

3강

JSP 개요

방송대 컴퓨터과학과 김희천 교수

목차

- 1 JSP 기술
- 2 JSP 문서의 구성
- 3 page 지시어
- 4 스크립트 요소



01

JSP 기술

1) JSP 기술

1. JSP 기술

> JSP

- 동적으로 웹 콘텐츠를 만들기 위한 기술
 - 동적으로 웹 페이지를 만드는 기술
- 서버 측 스크립트 언어
 - HTML 페이지 안에 자바 코드를 포함시킬 수 있음
 - **서블릿 코드를 작성하는 편리한 방법**을 제공

> JSP 언어의 확장 메커니즘

- 표현 언어(EL)의 사용
 - 서버 측 객체나 데이터를 쉽게 사용하고 표현할 수 있음
- JSTL과 같은 태그 라이브러리의 사용



2) JSP 문서(1/2)

➤ 템플릿 데이터

- 정적 데이터로 HTML이나 XML 형식의 텍스트
- 응답 결과를 만들 때, 그대로 출력되는 텍스트

➤ JSP 요소 - 동적인 데이터의 추가

- JSP 태그로 표현되는 프로그래밍적 요소
- 적절한 Java 코드로 번역되어 동적 콘텐츠가 만들어짐

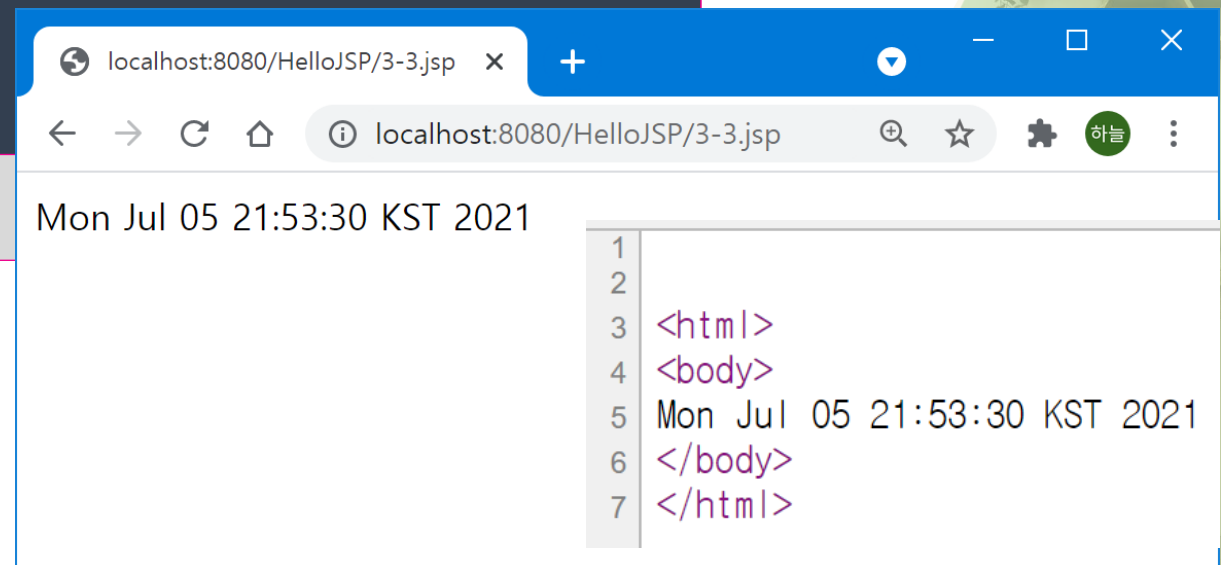


2) JSP 문서(2/2)

1. JSP 기술

```
1 <%@ page contentType="text/html; charset=UTF-8"%>
2 <%@ page import="java.util.Date" %>
3 <html>
4 <body>
5 <%
6     Date today = new Date();
7     out.println(today);
8 %>
9 </body>
10 </html>
```

