

## \_1장

1.

① JSP

해설 : JSP는 웹 애플리케이션을 지원하는 Java 지원 기술입니다.

JSP = HTML + JSP 태그(스크립트릿)

2.

웹 서버는 클라이언트의 요청을 받으면 해당 처리를 하여 클라이언트에 결과 페이지를 응답해줄 수 있도록 구성된 서버 또는 그 서버가 설치된 컴퓨터를 의미합니다. 웹 애플리케이션 서버는 웹 서버만으로는 데이터베이스와 연동되어 처리되어야 하는 동적인 데이터들을 처리해낼 수 없기 때문에 DBMS와 웹 서버 사이에서 제어를 해주는 프로그램을 말합니다.

3.

BEA사의 웹로직(WebLogic), IBM의 웹스피어(WebSphere), SUN사의 iPlanet, Oracle 9iAS, 티맥스의 제우스(jeus)

4.

주소 입력란에서 직접 요청을 하는 경우

form 태그의 method 속성 값을 get으로 주거나 생략할 경우

<a> 태그에 의해서 하이퍼링크가 걸린 경우

5.

서블릿은 자바 클래스 내부에서 자바 코드가 기본이 되고 출력 스트림을 통해 HTML 코드를 문자열 형태로 기술하여 클라이언트에 전송하고 JSP는 HTML 코드가 기본이 되고 스크립트릿 안에 자바 코드를 기술합니다.

6.

인코딩은 문자셋을 컴퓨터가 이해할 수 있는 바이트와의 매핑 규칙을 말합니다. 예를 들면 ASCII Code에서 ABC 등은 문자셋이고 A는 코드 65, B는 코드 66 등 바이트 순서와 매핑한 것이 인코딩입니다. 따라서 문자셋을 어떻게 매핑하느냐에 따라 하나의 문자셋이 다양한 인코딩을 가질 수 있습니다.

컴퓨터는 영미권에서 만들어졌기 때문에 영어를 표현하는 경우는 아무런 문제가 없지만 다른 문자를 사용하는 나라에서 자국의 언어로 표현하려면 문제가 되는 경우가 있습니다. 그

렇기 때문에 한글을 표현하는 인코딩을 적용하여 문서를 만들어야 합니다. 한글 인코딩은 MS949(윈도 OS에서 기본으로 사용), UTF-8 등이 있습니다.

7.

127.0.0.1

8.

server.xml

9. ①

## \_2장

1.

init(), doGet() 혹은 doPost(), destroy()

우선 객체가 생성되면서 init() 메소드가 단 한번 호출됩니다. init() 메소드에서는 주로 초기화 작업을 합니다. 그 후에는 클라이언트의 요청이 있을 때 doGet() 혹은 doPost() 메소드가 실행됩니다. 여러 클라이언트가 동시에 요청을 하더라도 스레드가 생성되므로 동시에 doGet() 혹은 doPost() 메소드가 실행되기 때문에 수행 속도가 빠릅니다.

서블릿이 더 이상 서비스를 하지 않을 경우 destroy() 메소드가 호출되는데 예를 들면 서블릿 컨테이너가 종료(톰캣을 재가동)되거나 서블릿 내용이 변경되어 재 컴파일해서 클래스 파일이 바뀌는 경우입니다.

2.

클라이언트에게 전송할 메시지 중 한글 깨짐을 방지하기 위한 방법

```
response.setContentType("text/html; charset=UTF-8");
```

혹은

```
response.setCharacterEncoding("UTF-8");
```

입력 폼에서 입력된 한글을 깨지지 않도록 하는 방법

get 방식일 경우,

서버 설정 파일인 server.xml 파일에서 첫 번째 등장하는 <Connector> 태그를 찾아서 URLEncoder 속성을 추가해야 합니다.

post 방식일 경우,

```
request.setCharacterEncoding("UTF-8");
```

3.

개발자가 서블릿 이름을 바꾸더라도 사용자는 동일한 URL로 접근할 수 있도록 하기 위함입니다.

4.

