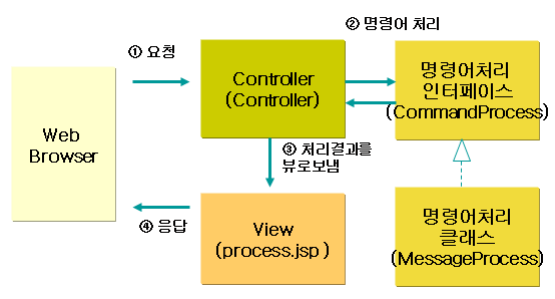


MVC 패턴(Model-View-Controller pattern): MVC(Model-View-Controller) 구조는 전통적인 GUI(Graphic User interface) 기반의 어플리케이션을 구현하기 위한 디자인 패턴이다. MVC 구조는 사용자의 입력을 받아서, 그 입력 혹은 이벤트에 대한 처리를 하고, 그 결과를 다시 사용자에게 표시하기 위한 최적화된 설계를 제시한다.



+ mvc(model view control)

MVC Pattern

- JSP (디자인, 로직, 흐름) : 한 페이지에 들어있음 (Model1 방식)

- 프로젝트가 엔터프라이즈 급으로 커진다면?

1. 화면 : JSP ==> View

2. DB 연결, 데이터 조작(연산) : JAVA ==> Model

3. 데이터 흐름 : Servlet ==> Control

- 이런 방식을 MVC Pattern 또는 Model2 방식

- 유지, 보수가 훨씬 용이

- 개발은 더 번거롭다.

+ Servlet : 웹에서 동작하는 작은 자바 프로그램

서블릿 생성 방법

1. HttpServlet 상속

2. do~ method overide => 필요한 로직을 작성

- doGET() : Request-URI로 지정한 자원에 대해 요청 (default)

- doPOST() : Request BODY에 데이터를 담아서 서버로 전송하면서 요청

- doHEAD() : GET 방식과 동일하나 서버의 응답시 Response Header만을 전송

- doPUT() : Request BODY의 내용을 Request-URI로 지정한 곳에 저장

- doDELETE() : Request-URI로 지정한 자원을 삭제하도록 요청

- doOPTIONS() : 서버가 제공하는 통신옵션에 대한 정보를 요청

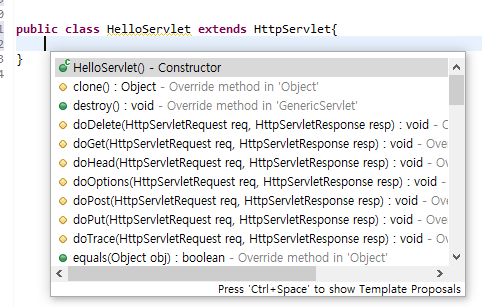
- doTrace() : 요청 메세지가 거쳐가는 경로에 대한 정보를 요청