3-3.jsp



3) JSP 페이지의 실행

- 실습: 웹 프로젝트 폴더를 만들지 않고 간단히 실행 결과만을 확인하기
 - [톰캣설치폴더] `₩webapps₩ROOT₩3-3.jsp`를 작성
 - [톰캣설치폴더] `₩bin₩startup.bat`를 실행
 - Tomcat(서버)을 시작함
 - 웹 브라우저(클라이언트)로 `http://localhost:8080/3-3.jsp`를 실행



02

JSP 문서의 구성

1) JSP 요소

2. JSP 문서의 구성

➤ JSP 페이지의 구성 요소

- 스크립트 요소 (scripting elements)
- 지시어 (directives)
- 액션 태그 (action tags)
- 내장 객체 (implicit objects)
- 표현 언어 (expression language)
- JSTL과 사용자 정의 태그
- 주석
 - `<%-- 설명 --%>`



2) 스크립트 요소

2. JSP 문서의구성

> 스크립트 요소의 종류

■ 스크립트릿 (Scriptlet)

- JSP 페이지에 삽입되는 Java 코드 조각
- `<% ... %>`

```
<%  
    String a = "Hello, Java";  
%>  
  
<h1><%= a %></h1>
```

■ 표현식 (Expression)

- 변수나 수식의 값을 출력할 때, JSP 페이지에 삽입하는 식
- `<%= ... %>`

■ 선언 (Declaration)

- JSP 코드에서 사용될 변수 또는 메서드의 선언문
- `<%! ... %>`

```
<%!  
    public int add(int a, int b)  
    {  
        return a + b;  
    }  
%>
```

3) 지시어(1/3)

> 의미

- JSP 페이지에 대한 설정 정보를 웹 컨테이너에 지시
- JSP 페이지의 **번역과 실행 과정에 사용되는 정보**

```
<%@ 지시어이름 속성1="값1" 속성2="값2"... %>
```



3) 지시어(2/3)

2. JSP 문서의구성

➤ 지시어의 종류

■ include

- JSP 페이지의 특정 영역에 다른 문서를 포함
- 번역 과정에서 복사하여 삽입시킬 텍스트나 코드를 지정

```
<%@ include file="/jsp/userInfo.jsp" %>
```

■ page

- JSP 페이지에 대한 정보를 지정

```
<%@ page contentType="text/html; charset=UTF-8" %>  
<%@ page import="java.util.Date" %>
```



3) 지시어(3/3)

2. JSP 문서의구성

■ taglib

- JSP 페이지에서 사용할 태그 라이브러리를 선언함
- 태그 라이브러리의 위치와 접두어를 지정

```
<%@ taglib uri= "/mycustomtags" prefix= "mycust" %>
```



4) 내장 객체

➤ 의미

- 자주 사용되는 기능을 JSP 컨테이너가 객체 형태로 제공한 것
 - 묵시적으로 이미 정의되어 있는 변수(자바 객체)

➤ 내장 객체(implicit object)의 사용

- 별도의 선언이나 초기화 없이 사용 가능
- 요청 파라미터 읽어오기, 응답 결과 전송하기, 세션 처리하기, 웹 애플리케이션 정보 읽어오기 등
- 예: request, response, session, application, out, page 등



5) 액션 태그(1/2)

2. JSP 문서의구성

➤ 액션 태그의 의미

- 요청을 처리할 때, 특별한 기능을 수행하는 것
- 형식은 XML 태그와 같은 모양
 - 시작 태그(접두어와 액션이름, 속성과 값), 몸체, 마감 태그
 - 표준 액션 태그는 접두어로 jsp를 사용

```
<jsp:액션이름    속성="값" ... >  
    몸체  
</jsp:액션이름>
```

```
<jsp:액션이름    속성="값" ... />
```



5) 액션 태그(2/2)

2. JSP 문서의 구성

```
1 <%@ page contentType="text/html; charset=UTF-8" %>
2 <html>
3   <body>
4     <jsp:include page="3-3.jsp" />
5   </body>
6 </html>
```

3-4.jsp



6) 표현 언어

2. JSP 문서의구성

➤ 의미

- 데이터를 쉽게 다루기 위한 간단한 스크립트 언어
- 형태는 **`${expr}`**이며 값을 표현
 - 스크립트 요소(표현식)보다 사용하기 쉬움
- 스크립트 요소 밖에서 사용
 - 액션 태그의 속성 값을 설정하거나 템플릿 데이터 내부에서 값을 출력할 때 사용
 - 표현 언어의 내장 객체나 JSTL과 함께 사용할 수 있음



