JSP 학습 필기용

* 1. **웹 프로그램이란 ?**

브라우저에서 다른 컴퓨터로 정보를 요청 (Request)하면 서버는 응답(Response)를 한다.

즉, 인터넷 서비스를 이용해서 서로 다른 구성요소들(PC 등)이 통신할 수 있는 프로그램이 웹 프로그램이다.

* 1. **프로토콜(Protocol)과 IP**

프로토콜 : 통신을 하기위한 규약 ex. HTTP, FTP, SMTP, POP 등

**HTTP** : 브라우저에서, 웹 사이트에 접속을 하면 하이퍼링크를 통해 그림, 글 등의 정보를 제공하는 규약

FTP(File Transfer Protocol) : 파일 자체를 전송하는 규약

IP 주소 : 숫자로 표현된 인터넷의 특정한 주소

DNS : IP주소를 매핑한 이름 주소( naver.com , google.com 등)

Port : 웹 서버의 여러 프로그램 중에 어떤 프로그램으로 들어갈 지 정하는 것. (기본 포트 80)

Index.html : 특정 웹 사이트 접속 시 가장 기본적으로 실행되는 경로

* 1. **웹 프로그래밍의 동작 원리**

유저가 웹 서버로 요청을 보낸다.

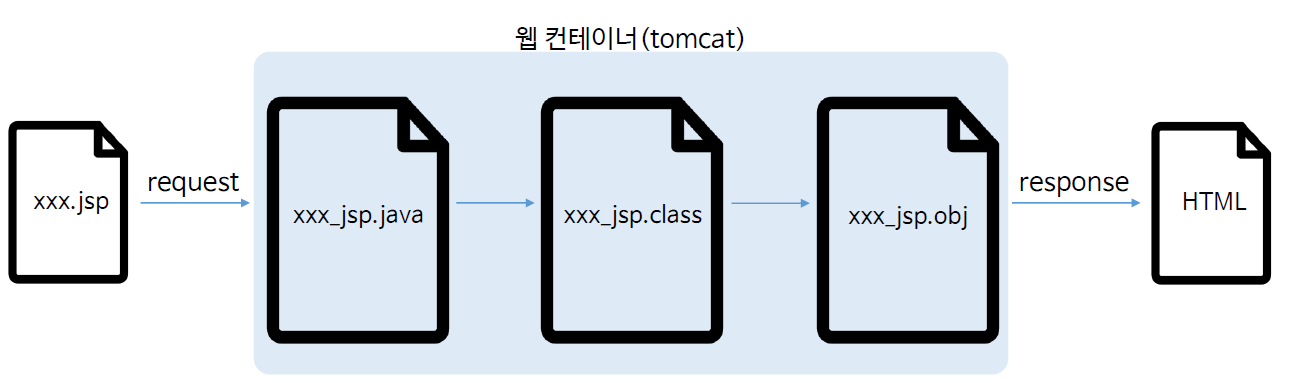
서버는 요청을 판단해서

1. 데이터 베이스에서 query를 날려서 result를 얻은 후 유저에게 response를 보낸다.
2. 정적인 데이터인 html로 응답한다.
3. 동적인 데이터인 웹 컨테이너로 데이터를 가공해, 새로운 데이터를 뽑아낸 후 Html로 응답한다..

**3.1 웹 컨테이너 구조**

WAS : Web Application Server 속에 Web Container가 있다.

개발자가 만든 jsp 파일을 Web Container 서버가 jsp.java로 변환, 컴파일해서 .class파일로 변환, .obj파일로 변환해서 그 결과가 html로 응답된다.



**3.2 JSP 파일 작성**

Jsp 파일은 Html 파일과 형식이 같고 호환이 된다.

**4.1**

Servlet : jsp와 달리 순수 java 파일.

Servlet 파일 내에서 html 태그를 이용해서 html 출력이 가능하다.

**5.1 servlet mapping**

Jsp : view 부분을 주로 개발

Servlet : model 부분 개발

Servlet 맵핑이란 : 브라우저의 요청에 서버에서 적당한 servlet을 골라야 한다.

Full path : Servlet의 정확한 경로, 보안에 취약하고 URL이 복잡하다.

Mapping path : context path는 그대로 쓰고 package와 servlet 이름을 간단한 닉네임으로 변경하는 것

**5.2 맵핑하는 방법 (1) : web.xml 파일을 이용한 맵핑**

Web.xml에서 아래 코드를 써준다.

Servlet을 먼저 선언하고 정확한 위치를 package를 포함해서 적어준다.

그 후 mapping tag에서 맵핑할 서블릿 이름과 맵핑할 url 주소를 적어준다.