3. WebServlet annotation => 주소를 매칭

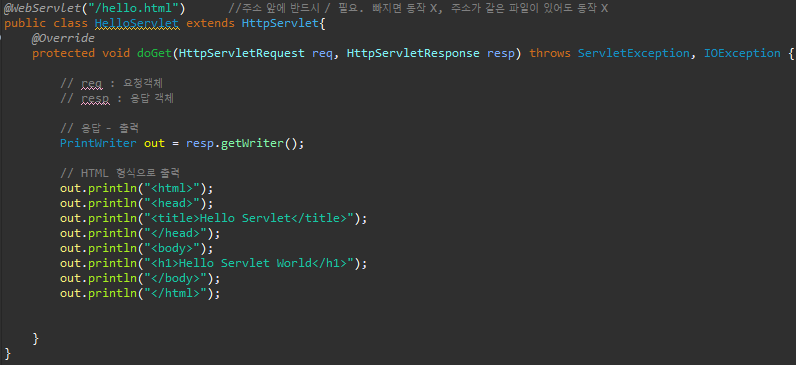
Control 단 – Servlet 으로 만듬

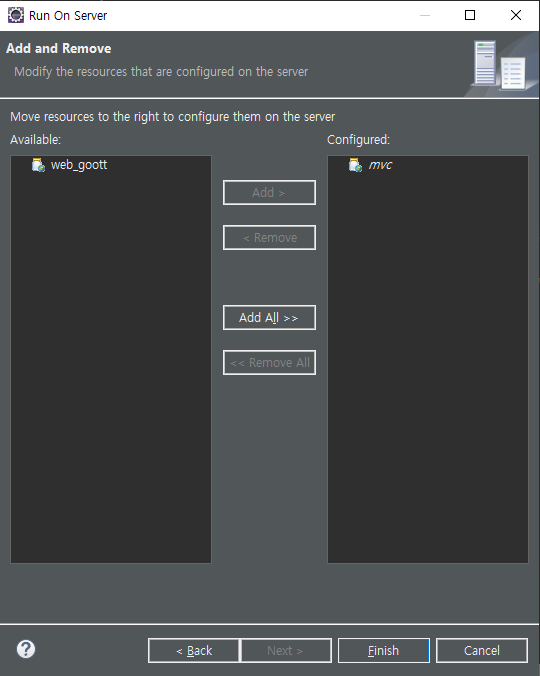
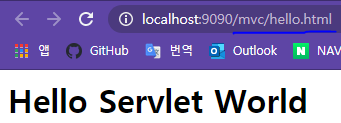
Servlet: 웹에서 동작하는 작은 자바프로그램, 흐름제어

Servlet 객체를 만들어서 사용하려면 상속받아서 쓰면 된다.



요청이라는 기능을 할 때는 주소가 필요



구동시킬 때 톰캣이 다 읽게 하지말고 지금 하고있는 녀석만 실행되도록 함

주소 노출을 막을 수 있음

1. HiServlet..java
2. 가상주소 <http://localhost:9090/mvc/hi.html>
3. 웹 브라우저에

h1 – 내가 만들어본 두번째 서블릿 파일

한글이 깨지는 현상 발생, 요청이 POST인 경우 인코딩 세팅 필요

@WebServlet("/hello")

public class FirstServlet extends HttpServlet {

    @Override

    public void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

**req.setCharacterEncoding("UTF-8");**

**resp.setContentType("text/html; charset=UTF-8");**

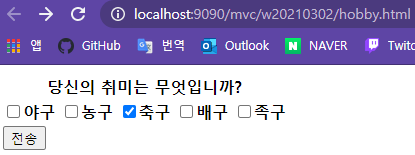
        ...req.getParameter() 와 resp.getWriter() 사용 전 셋팅....

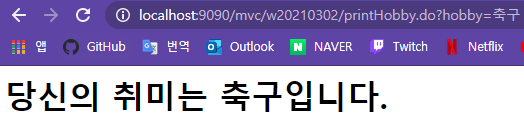
    }

hobby.html printHobby.do로 get방식으로 데이터 전송

Hobby.java *@WebServlet*("/w20210302/printHobby.do") 로 자신이

printHobby.do임을 인식시킴, 주소 매칭 확인!!!