정답 : ③

해설 : Servlet의 멤버 변수는 Servlet의 객체 생성시 메모리 영역을 할당하므로 한 번 할당된 Servlet의 멤버 변수는 Servlet 객체가 메모리에서 해제될 때까지 유효합니다. 요청시마다 할당되는 것이 아닙니다.

5.

정답 : ③

해설 : Servlet은 PHP나 ASP와 달리 스레드(Thread)를 근간으로 하므로, 요청이 올 때마다 새로운 프로세스를 생성하지 않습니다. 이런 특징으로 인해 서버의 자원을 보다 효율적으로 운영할 수 있어 성능 향상을 이룰 있는 것입니다.

6.

정답 : ①

해설 : Javascript는 웹 애플리케이션의 클라이언트 측 자바 기술입니다. HTML의 정적인 페이지에 동적인 기능을 부여하는 기술입니다.

7.

정답 : ③

해설 : HTML의 통신 규약인 HTTP를 이용하여 사용자 요청을 처리합니다. 규약 중 Get 방식의 사용자 요청을 처리하기 위한 Servlet 메소드는 doGet() 메소드입니다.

8.

정답 : ②

해설 : Apache 웹 서버의 환경 설정 파일은 httpd.conf 입니다. server.xml은 Tomcat 서버의 환경 설정 파일이며 log4j.xml은 Log4J 툴의 환경 설정 파일입니다. build.xml은 Ant 툴의 환경 설정 파일입니다.

9.

정답 : ②

해설 : 여러 jsp 에서 사용되는 공통된 javascript는 .js 확장자로 묶어서 관리하는 것이 효

과적입니다.

10.

정답: ① - X, ② - O, ③ - X, ④ - O

11.

정답: ④

12.

정답: ① value, ② maxlength

13.

정답: ① <select>, ② multiple, ③ <option>, ④ selected

14.

정답: select, textarea

15.

정답: <http://www.test.com/test.jsp?userid=pinksung&pwd=1234&page=1>

16.

①

17. ① action, ② method

18.

④

19.

④

20.

정답 : ②

해설 : JSP는 요청시마다 Servlet으로 변환되는 것이 아니며 JSP가 변환되어 생성된 Servlet은 JSP가 수정될 때까지 한 번 변환된 Servlet이 계속 사용됩니다.

21.

정답 : ②

해설 : doGet() 메소드의 매개 변수 사양은 doGet(HttpServletRequest, HttpServletResponse)이다.

22.

정답 : ②

해설 : 공백은 + 문자로 전달된다.

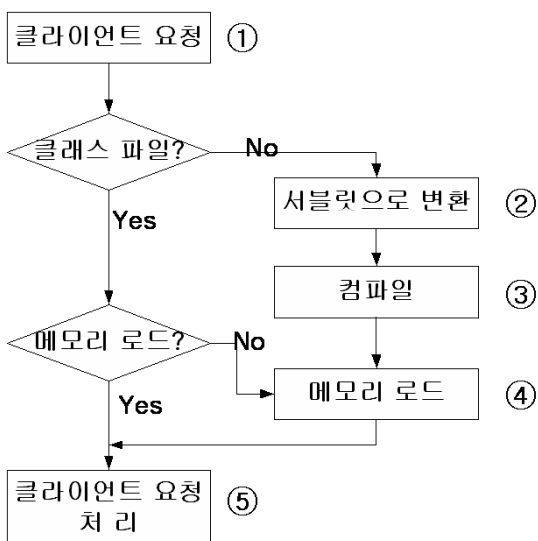
## \_3장

1. JSP는 Java Server Page의 약어이고, 인터넷을 사용하는 사용자에게 정보를 제공하기 위해서는 HTML 태그를 사용하여 웹 브라우저 내에 정보를 표시하는데 JSP에서는 이러한 HTML 태그를 사용할 수 있어 웹 애플리케이션의 프레젠테이션 역할을 합니다. JSP는 JSP에서만 사용가능한 다양한 태그(예를 들면 스크립트릿과 같은 것)를 이용하여 쉽게 웹 프로그래밍을 할 수 있도록 합니다.