<servlet>

<servlet-name>servletEx</servlet-name>

<servlet-class>com.servlet.ServletEx</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>servletEx</servlet-name>

<url-pattern>/SE</url-pattern>

</servlet-mapping>

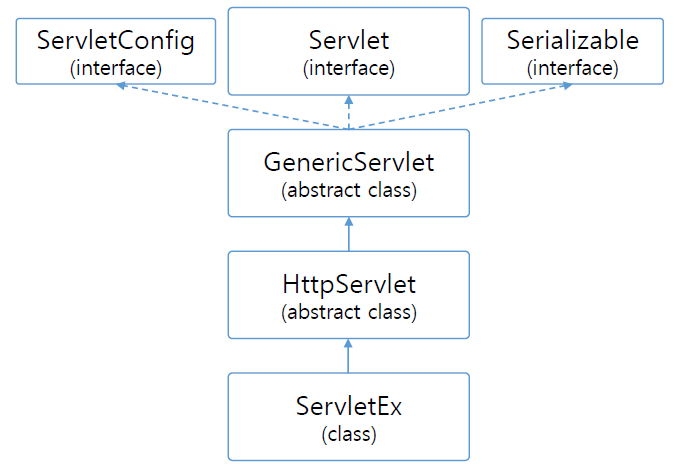
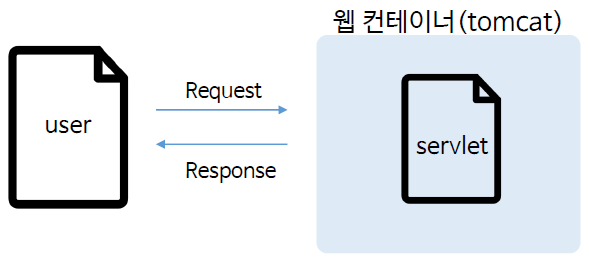
**5.2 맵핑하는 방법 (2) : 어노테이션 사용**

@WebServlet("/SE")

코드를 servlet 파일의 class 위에 적어준다.

Web.xml 사용하는 방법보다 어노테이션이 더 최신이고 잘 사용하지만 둘 다 알아두자.

**6.1 HttpServlet**



사용자가 요청하는 객체를 Request, 서버가 응답하는 객체를 Response 객체라고 한다.

HttpServlet : 추상 클래스로, 개발하는 servlet 코드에서 상속받아서 구현

HttpServlet은 GenericServlet을 상속받고 또 여러 인터페이스가 상속되지만, 개발자는 HttpServlet만 상속해서 기능을 구현하면 된다.

**6.2 HttpServletRequest**

사용자가 서버측에 요청을 보낼 때 사용하는 객체

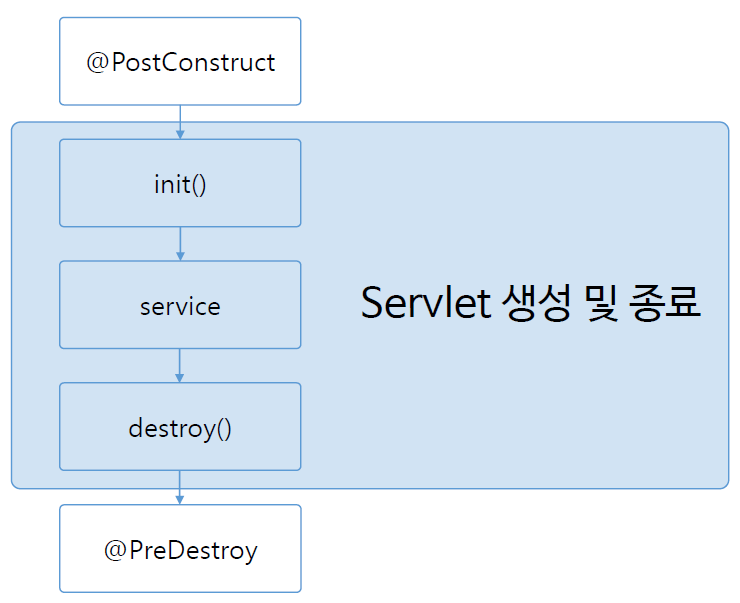
Request.addCookie, getSession, getAttribute 등의 기능이 있다.

**6.3 HttpServletResponse**

서버가 사용자측에 응답을 보낼 때 사용하는 객체

Response.addCookie, getStatus, sendRedirect 등이 있다.

**7.1 Servlet Life Cycle**



@PostConstruct : Servlet의 준비 단계

Init() : servlet의 생성 단계

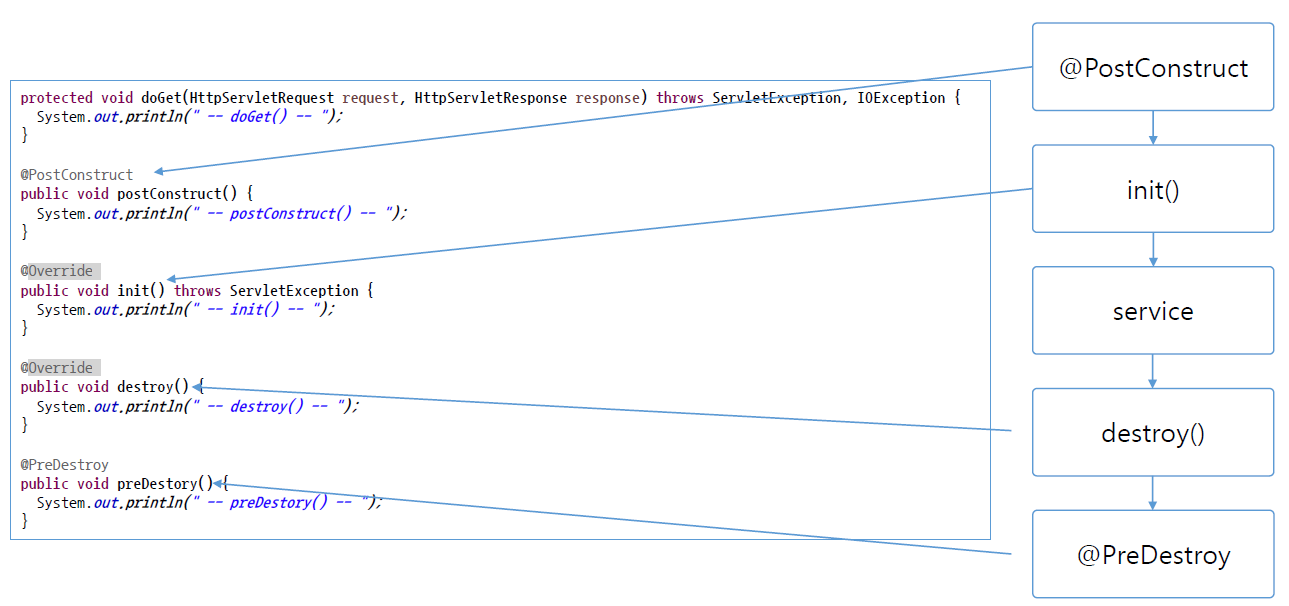
Service : servlet의 기능을 하는 것, 코드에서 doGet 부분이다.

