**JSP, Servlet 내용정리**

□ WAS

∙ WAS(Web Application Server)

- WAS : Web Server + Application Server + Service 의미

- 웹 서버는 JSP, Servlet를 위한 웹 컨테이너와 HTML을 서비스하는 HTTP Server가 있음

- Application 서버는 비즈니스 로직을 수행하는 컨테이너가 있음

- 서비스는 JNDI, JMS와 같은 것을 제공하여 다른 기능의 서버와 연결하여 사용하게 하거나

서버 안에서 편리하게 사용할 수 있다

- WAS에는 Tomcat, WebLogic, Zeus 등이 있음

- 클라이언트 서버 시스템 구조에서 서버 쪽 어플리케이션의 생성과 실행, 소멸을 관리하는

프로그램을 애플리케이션 서버라 함. 서블릿과 서블릿 컨테이너와 같이 웹을 기반으로

동작되는 애플리케이션 서버를 WAS라 부름

∙ 웹 컨테이너(Web Container)

- 웹 서버에는 HTML을 클라이언트로 보내는 HTTP서버, JSP/Servlet을 실행하는 ‘엔진’, 엔진과

JSP/Servlet을 실행하는데 필요한 라이브러리 등을 포함한 웹 컨테이너가 있음.

웹 프로그래밍에는 웹 컨테이너가 반드시 있어야 함

∙ 서블릿(Servlet)

- 서블릿은 컨테이너에서 실행되는 프로그램. 요청에 대한 내용을 처리하고 결과를 HTML로

만들고, 자바 코드로 구현함

∙ JSP(Java Server Page)

- 자바 서버 페이지의 약자로 서버에서 실행되는 웹 프로그래밍용 클래스를 말함

- JSP로 된 파일은 Servlet으로 변환됨

- JSP도 Servlet이므로, Servlet에서 사용하는 객체나 메서드를 JSP에서도 사용할 수 있음

∙ 웹 프로젝트 폴더 구조

- src : 자바 소스 파일을 두는 폴더. 이 폴더에 서블릿 클래스나 필터, 리스너 등 필요한

파일을 모든 자바 클래스 파일을 저장함

- build/classes : 컴파일 된 자바 클래스 파일(.class)이 높이는 폴더. 패키지에 소속된 클래스인

경우, 이 폴더에 해당 패키지가 자동으로 만들어짐

- WebContent : 웹 컨텐츠를 두는 폴더. 웹 어플리케이션을 서버에 배치할 때 이 폴더의

내용들이 그대로 복사됨 (ex) HTML, CSS, Javacscript, JSP, 이미지 등)

- WebContent / WEB-INF : 웹 어플리케이션의 설정과 관련된 파일을 두는 폴더. 이 폴더에

있는 파일은 클라이언트에서 요청할 수 없음. 따라서 HTML 등의 클라이언트에서 요청할

파일을 이 폴더에 두어서는 안됨

- WebContent / WEB-INF / web.xml : 애플리케이션 배치 설명서(Deployment Descriptor) 파일.

영어표현으로는 줄여서 DD파일이라고도 부름. 서블릿이나 필터, 매개변수, 기본 웹페이지

등 컴포넌트들의 배치 정보를 이 파일에 작성. 서블릿 컨테이너는 클라이언트의 요청을

처리할 때 이 파일의 정보를 참고하여 서블릿 클래스를 찾거나 필터를 수행하는 등의

작업을 수행. 매우 중요한 xml 파일

- WebContent / WEB-INF / lib : 자바 아카이브 파일(.jar)을 두는 폴더

∙ 서블릿의 라이프 사이클(Life Cycle)

- 웹 프로그래밍은 컨테이너가 반드시 필요하며, 프로그래밍을 엔진에서 실행하여 원하는

결과를 생성

- 서블릿은 컨테이너의 명령에 따라 생성, 초기화, 실행, 소멸 등 정해진 행동을 하고, 이렇게

하는 것을 라이프 사이클이라고 함

- 서블릿은 요청을 처음으로 받으면 서블릿을 생성

- init -> service -> destroy를 호출하게 됨. 일반적으로 init 메서드를 호출하여 service를 호출

- 서비스를 계속 수행하다 더 이상 요청이 없다면 destroy 메서드를 호출하여 자원을

회수하고 서블릿 객체를 제거함. 컨테이너는 초기화와 소멸 메서드를 한 번만 호출

∙ GET 특징

- url에 데이터를 포함 -> 데이터 조회에 적합

- 바이너리 및 대용량 데이터 전송 불가

- 요청라인과 헤드 필드의 최대 크기

- http사양에서는 제한사항 없음

- 대용량 url로 인한 문제 발생 -> 웹 서버에 따라 최대 크기 제한

∙ GET요청을 발생시키는 방법

- 웹 브라우저 주소창에 url을 입력하는 경우

- 링크를 클릭하는 경우 (a태그)

- 입력 폼의 method 속성값이 get인 경우

∙ GET의 쓰임새

- 자료를 검색한다거나, 게시글의 상세 정보를 본다거나, 특정 상품의 정보를 조회하는 것과

같이 데이터를 조회하는 경우에 적합

- url에 검색어나 게시글 번호를 포함할 수 있기 때문에 메일이나 단문 메시지 등을 통해

다른 사람에게 전달하기가 용이

- 즐겨찾기에 저장해 두면 언제든지 해당 페이지로 바로 찾아갈 수 있어 편의성이 높음

- 단점 : 보안에 취약. 입력한 정보가 그대로 노출. 로그인이나 개인정보 관리와 같은 보안을

요구하는 경우에는 GET요청을 하지 말아야 함

- 바이너리 데이터 전송 불가 : 파일을 전송할 수 없다. 이미지, 동영상 등은 바이너리 파일

- 정보가 너무 크면 웹서버에서 처리할 수 없어서, 바이너리를 전송하기에 GET은 부적합

- 상대적으로 GET의 전송 속도는 POST보다 빠르다

∙ POST 특징

- url에 데이터가 포함되지 않음 -> 외부 노출 방지

- 메시지 본문에 데이터 포함 -> 실행 결과 공유 불가

- 바이너리 및 대용량 데이터 전송 가능

- 로그인이나 결제정보와 같이 보안이 중요한 사항에 POST를 사용

- 보내는 데이터의 크기 제한이 없어서 대용량 데이터들이 사용되는 곳에 사용

- 장점 : 입력 값을 url에 노출하지 않음

- 단점 : 요청 결과를 공유할 수 없음