7) JSTL과 사용자 정의 태그

➤ JSTL(JSP Standard Tag Library)

- 자주 사용되는 기능을 태그로 만들어 쓸 수 있음
- JSTL은 유용한 사용자 정의 태그들을 모아 표준화한 태그 모음
- 조건문, 반복문과 같은 프로그래밍적 요소를 태그 형태로 사용할 수 있게 함

➤ 사용자 정의 태그

- 사용자가 정의한 커스텀 태그
- JSP 표준 태그를 확장하기 위한 것



03

page 지시어

1) page 지시어

3. page 지시어

> 용도

- JSP 페이지에 대한 정보를 지정하여 JSP 처리에 사용하게 함
- 사용할 스크립트 언어, 상속받을 클래스, import할 클래스, JSP가 생성하는 문서의 타입, 출력 버퍼 관련 속성, 사용할 에러 페이지 등의 속성을 지정

```
<%@ page contentType="text/html; charset=UTF-8" import="java.sql.*" %>  
<%@ page session="false" %>
```

```
<%@ page import="packages.mypackage.*" errorPage="boof.jsp" %>  
<%@ page buffer="20kb" %>
```

2) page 지시어의 주요 속성(1/5)

3. page 지시어

| 속성 | 설명 | 기본 값 |
|-------------|---|-----------|
| autoFlush | 'true'일 경우 버퍼가 차면 자동으로 버퍼를 비워 내용을 클라이언트에 보냄. 'false'일 경우 버퍼가 차면 예외 발생 | true |
| contentType | 결과로 생성되는 응답의 종류(MIME 타입)와 사용될 charset 정보를 설정 | text/html |
| import | JSP 페이지에서 사용할 자바 클래스를 지정 | |
| language | JSP 페이지의 스크립트 요소에서 사용할 프로그래밍 언어를 설정 | java |
| session | JSP 페이지에서 session 객체를 사용할 지의 여부를 지정 | true |
| buffer | JSP 페이지의 출력 버퍼 크기를 지정. 'none'일 경우 버퍼를 사용하지 않음 | 8kb |

2) page 지시어의 주요 속성(2/5)

3. page 지시어

| 속성 | 설명 | 기본 값 |
|------------------------------|---|-------|
| info | JSP 페이지에 대한 설명을 텍스트로 입력할 때 사용 | |
| errorPage | JSP 페이지 실행 중 에러가 발생할 때 보여줄 페이지를 설정 | |
| isErrorPage | 현재 페이지가 에러 페이지인지의 여부를 지정. 'true'이면 현재 페이지를 에러 페이지로 사용할 수 있고, 'false'이면 에러 페이지로 사용할 수 없음 | false |
| pageEncoding | JSP 페이지 자신의 문자 인코딩을 지정 | |
| isELIgnored | 표현 언어를 사용할 지를 지정 | false |
| trimDirective Whitespaces | 생성되는 문서에서 불필요한 공백 문자를 제거할 지를 지정 | false |

2) page 지시어의 주요 속성(3/5)

➤ contentType 속성

- JSP 페이지로부터 생성되는 응답 문서의 MIME 타입과 charset을 설정
- MIME은 문서 형식에 관한 표준
 - 기본 값은 text/html이며 text/xml, text/plain 등이 있음
 - 웹 서버가 응답 메시지를 전송할 때, 먼저 문서 종류에 대한 정보를 보냄
- 한글을 사용할 경우, 다음과 같이 문자 인코딩 방법을 지정

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

2) page 지시어의 주요 속성(4/5)

> import 속성

- JDK에서 제공하는 클래스를 사용할 때, 사용할 클래스를 명시함

```
1 <%@ page contentType="text/html; charset=UTF-8" %>
2 <%@ page import="java.util.Calendar" %>
3 <html>
4 <body>
5 <%
6     calendar today = Calendar.getInstance();
7     int year = today.get(Calendar.YEAR);
8     int month = today.get(Calendar.MONTH) + 1;
9     int date = today.get(Calendar.DATE);
10 %>
... ..
```