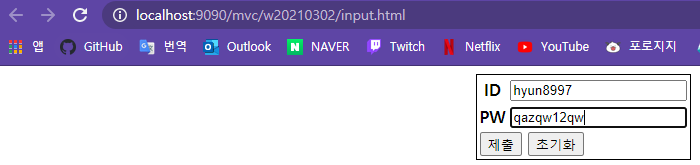


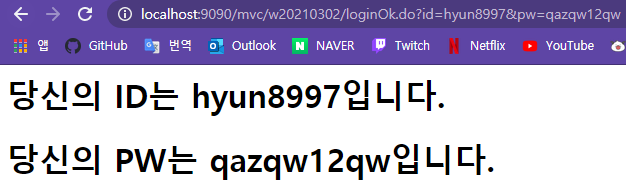


Input.html 로그인 폼

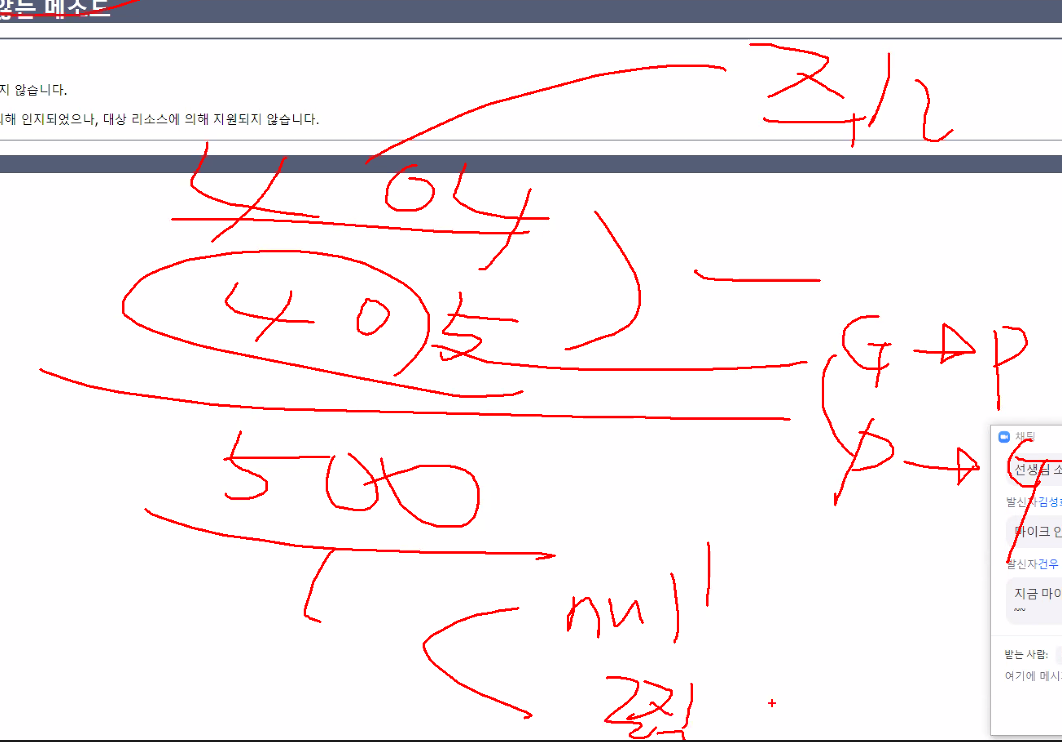
Login.java 로그인 폼에서 ID, PW를 받아서 출력

get방식, post방식 (doGet, doPost)





자주 볼 오류코드



https://tomcat.apache.org/download-taglibs.cgi

jar files 다운로드

[taglibs-standard-impl-1.2.5.jar](https://downloads.apache.org/tomcat/taglibs/taglibs-standard-1.2.5/taglibs-standard-impl-1.2.5.jar)

[taglibs-standard-spec-1.2.5.jar](https://downloads.apache.org/tomcat/taglibs/taglibs-standard-1.2.5/taglibs-standard-spec-1.2.5.jar)

[taglibs-standard-jstlel-1.2.5.jar](https://downloads.apache.org/tomcat/taglibs/taglibs-standard-1.2.5/taglibs-standard-jstlel-1.2.5.jar)

[taglibs-standard-compat-1.2.5.jar](https://downloads.apache.org/tomcat/taglibs/taglibs-standard-1.2.5/taglibs-standard-compat-1.2.5.jar)

데이터 이동을 어떻게 더 편하게 할 건지

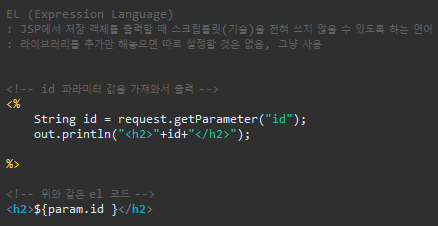
A -> B 로 데이터를 보내 볼 것

loginForm.jsp el.jsp로 데이터를 전송

el.jsp EL (Expression Language): JSP에서 저장 객체를 출력할 때

스크립틀릿(기술)을 전혀 쓰지 않을 수 있도록 하는 언어,

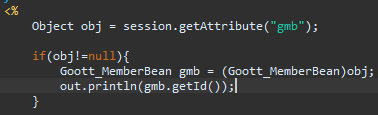
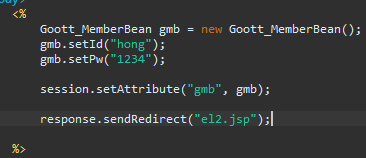
라이브러리를 추가만 해놓으면 따로 설정할 것은 없음, 그냥 사용



sendObject.jsp 오브젝트 보내기

el2.jsp 오브젝트 받아서 출력

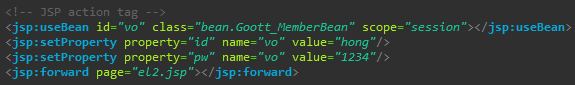
1. 스크립틀릿으로 구현한 것



1. JSP actiob tag 사용하여 또 구성

Property가 id?같은거고 value가 안에 들어있는 값 name은 usebean의 클래스와 매칭용

Forward는 주소값을 밝혀주지 않는다.