2.

- ① 브라우저에서 JSP 페이지("myCount.jsp")를 웹 서버에게 요청합니다.
- ② 웹 서버는 JSP에 대한 요청을 서블릿 컨테이너에게 넘깁니다. 그러면 웹 컨테이너는 해당 JSP를 찾아서 서블릿 파일(myCount\_jsp.java)을 생성합니다.
- ③ 서블릿 파일은 컴파일되어 서블릿 클래스 파일(myCount\_jsp.class)을 생성합니다.
- ④ 컴파일된 서블릿 클래스 파일을 웹 서버로 보냅니다.
- ⑤ 사용자는 서블릿의 출력 결과인 HTML 형태로 응답 결과를 보게 됩니다.

3.



4.

정답 : ①

해설 : exception 이라는 내장객체 변수를 사용하기 위해서는 page 지시문의 isErrorPage

속성의 값을 true 로 설정하여야 합니다.

5.

정답:

|        |   |
|--------|---|
| Number | 1 |
| Number | 2 |
| Number | 3 |

6.

정답 : ③

해설 : ①은 JSP의 지시자(directive) 태그의 extends 속성에 대한 설명입니다. ②는 session 속성에 대한 설명입니다. ④은 **contentType** 속성에 대한 설명입니다. JSP에서 사용될 지역 변수 선언은 수행문 태그로 구현하여야 합니다.

7.

정답 : ②

해설 : import 속성은 두 번 이상 정의될 수 있습니다.

8.

정답 : ③

해설 : 선언문의 내용은 Servlet의 멤버가 되므로 \_jspService() 메서드 안에 들어가지 않는다.

9.

정답: ① - <%!, ② - %>, ③ - <%= multiply(10,25) %>

10.

④

11.

②

12.

①

13.

③

14.

①

15.

정답 : ②

해설 : JSP는 요청시마다 Servlet 으로 변환되는 것이 아니며 JSP가 변환되어 생성된 Servlet은 JSP가 수정될 때까지 한 번 변환될 Servlet이 계속 사용됩니다

16.

정답: ① - O, ② - O, ③ - X, ④ - O, 5) - O, 6) - X

## \_4장

1.

| 태그의 종류            | 설명   |
|-------------------|--|
| <jsp:forward>     | 다른 사이트로 이동할 때 사용한다.<br>페이지의 흐름을 제어할 때 사용한다.                            |
| <jsp:include>     | 정적 혹은 동적인 자원을 현재 페이지의 내용에 포함시킨다.<br>페이지를 모듈화할 때 사용한다.                  |
| <jsp:param>       | <jsp:forward>, <jsp:include>, <jsp:plugin> 과 같이 사용되어 전달인자를 추가할 때 사용한다. |
| <jsp:useBean>     | 빈(Beans)을 생성하고 사용하기 위한 환경을 정의하는 액션태그이다.                                |
| <jsp:setProperty> | 액션은 빈에서 속성 값을 할당한다.  |
| <jsp:getProperty> | 액션은 빈에서 속성 값을 얻어올 때 사용한다.  |

2.

request

웹 서비스를 위한 클라이언트와 웹 서버 사이의 요청에 관련된 정보는 request 객체에 저장되어 관리됩니다.

response

클라이언트에 대한 응답 처리를 하는 객체인 response는 실행결과를 브라우저로 되돌려 줄 때 사용하는 내장 객체입니다.

out

서버에서 클라이언트로 열려있는 출력 스트림을 의미합니다.

page

하나의 JSP 페이지를 처리할 때 해당 영역을 지시할 경우 사용합니다.

config

서블릿 관련 환경 설정과 관련된 객체입니다.

session

세션과 관련된 속성과 메소드를 제공하는 객체입니다.

application

하나의 웹 애플리케이션을 관리하고 웹 애플리케이션 안에서의 자원을 공유합니다.

pageContext

외부 환경 정보를 제공하는 객체입니다.

exception

예외 처리를 위한 객체입니다.