3-5.jsp(일부)

2) page 지시어의 주요 속성(5/5)

➤ trimDirectiveWhitespaces

- 앞 부분의 지시어로 인해 생기는 불필요한 줄바꿈 문자를 제거함

```
1 <%@ page contentType="text/html; charset=UTF-8" %>
2 <%@ page import="java.util.Calendar" %>
3 <%@ page trimDirectiveWhitespaces="true" %>
4 <html>
5 <body>
6     <%
7         calendar today = Calendar.getInstance();
8         int year = today.get(Calendar.YEAR);
9         int month = today.get(Calendar.MONTH) + 1;
...     ...
```

```
1 <html>
2 <body>
3
4     오늘은 2021년 7월 6일입니다
5 </body>
6 </html>
```

(수정)3-4.jsp(일부)

04

스크립트 요소

1) 스크립트릿(1/2)

➤ 스크립트 요소

- 동적인 콘텐츠를 만들기 위해 프로그램 작성에 사용되는 요소
- 스크립트릿, 표현식, 선언

➤ 스크립트릿

- JSP 페이지에 삽입되는 Java 코드 조각
- `<% ... %>`의 형태
- 서블릿 클래스의 서비스 메서드를 작성할 때 반영됨

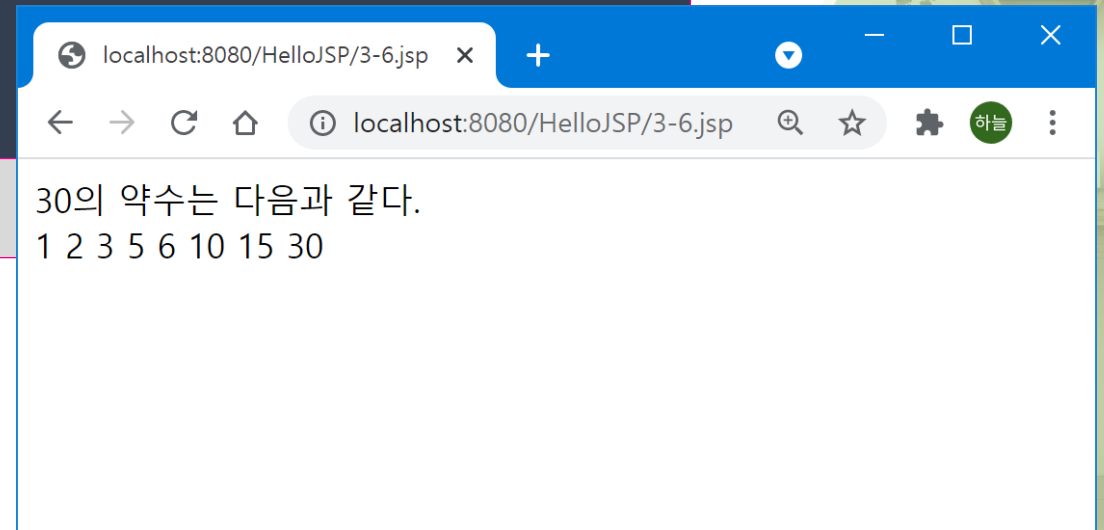


1) 스크립트릿(2/2)

4. 스크립트요소

```
1 <%@ page contentType="text/html; charset=UTF-8" %>
2 <html>
3 <body>
4   30의 약수는 다음과 같다.<br>
5   <%
6     for(int i = 1; i <= 30; i++) {
7       if(30 % i == 0) out.print(i + " ");
8     }
9   %>
10 </body>
11 </html>
```

3-6.jsp



2) 표현식(1/2)

➤ 표현식

- 스크립트릿을 사용하지 않고 변수나 수식의 값을 JSP 페이지 실행 결과에 포함시키기 위한 것
- **<%= 수식 %>**의 형태로 **세미콜론이 없음**
- 수식의 결과는 String으로 변환됨