Destroy() : servlet이 소멸되는 단계

@PreDestroy : Servlet의 종료 후 정리하는 단계

Container인 톰캣 서버가 알아서 적절한 시기에 각 단계를 실행시켜준다.

**7.2 생명주기 관련 메서드**



**8.1 form 태그**

form 태그 : 브라우저에서 서버로 데이터를 전송하는 태그. Request 객체를 통해 Data를 서버로 전송하게 된다.

**8.2 doGet 방식**

form태그에서 method=”get”일 때, 서버에서 doGet() 으로 데이터를 받는다.

default가 get방식이다.

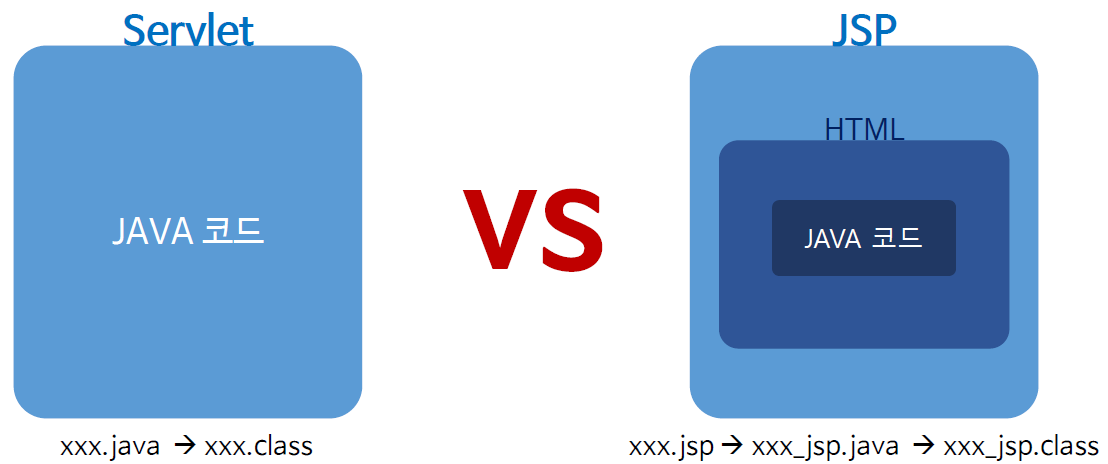
form 태그에 입력한 데이터가 웹 브라우저 URL에 노출되어 보안에 약하다.

URL 길이가 한계가 있어서 많은 데이터 전송에 무리가 있다.

**8.3 doPost 방식**

form 태그에서 method=”post” 방식일 때, 서버에서 doPost()로 데이터를 받는다.

URL에 맵핑 정보만 보이기 때문에 보안에 더 강하다.

**9.1 Servlet vs Jsp**

Servlet : 순수 자바 파일

JSP : HTML + 자바 코드

**9.3 JSP 주요 스크립트**

1) 선언 태그 : <%! %> : JSP 페이지에서 Java 멤버 변수 또는 메소드를 선언할 수 있다.

2) 주석 태그 : <!-- 주석태그 --> HTML 주석

<%-- 주석태그 --%> JSP 주석

3) 스트립트릿 태그 : <% %> : JSP 페이지에서 Java 코드를 넣기 위한 태그

4) 표현식 태그 : <%= 변수 or 메소드 %> : Java의 변수 및 메서드의 반환값을 출력하는 태그

크롬에서 개발자 도구로 소스 코드를 확인하면 HTML 코드를 제외한 JSP 태그, 코드들은 다 사라진다.

지시어 : 서버에서 JSP 페이지를 처리하는 방법에 대한 정의

1. Page : 페이지 기본 설정 <%@ page 속성 = “속성 값”%>

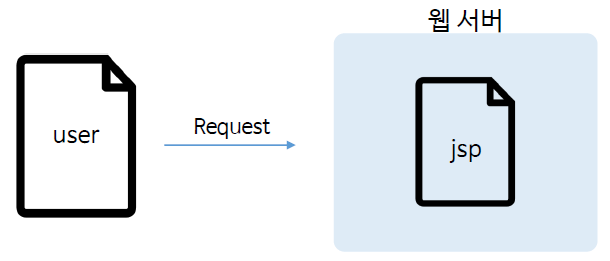
보통 고정되어 있어 타이핑 할 일은 적다.