### 3.

| 내장 객체       | 패키지                | 클래스                 |
|-------------|--------------------|---------------------|
| request     | javax.servlet.http | HttpServletRequest  |
| response    | javax.servlet.http | HttpServletResponse |
| application | javax.servlet      | ServletContext      |
| out         | javax.servlet.jsp  | JspWriter           |

### 4.

① request

### 5.

②

### 6.

정답 : ④

해설 : page는 java.lang.Object 형을 참조하는 내장 객체 변수이다.

### 7.

정답: ① - X, ② - O, ③ - X, ④ - O, 5) - O, 6) - X

### 8.

②

해설 : 요청을 다른 자원으로 완전히 넘기고자 할 때 사용되는 액션 태그는 <jsp:forward>입니다.



9.

HttpRequest, HttpResponse, HttpSession, ServletContext

10.

③

11.

③

12.

①

13.

정답: ① request.getParameter("id"), ② request.getParameter("pw"), ③  
request.getParameter("desc")

14.

정답: ① request.getParameter("edu"); ② request.getParameter("na");  
③ request.getParameterValues("like"); ④ likes.length ⑤ <%=likes[i]%>

15.

정답: sendRedirect()

16.

②

## 5장

1.

쿠키를 사용하는 가장 주된 이유는 간단한 정보를 클라이언트에 저장함으로써 서버의 부하를 크게 줄일 수 있습니다. 쿠키는 웹 브라우저(클라이언트)의 정보를 웹 브라우저(클라이언트)에 저장하기 때문입니다.

쿠키의 문제점은 쿠키는 웹 브라우저가 거쳐 간 웹사이트 및 개인의 정보가 기록되기 때문에 개인의 사생활을 침해할 소지가 있고 보안 문제를 유발하기 때문에 웹 브라우저 자체에 쿠키 거부 기능이 있습니다. 쿠키 거부 기능이 브라우저에 설정되어 있으면 쿠키 본래의 목적인 브라우저와의 연결을 지속시키는 기능을 수행할 수 없게 됩니다. 이것이 쿠키의 가장

치명적인 단점입니다.

2.

| 비교 함수          | 쿠키                                 | 세션  |
|----------------|------------------------------------|---|
| 사용 클래스 및 인터페이스 | class<br>javax.servlet.http.Cookie | interface<br>javax.servlet.http.HttpSession |
| 관련 내장 객체       | response, request                  | session                                     |
| 저장 값 유형        | 문자열(String)만                       | Object                                      |
| 정보 크기          | 총 1.2M로 제한                         | 제한 없음                                       |
| 보안             | 취약함                                | 강력함   |
| 만료시점           | 쿠키 저장시 설정, 설정하지 않으면 브라우저 종료 시      | 클라이언트 로그아웃,<br>설정된 기간 동안 반응 없을 때            |

3.

HTTP 프로토콜은 웹 브라우저(클라이언트)의 요청에 대한 응답을 하고 나면 해당 클라이언트와의 연결을 지속하지 않습니다. 그래서 이전 작업이나 현재 작업 중인 것에 대한 정보를 가지고 있지 않습니다.

이런 부분을 해결하기 위해서 세션을 활용하면 웹 서버에 웹 브라우저의 정보를 저장해서 이후에 계속되는 웹 브라우저의 요청 시 해당 웹 브라우저의 정보와 서버에 저장되어 있는 각각의 웹 브라우저에 대한 정보를 비교해서 동일한 웹 브라우저로부터 온 요청을 판단할 수 있습니다.

로그인된 사용자나, 장바구니 등의 서비스를 제공하기 위해서 지속적인 상태 관리가 필요한데 상태 관리를 위해서 세션을 이용합니다.

4.

②

해설 Cookie는 저장하고자 하는 정보를 브라우저로 보내고, 이를 브라우저가 받아서 사용자 PC의 특정 영역에 저장합니다.

5.

③

해설 Cookie는 클라이언트의 고유한 정보를 브라우저가 받아서 PC의 특정 영역에 저장하게 됩니다.

6.

