_page_context);
```



Thank You!

뇌를 자극하는 JSP & Servlet