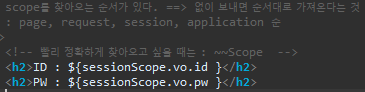


EL은 보안상 권장사항이 아님!!!!!!!

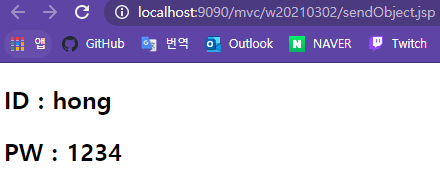
El2.jsp 에서는 이거만 쓰면 값 받아올 수 있다.



(찾아오는 순서, scope등 설명)



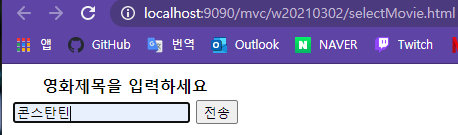
결과는 같다.

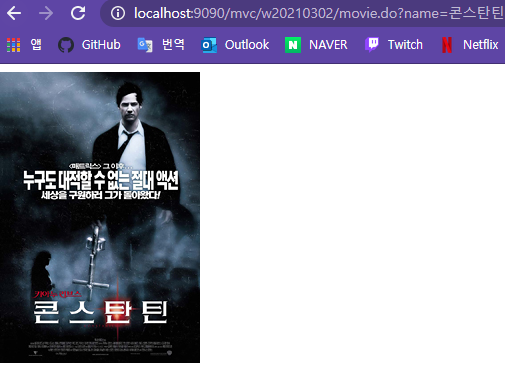


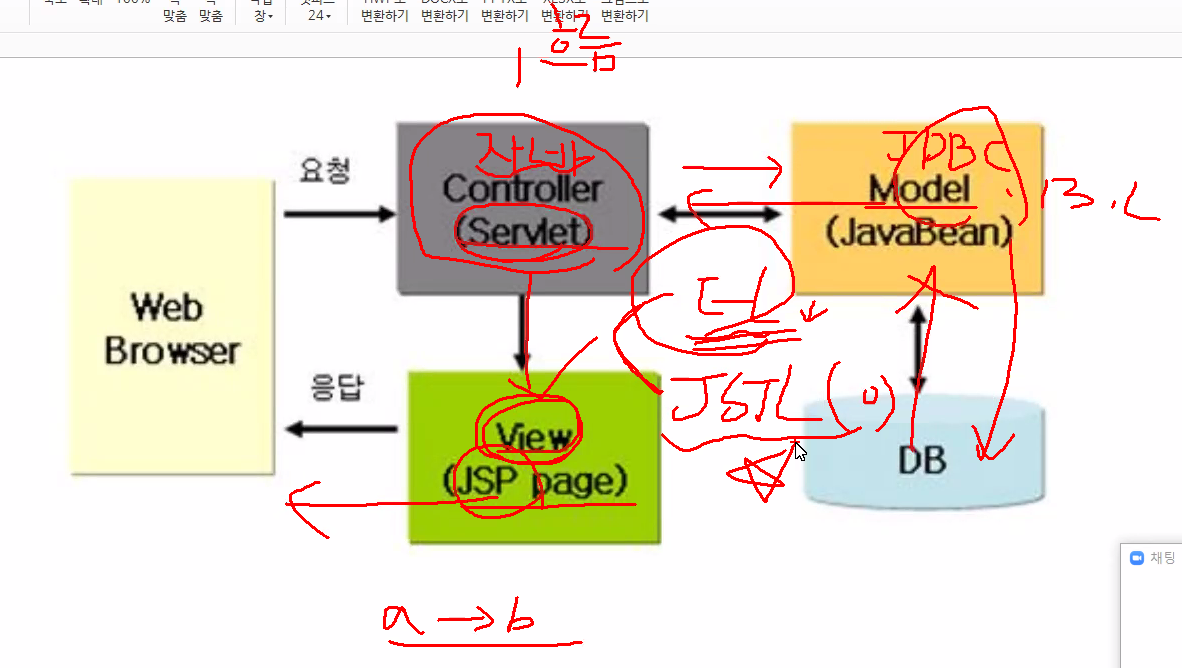
2번의 방식이 더 권장사항이다.

selectMovie.html

movie.do







수업은 EL로 진행하지만 프로젝트는 JSTL로 해야한다.