④ Cookie

해설 Cookie는 클라이언트의 고유한 정보를 브라우저가 받아서 PC의 특정 영역에 저장하게 됩니다. Session은 클라이언트의 고유한 정보를 서버의 메모리에 저장하게 됩니다.

7.

②

해설

HttpServletRequest는 Cookie를 추출할 때 HttpServletResponse는 Cookie를 설정할 때 사용합니다. Cookie에 대해서는 Cookie 객체가 사용됩니다. ServletContext는 Cookie 기술과 관련이 없습니다.

8.

③ setAttribute()

해설

HttpSession 객체에 일정시간 동안 유지하려는 정보를 등록합니다.

9.

① getSession()

해설 getSession() 는 HttpSession 객체를 리턴하는 기능으로 HttpServletRequest 에 존재한다.

10.

④

11.

③

12.

①

13.

response.addCookie

## **\_6장**

1.

정보 덩어리로 데이터 저장소라고 정의할 수 있습니다. 프로그램에서 사용되는 정보가 여러 개라면 이를 변수에 저장하고 필요할 때마다 개별적으로 접근해서 사용하기보다는 자바 빈을 사용하면 필요한 정보를 객체를 구성하는 멤버로 기술해 두고 한꺼번에 데이터를 접근해서 사용할 수 있습니다.

2.

```
member.getName();
```

3.

```
member.setName("전수빈");
```

4.

④

해설 : <jsp:include> 태그는 다른 프로그램(JSP, Servlet)의 수행 결과를 포함하는 기능의 태그이다.

5.

④

해설 : scope 속성은 아래와 같은 적용 범위를 가집니다. page < request < session < application ServletContext는 웹 어플리케이션들이 공통적으로 관리하고 참조할 수 있는 정보를 제공합니다.

6.

①

해설: name 속성은 <jsp:setProperty>, <jsp:getProperty> 사용 시에 접근하고자 하는 JavaBeans 객체의 id 명을 지정하는 용도로 사용된다.

7.

③

해설 : application은 서버가 기동 되어 있는 동안 유효한 JavaBeans 객체를 생성하며 page는 JSP 가 수행되는 동안 유효한 JavaBeans 객체를 생성합니다. 그리고 session은 생성된 HttpSession 객체가 유지되는 동안 유효한 JavaBeans 객체를 생성합니다.

8.

①

해설 : <jsp:useBean> 액션 태그를 사용하는 두 개의 JSP 간에, 생성되는 JavaBeans 객체를 공유하려면 반드시 id 속성의 값도 동일해야 합니다.

## 7장

1. 표현 언어는 값(데이터)을 웹 페이지에 표시(표현)하는 데 사용되는 태그이다.

2. JSTL 태그 라이브러리는 JSP는 스크립트릿과 자바 코드 등 여러 코드들이 섞여서 복잡한 구조로 되어있는데, 간결하고 이해하기 쉽게 코딩을 하기 위해 추가한 태그를 말한다.

3. 보다 간단하고 이해하기 쉽게 JSP 페이지를 작성할 수 있다.

4.

② `${paramValues.season[0]}`

5.

② `${list[3]}`, ④ `${list[index+ 2]}`

6.

`sung yoonjung`

7.

③ `<c:forEach>`

8.

`<select name="season">`

`<c:forEach items="${list }" var="item">`

`<option value="${item}"> ${item}`

`</option>`

`</c:forEach>`

`</select>`

## 8장

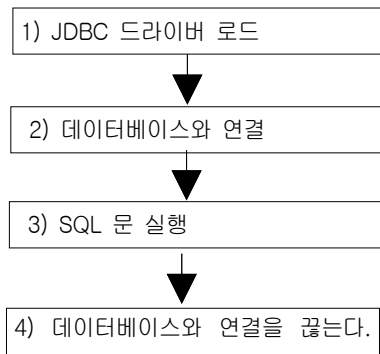
1.

DBMS는 DataBase Management System의 약어로 오라클이나 MS SQL Server와 같은 데이터베이스 관리 시스템을 말한다.