1. Include : include file 설정 <%@ include file = “파일명”%>

Header나 footer처럼 반복되는 파일을 포함시킬 때 유용하다.

1. taglib : 외부 라이브러리 태그 설정 <%@ taglib uri=”<http://~~~>” prefix=”네임스페이스명”%>

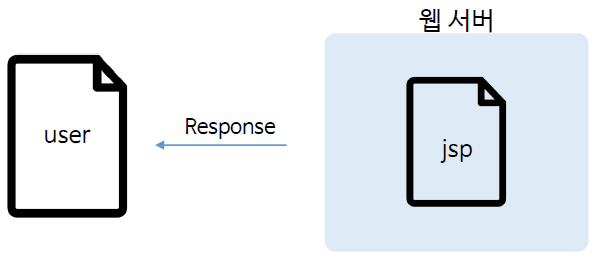
**10.1 requeset 객체**



주로 사용자가 HTML에서 <form> 태그를 이용해 전달한 정보를 웹 서버에서 jsp를 통해 처리

request.getParameter를 주로 이용한다.

**10.2 response 객체**



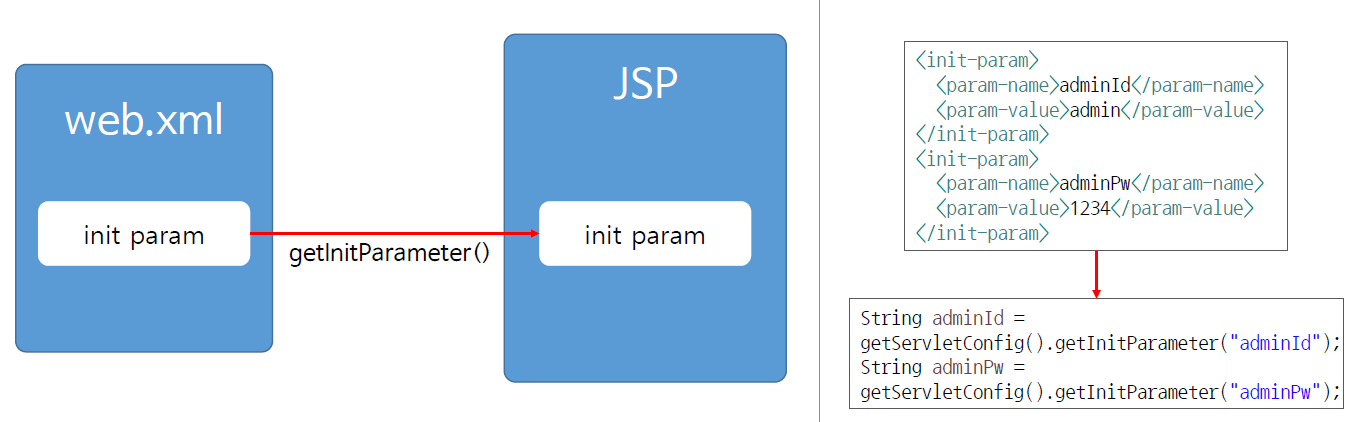
서버에서 사용자에게 응답한다.

response.sendRedirect(“xxx.jsp”)로 다른 페이지를 보여주는 기능 등이 있다.