+ JSTL(JSP Standard Tag Library)

: jsp action tag는 커스텀이 가능 => 잘만들어진 tag를 모아놓은 tag libarary

cf) jQuery: javascript 잘 되어있는 것을 모아놓은 library

: 사용방법 – jQuery 호출하는 것처럼 jstl도 추가해서 사용하면 됨

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

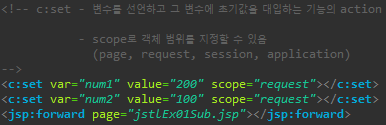
<!**DOCTYPE** html>

jstlEx01.jsp

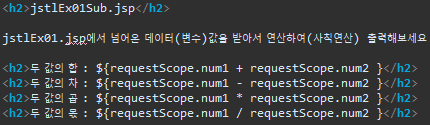
**c:set - 변수를 선언하고 그 변수에 초기값을 대입하는 기능의 action**

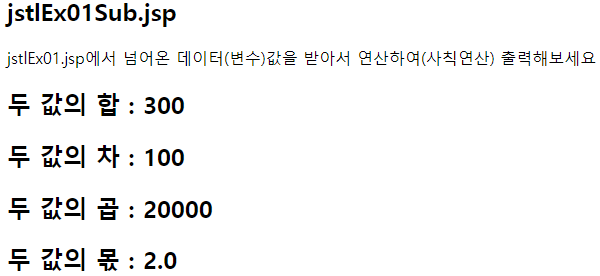
**- scope로 객체 범위를 지정할 수 있음**

**(page, request, session, application)**



jstlEx01Sub.jsp





EL 사용함

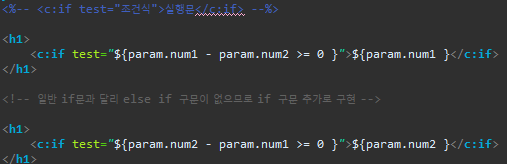
필요한 데이터는 el로 받아오고 최종 출력은 jstl로 하는게 권장 사항이다.

jstlEx02.jsp 사용자로부터 입력값 받아서 sub로

jstlEx02Sub.jsp 최대값 구하기

<**c:if** test=*"조건식"*>실행문</**c:if**>

일반 if문과 달리 else if 구문이 없으므로 if 구문 추가로 구현

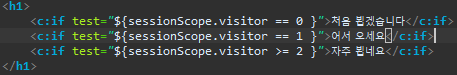


form으로 넘기니까 param으로 넘겨짐

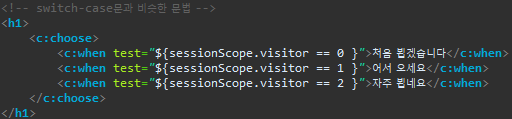
jstlEx03.jsp 방문횟수 보내기(원래는 DB에서 뽑아쓰는 것)