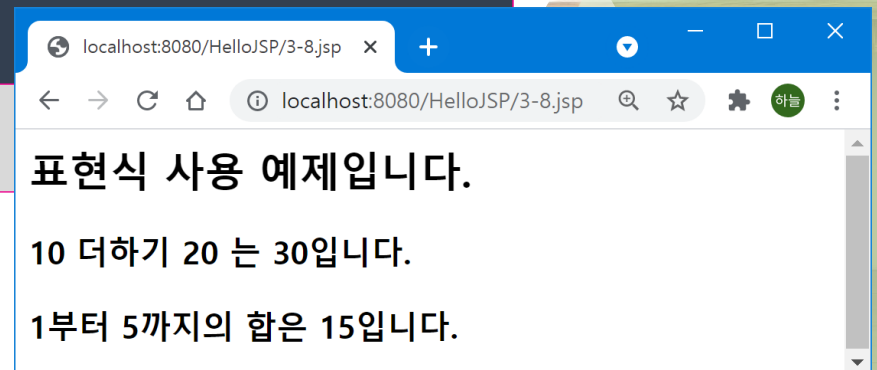


2) 표현식(2/2)

4. 스크립트 요소

```
1 <%@ page contentType="text/html; charset=UTF-8" %>
2 <html>
3 <body>
4   <%
5     String str = "표현식 사용 예제입니다.";
6     int a = 10, b = 20;
7   %>
8   <h2><%=str%></h2>
9   <h3><%=a%> 더하기 <%=b%> 는 <%=a+b%>입니다.</h3>
10  <h3>1부터 5까지의 합은 <%=1+2+3+4+5%>입니다.</h3>
11 </body>
12 </html>
```

3-8.jsp



3) 선언(1/2)

> 선언

- 스크립트릿이나 표현식에서 사용되는 서블릿 클래스의 **메서드나 멤버 변수를 선언**하기 위한 것
- 메서드나 변수는 사용 전에 먼저 선언되어야 함

```
<%!  
    public 리턴타입 메서드이름(파라미터목록) {  
        메서드 내용;  
    }  
%>
```

```
<%! public double f1=0.0; %>
```

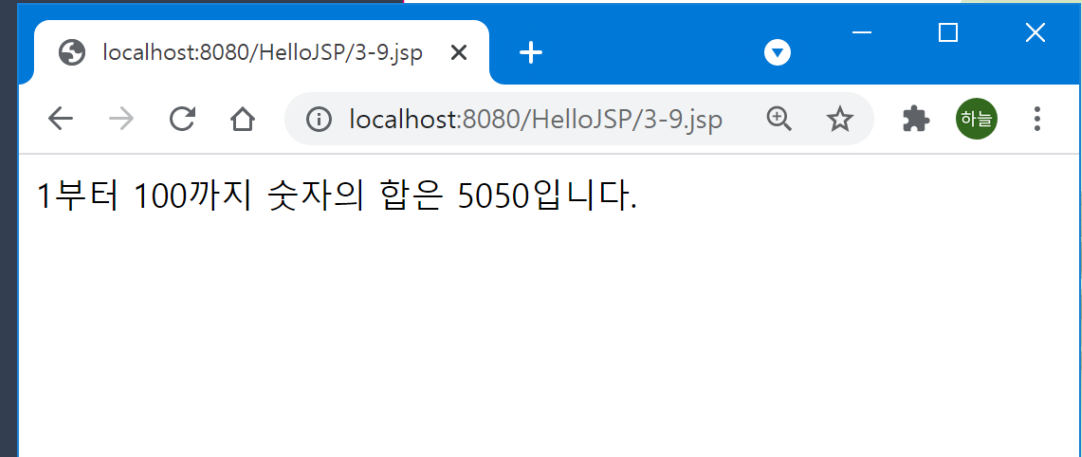


2) 선언(2/2)

4. 스크립트 요소

```
1 <%@ page contentType="text/html; charset=UTF-8"
2 %>
3 <%!
4     public int sum(int a) {
5         int s = 0;
6         for(int i = 1; i <= a; i++)
7             s += i;
8         return s;
9     }
10 %>
11 <html>
12 <body>
13 1부터 100까지 숫자의 합은 <%=sum(100)%>입니다.
</body>
</html>
```

3-9.jsp



4강

다음시간안내

JSP 동작 원리