**11장 jsp 내장 객체**

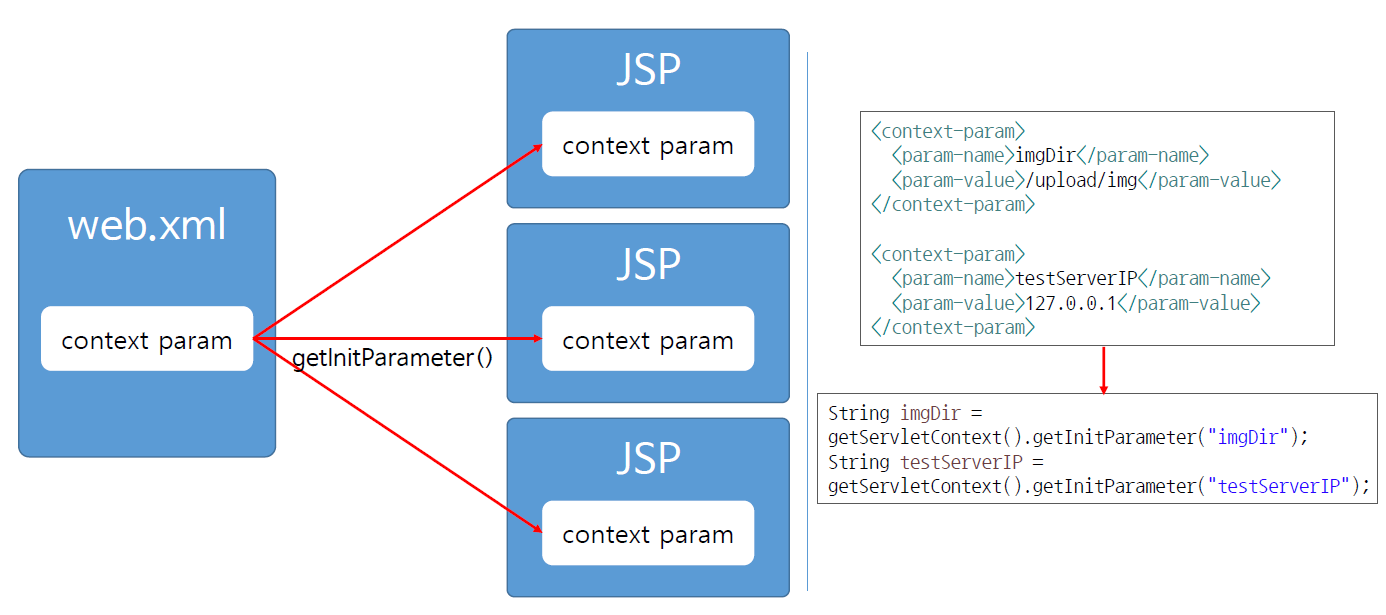
**11.1 config 객체**

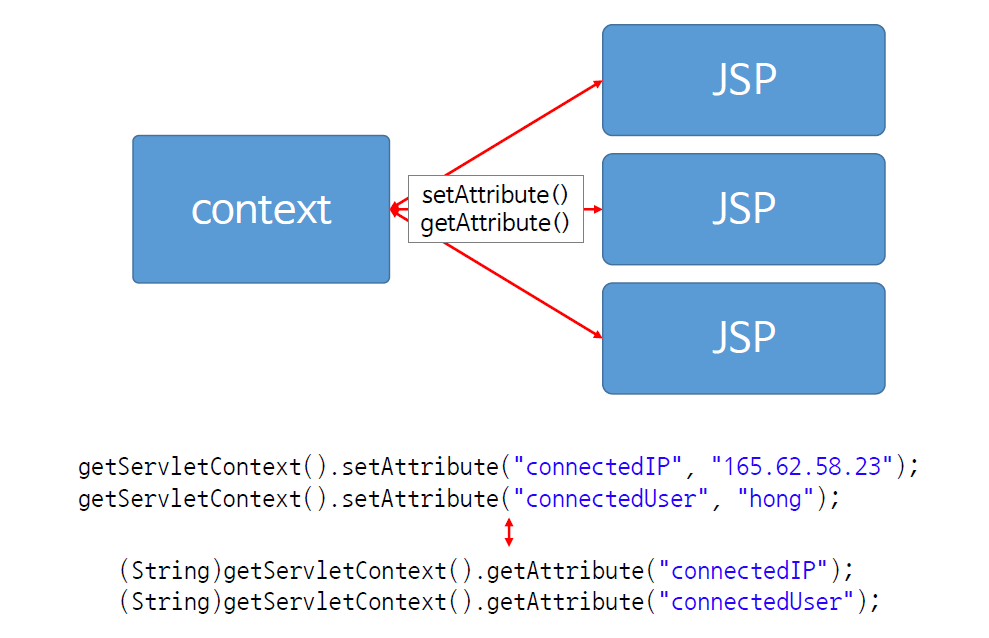
web.xml에 데이터를 저장해둔 후 jsp 파일에서 가져와서 사용하는 객체



**11.2 application 객체**

어플리케이션 전체에 데이터를 공유하기 위해 web.xml에 context param으로 정보를 저장해두는 것





Jsp 파일에서 get, setAttribute를 통해 새로 정보를 저장하고 꺼내올 수 있다.

getAttribute를 쓸 땐 필수적으로 String으로 변환해주어야 하는 것을 주의하자.

**11.3 out 객체**

out.print(“ ~ “)를 통해 HTML 태그를 Jsp에서 사용할 수 있다.

**11.4 exception 객체**

에러 발생 시 에러 처리를 해주는 객체

<%@ page errorPage = *"errorPage.jsp"* %> 를 추가해줘야 한다.

Error 처리를 하는 파일에서는

<%@ page isErrorPage=*"true"* %> 를 추가해준다.

**12 servlet 데이터 공유**

**12.1 servlet parameter**

해당 servlet에서만 데이터를 공유하는 방법

web.xml에서 데이터 저장 후 init parameter를 이용해서 사용. jsp의 config 객체와의 차이는 jsp에 있는지 servlet에 있는지 밖에 없다.

getServletConfig().getInitParameter(“변수명”); 을 통해 값을 가져온다.

**12.2 context parameter**

여러 servlet에서 데이터를 공유하는 방법

**12.3 context attribute**

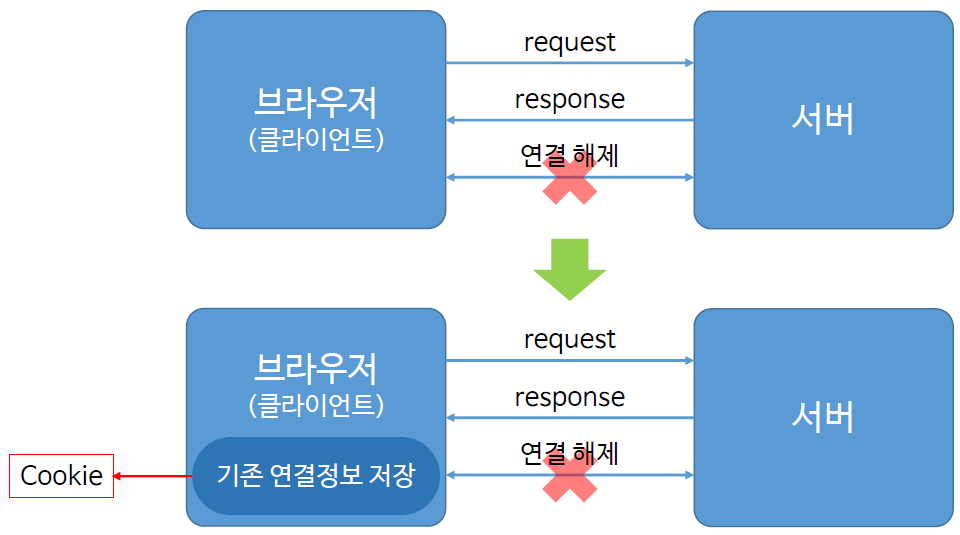
getServletContext().setAttribute(“변수명”,”값”) 을 통해 데이터를 저장하고.