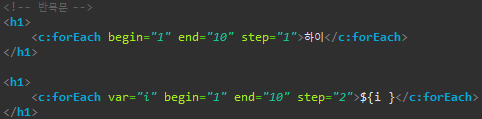
jstlEx03Sub01.jsp 방문횟수 받아서 출력

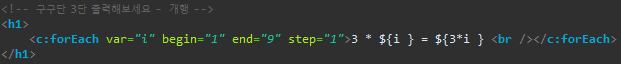


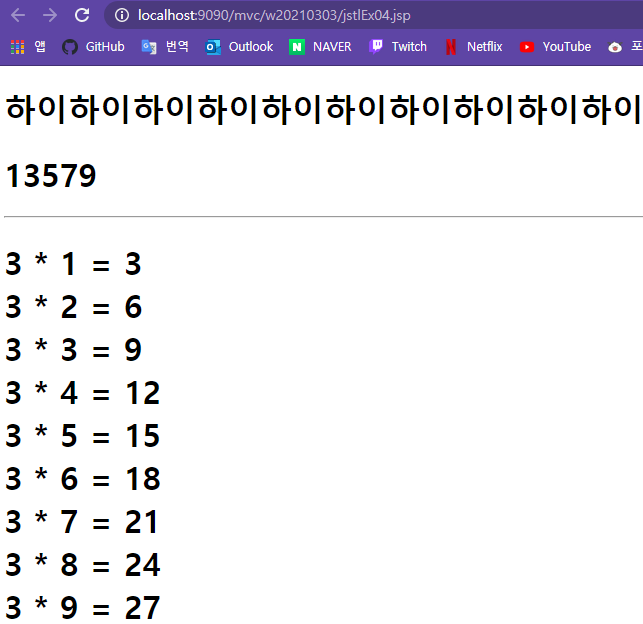
jstlEx03Sub02.jsp switch-case문과 비슷한 문법

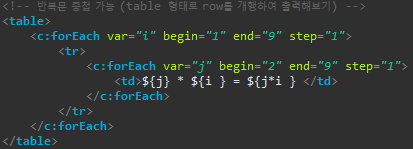


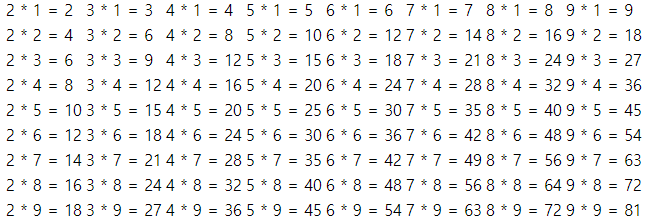
jstlEx04.jsp 반복문



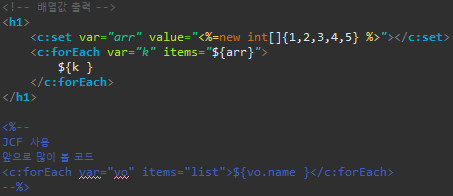


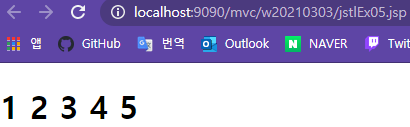




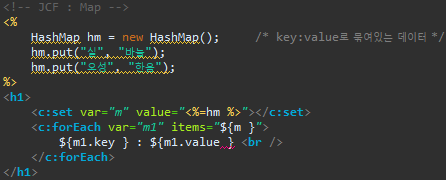


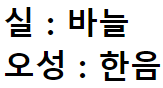
jstlEx05.jsp 배열값 출력



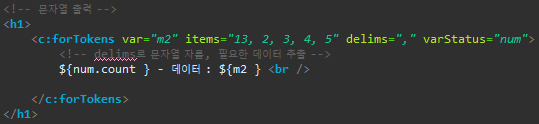


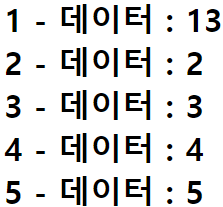
JCF: Map





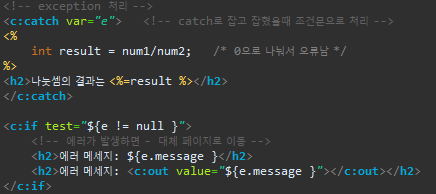
문자열 출력, 필요한 데이터 추출



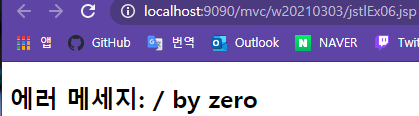


jstlEx06.jsp exception 처리, catch로 잡고 잡혔을때 조건문으로 처리

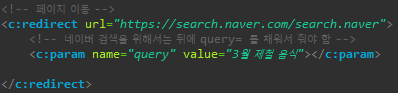
페이지 이동(네이버 검색)

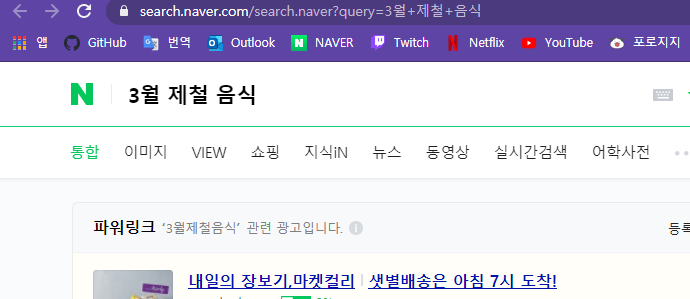


보안상 두 에러 메시지 출력 방법 중 두번째 방법을 사용해야 한다.