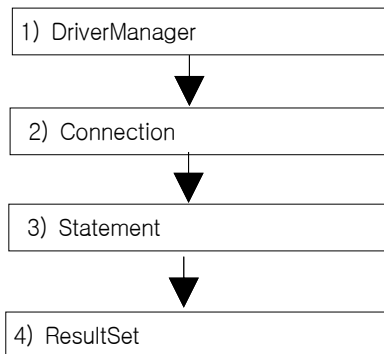
2.

JDBC는 Java Database Connectivity의 약어로 자바 프로그램에서 데이터베이스에 일관된 방식으로 접근할 수 있도록 API를 제공하는 클래스의 집합을 말합니다.

3.



#### 4.



#### 5.

Connection :데이터베이스와 연결과 관련된 인터페이스

Statement :질의, 갱신 실행과 관련된 인터페이스

ResultSet 결과물을 가져오는 인터페이스

#### 6.

JDBC 드라이버에 대한 이해를 돕기 위해서 프린터기를 예를 들어보겠습니다. 여러분이 프린터를 구입했다고 합시다. 이 프린터를 사용하려면 드라이버를 설치해야 합니다. 드라이버는 컴퓨터와 프린터 사이의 의사소통을 위한 프로그램입니다. 우리가 사용하려는 JDBC 드라이버도 자바와 오라클이라는 데이터베이스 사이의 의사소통을 위한 프로그램으로 이를 자바에 환경 설정해야만 오라클을 자바에서 사용할 수 있게 됩니다.

#### 7.

PreparedStatement는 Statement 객체로 쿼리문을 작성할 경우 복잡해지는 쿼리문을 보다 간단하게 작성할 수 있도록 하기 위해서 제공되는 객체이다.

#### 8.

④

9.

②

10.

정답: JDBC(Java DataBase Connectivity)

11.

① "oracle.jdbc.driver.OracleDriver"

② url

③ uid

④ pwd

⑤ pstmt=conn.prepareStatement(sql);

⑥ pstmt.setString(1, id);

⑦ pstmt.setString(2, passwd);

⑧ pstmt.setString(3, name);

⑨ pstmt.executeUpdate();

⑩ conn.close();

12.

①

해설 : javax.sql 패키지는 JDBC의 확장 API로 서버측의 커넥션 풀링, 분산 트랜잭션 처리 등을 위한 DataSource /ConnectionPool/RowSet 등과 관련된 API가 추가되어 있습니다. JDK 1.4버전부터는 기본적으로 포함되어 있으나, 이전버전에는 없으므로 별도로 다운받아서 설치해야 합니다.

13.

①

해설 : Java DataBase 프로그램을 작성할 때 데이터베이스와 연결을 담당하는 것은 JDBC 드라이버입니다. JDBC 드라이버는 데이터베이스 업체에서 제공되고 있습니다.

14.

④

15.

②

16.

④

17.

④

## \_9장

1. dbcp

2. ③

3. ①

4.

```
<Resource name="jdbc/myoracle" auth="Container" type="javax.sql.DataSource"
    driverClassName="oracle.jdbc.OracleDriver"
    url="jdbc:oracle:thin:@127.0.0.1:1521:ORCL"
    username="hr" password="hr" maxActive="20" maxIdle="10"
    maxWait="-1" />
```

5.

5-1 List<BoardVO> getBoardList()

5-2 void insertBoard(BoardVO)

6. 접속인원이 많은 웹 페이지에서 데이터베이스의 효율성과 속도를 높이기 위해서 사용합니다.

7.

```
Context initContext = new InitialContext();
Context envContext = (Context)initContext.lookup("java:/comp/env");
DataSource ds = (DataSource)envContext.lookup("jdbc/mydb");
Connection conn = ds.getConnection();
```

## \_10장



1.

① "post" ② "multipart/form-data" ③ "file"

2.

com.oreilly.servlet.MultipartRequest

3.

getParameterNames()

## **\_11장**

1.

모델(Model)

2.

뷰(View)

3.

컨트롤러(Controller)