마찬가지로 getServletConfig().getAttribute(“변수명”)를 할 땐 String으로 형변환을 해줘야 한다.

**13장. Cookie**

**13.1 Cookie란?**

쿠키를 먹다가 가루를 흘리는 것처럼, 서버에 접속했던 흔적을 남기는 것

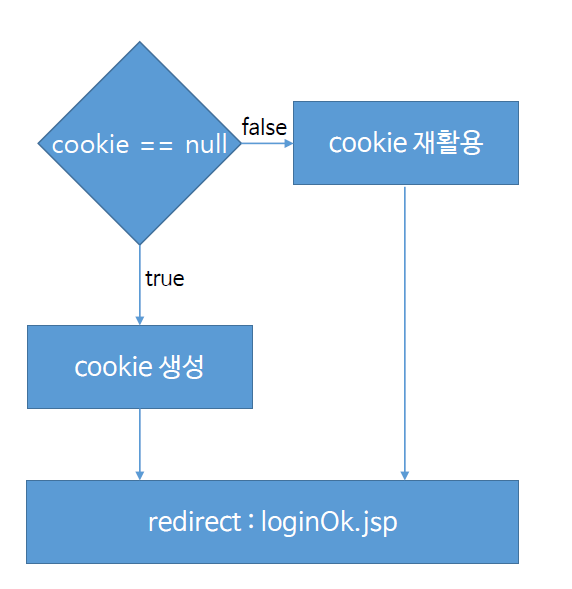


요청이 오고 응답을 하면 리소스를 낭비하지 않기 위해 바로 연결을 해제한다.

Cookie를 이용해 클라이언트와 서버의 데이터를 연결하는 고리로 사용한다.

클라이언트의 브라우저에 Cookie, 연결 정보를 저장한다.

**13.2 Cookie 구현**



Cookie는 정보가 여려 개이기 때문에 배열로 관리한다.

접속 이력이 있으면 그 정보를 이용하고 NULL이면 새로 생성한다.

사용자의 브라우저에 저장되기 때문에 보안에 약할 수 있다.

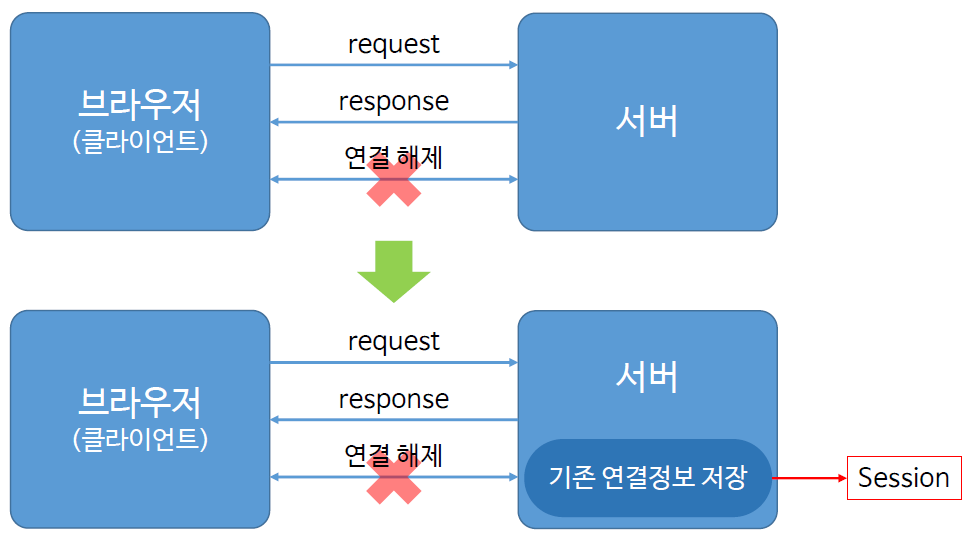
중요한 정보는 서버에 저장하는 게 좋다.