네이버 검색을 실행하도록 페이지 이동





sendData.html 데이터 전송

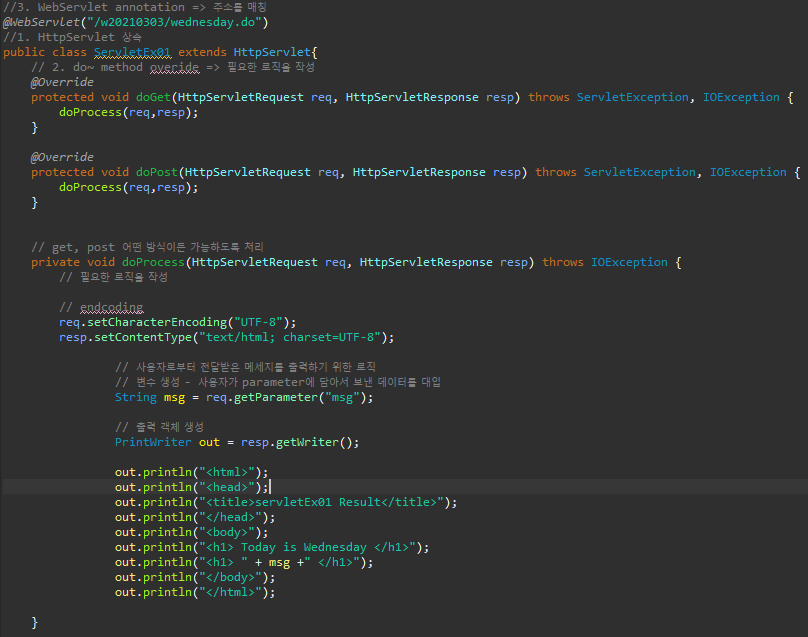
ServletEx01.java 서블릿 작성 방법 복습

1. HttpServlet 상속

2. do~ method overide => 필요한 로직을 작성

3. WebServlet annotation => 주소를 매칭

get이든 post든 받아서 처리할 수 있도록 함수를 선언하여 정리



ServletEx02.java 현재시간 띄우기

ServletEx03.java 또 다른 서블릿 사용 방법

// 서블릿을 작성하는 방법 - 2

// 1. GenericServlet 상속

// 2. service(~req, ~resp) override => 필요한 로직을 작성

// 3. @WebServlet 주소 지정]

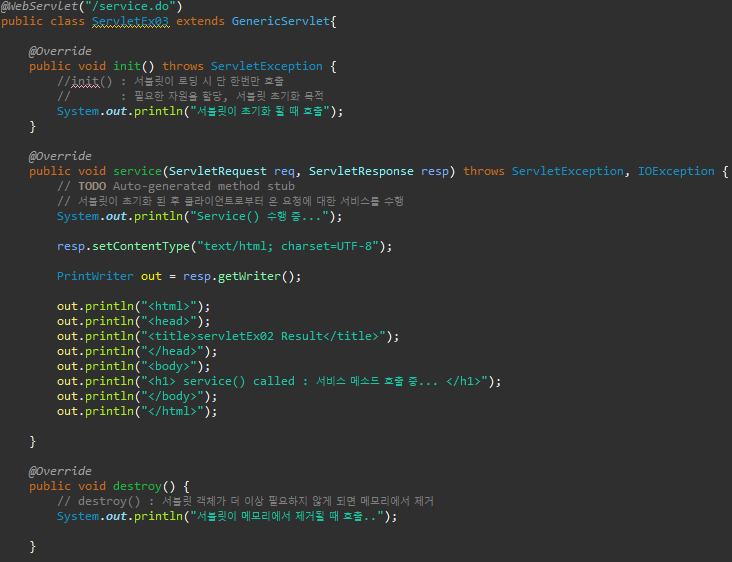
//서블릿은 자바 기술, 근데 톰캣쪽의 API에 javax.servlet.GenericServlet 이 기본

//init() : 서블릿이 로딩 시 단 한번만 호출

// : 필요한 자원을 할당, 서블릿 초기화 목적

// Service() : 서블릿이 초기화 된 후 클라이언트로부터 온 요청에 대한 서비스를 수행

// destroy() : 서블릿 객체가 더 이상 필요하지 않게 되면 메모리에서 제거



ServletEx04.java 객체 직렬화를 위한 코드