**14 Session**

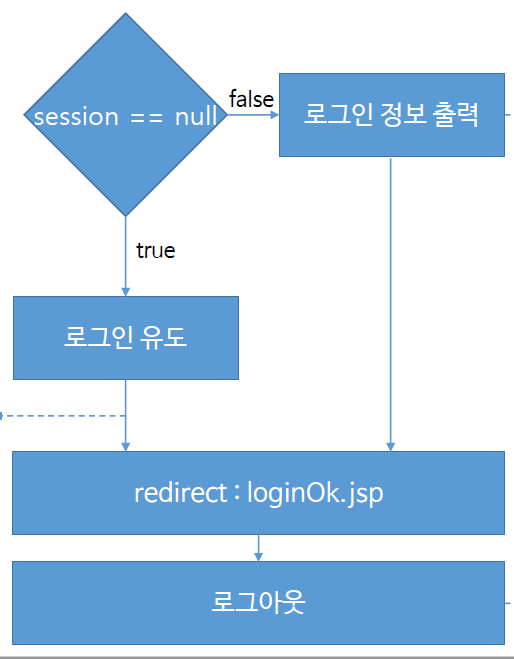
**14.1 Session 이란?**

클라이언트 브라우저에 저장하는 Cookie와 달리

서버(웹 컨테이너)에 연결정보를 저장해서 연결 정보를 유지시켜주는 것



**14.2 Session 구현**

session.invalidate()로 세션을 날릴 수 있다.

**15 한글처리**

**15.1 한글 처리**

한글을 깨지지 않고 보이게 하는 처리.

get 방식과 post 방식일 때 차이가 있다.

get 방식 : server.xml에 <Connector URIEncoding=”UTF-8”/> 을 추가해준다.

post 방식 : 서블릿, jsp에 request.setCharacterEncoding(“UTF-8”);을 추가해준다.

**15.2 Filter**

브라우저와 웹 서버가 통신을 할 때 filter로 걸러서 정보를 보내준다.

서블릿마다 URIEncoding 방식을 지정해주면 번거롭고 지저분하므로, filter를 이용해 간편하게 사용한다.

Filter라는 인터페이스를 오버라이딩해서 init, doFilter, destroy를 재정의 해줘야 한다.

재정의한 필터를 web.xml에서 filter를 적용시켜준다.

**17 SQL**

**17.1 테이블 생성 및 삭제**

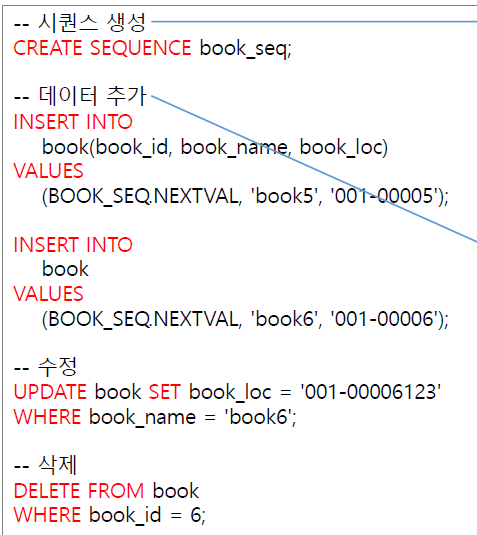
create Table book(

book\_id NUMBER(4),

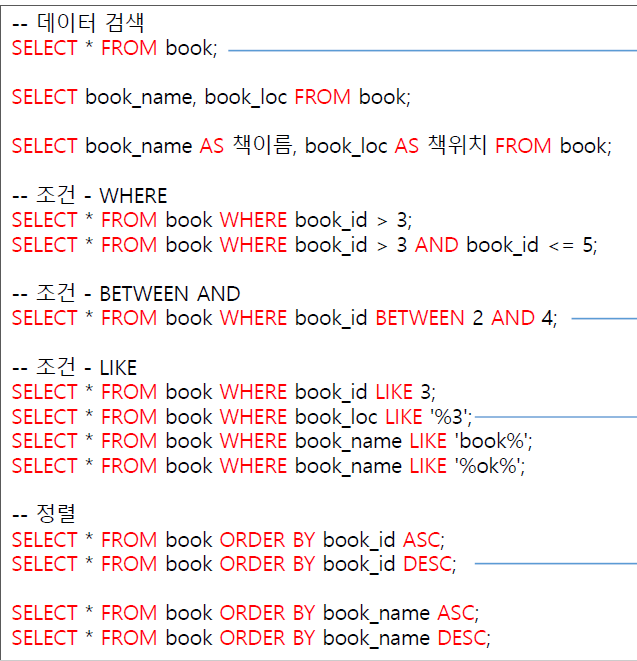
book\_name VARCHAR2(20)

);

**17.2 데이터 추가, 수정, 삭제**



**17.3 데이터 검색**



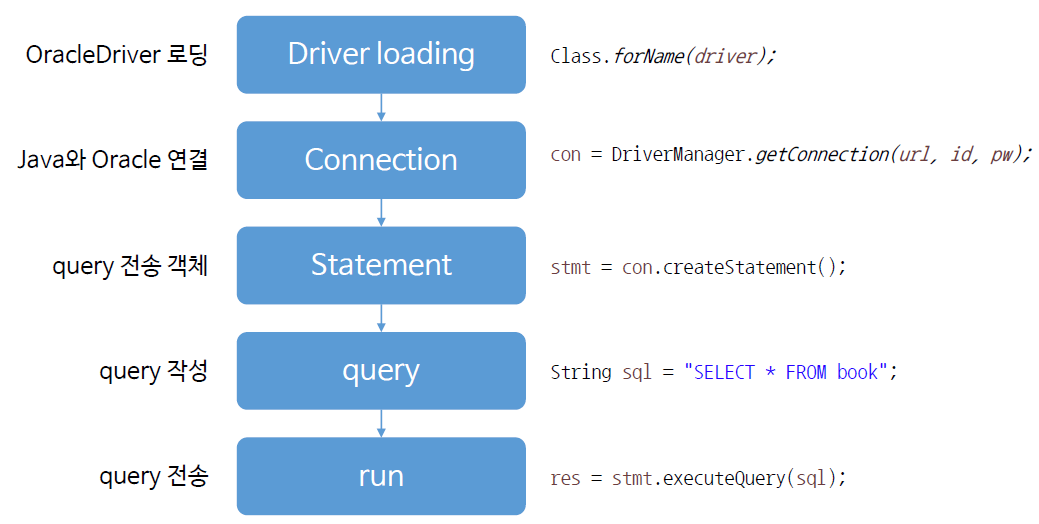
**18 JDBC**

**18.1 jdbc 설정**

jdbc : 자바와 Database가 통신할 수 있게 해주는 API

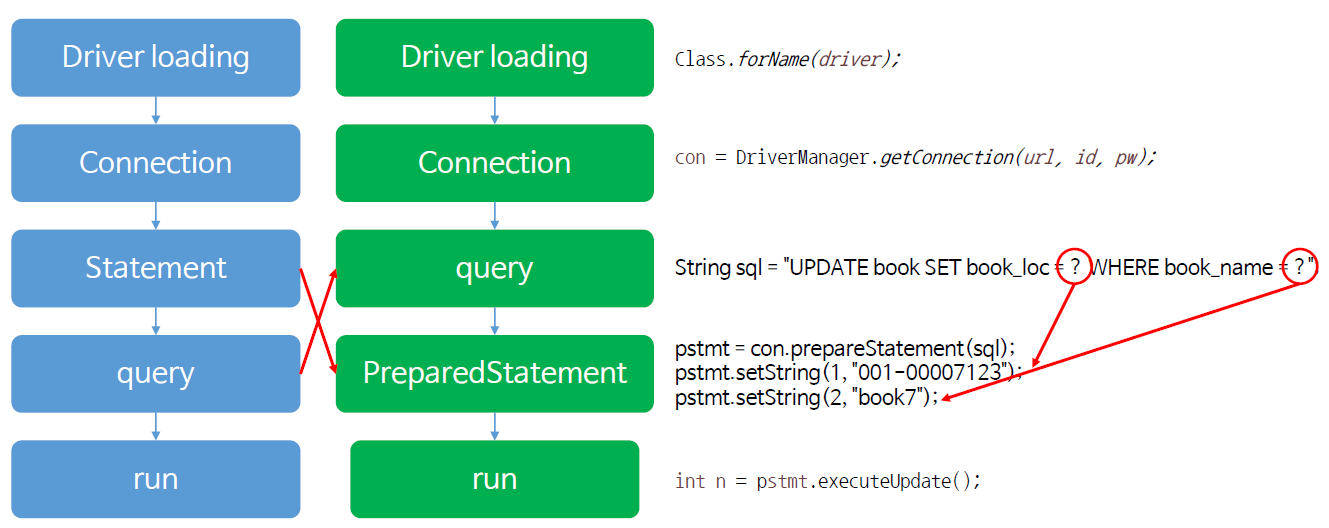
oracle library를 이클립스에 설정해둬야 통신이 가능하다.

**18.2 jdbc를 이용한 데이터 관리**



**18.3 PreparedStatement**

sql statement문을 더 간단하게 하기 위한 것.

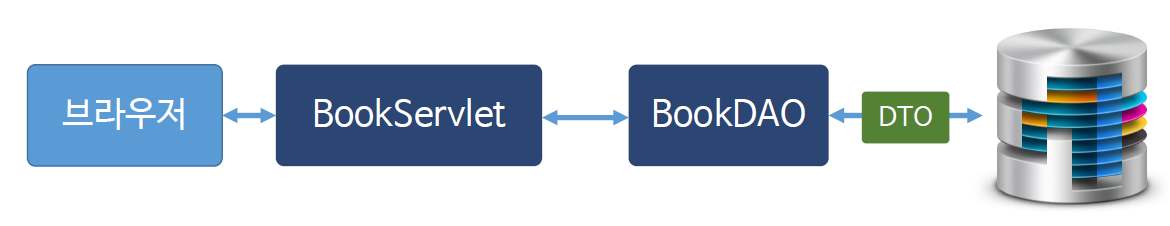


**19 DAO와 DTO**

**19.1 DAO와 DTO란?**

DAO : data access object, 데이터베이스에 접근하는 객체

DTO : data transfer object, 데이터베이스에 있는 데이터의 형태를 자바의 형태로 변환시켜주는 객체



**19.2 DAO, DTO 구현**

servlet에서 db에 접속하고 값을 가져오던 걸 dao와 dto에 나눠서 실행한다.

**20 Connection Pool**

**20.1 커넥션 풀이란**

db와 연결하는 연결고리가 풀에 담겨있는 것

기존 방법은 db에 매번 connection을 열어서 질의를 한 후 close를 해야하는데 이 과정에서 자원을 많이 사용하기 때문에, 과부하가 일어날 수 있다.

이를 해결하기 위해, 서버가 한가할 때 connection을 미리 만들어 놓은 후, 접근이 필요할 때 마다 기능을 사용하고 반환한다. 좀 더 안정적으로 수행할 수 있다. tomcat이 connection pool을 설정해둔다.

