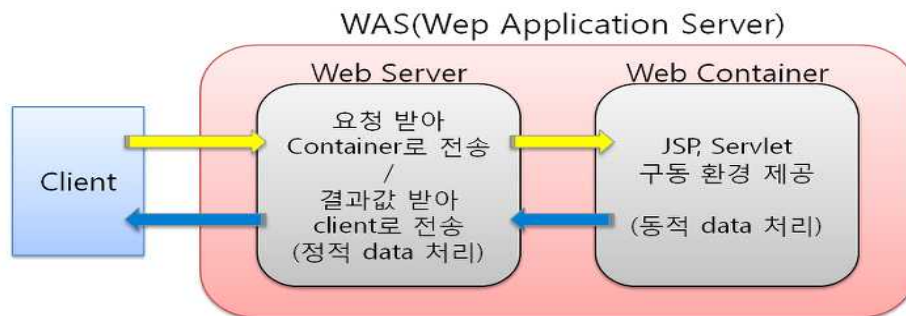


* Apache

- : 아파치 소프트웨어 재단의 오픈소스 프로젝트이다.
- : 웹서버로 불려진다
- : 클라이언트 요청이 왔을 때만 응답하는 정적 웹페이지에 사용된다. (HTML, CSS, 이미지 등)
- : 웹서버 - 80번 포트로 클라이언트 요청(POST, GET, DELETE)이 왔을 때만 응답

* Tomcat

- : Servlet이나 JSP의 컨테이너
- : WAS(Web Application Server)
- : 컨테이너, 웹 컨테이너, 서블릿 컨테이너라고 부른다
- : dynamic(동적)인 웹을 만들기 위한 웹 컨테이너



* 언어별 웹 서버 구성

- (JAVA) JSP, Servlet -> 아파치 톰캣 -> WAS 서버 (WEB서버 내장)
- (C, C++) PHP -> RWAPM -> WAS 서버 (WEB서버 내장)
- (MFC, .NET) ASP, ASPX -> IIS -> WAS 서버 (WEB서버 내장)

1. 다운로드

tomcat.apache.org

apache-tomcat-9.0.67.exe

2. 설치

apache-tomcat-9.0.67.exe 더블클릭 -> C:\WTomcat 9.0 위치 선택

Tomcat이 설치되는 default 위치

C:\WProgram Files\WApache Software Foundation\WTomcat 9.0

3. 서비스

- ① 시작 - 제어판 - 관리도구 - 서비스
- ② 내PC - 관리 - 서비스

시작유형 : 수동

서비스 : 중지

4. 서버실행

C:\Tomcat 9.0\bin\Tomcat9.exe 더블클릭

5. Tomcat의 메인화면

http://localhost:8080

http://localhost:8080/index.jsp

<http://127.0.0.1:8080>

* 환경변수

JAVA_HOME (JDK의 위치)

C:\Program Files\Java\jdk-11.0.10

TOMCAT_HOME or CATALINA_HOME (Tomcat의 위치)

C:\Tomcat 9.0

[실습] hello.html

default Context : ROOT

<http://localhost:8080/webapps/Context명/파일>

http://localhost:8080/ROOT/hello.html - X ← default Context명 ROOT는 쓰시면 안된다

<http://localhost:8080/hello.html> - O

ROOT가 아닌 다른 Context일 경우에는 반드시 URL에 써 주어야 한다.

Servlet

- : 웹에서 실행하는 프로그램
- : html in JAVA
- : public static void main(String[] args) 메소드가 없다
- : 주기함수(Life Cycle)
- : 반드시 public 이어야한다.
- : new X
- : 서버 안에 저장

주기함수(Life Cycle)

init() : 맨 처음에 1번만 호출
↓
service() - doGet() : 클라이언트가 요청시마다 호출
 - doPost()
↓
destroy()

서비스

1. get방식

- : default
- : 주소표시줄(Query String)를 통해서 이동
- : 이동되는 데이터가 보인다
- : 이동되는 데이터가 문자열만(String) 처리

2. post방식

- : 클라이언트가 post로 요청 시에만 적용
- : 내부적으로(페이지 단위) 이동
- : 이동되는 데이터가 안 보인다.
- : 대량 데이터

[실습] 회원가입과 로그인

Project : memberServlet

Folder : member

File : writeForm.html

loginForm.html

Package : member.service

Servlet : WriteServlet.java

Package : member.dao

Class : MemberDAO.java

Package : member.bean

Class : MemberDTO.java

[테이블 작성]

```
create table member(  
name varchar2(30) not null,  
id varchar2(30) primary key, --기본키, unique, not null, 무결성 제약 조건  
pwd varchar2(30) not null,  
gender varchar2(3),  
email1 varchar2(20),  
email2 varchar2(20),  
tel1 varchar2(10),  
tel2 varchar2(10),  
tel3 varchar2(10),  
zipcode varchar2(10),  
addr1 varchar2(100),  
addr2 varchar2(100),  
logtime date);
```

JSP (Java Server Page)

: 웹에서 실행하는 프로그램

: java in HTML

1. 선언문

<%! 전역변수 or 메소드 - 1번 처리 %> init()

2. 스크립트릿 (scriptlet)

<% 지역변수 or service처리 - 요청 시 마다 처리 %> service() - doGet(~), doPost(~)

3. 출력

<%= 값 or 변수 %>

[실습]

Context : testJSP

File : hello.jsp

http://localhost:8080/testJSP/hello.jsp

hello.jsp



hello_jsp.java (서블릿)



내용이 바뀔 때만 컴파일

hello_jsp.class

내장객체

1. request : javax.servlet.http.HttpServletRequest
2. response : javax.servlet.http.HttpServletResponse
3. out : javax.servlet.jsp.JspWriter
4. session : javax.servlet.http.HttpSession
5. application : javax.servlet.ServletContext
6. pageContext : javax.servlet.jsp.PageContext
7. page : javax.servlet.jsp.HttpJspPage
8. config : javax.servlet.ServletConfig
9. exception : java.lang.Throwable

주석

1. Java

// 1줄

/*

2줄 이상

*/

2. HTML

<!--

웹브라우저에는 안보이나 소스보기(F12)하면 보인다.

내부적으로는 처리(<% %> <%= %> 수행한다)

-->

3. JSP

<%--

웹브라우저에도 안보이고 소스보기(F12)해도 안 보인다.

--%>

[실습] 방명록

Project : guestbookJSP

Folder : guestbook

File : guestbookWriteForm.jsp
 guestbookWrite.jsp
 guestbookList.jsp

Package : guestbook.dao

Class : GuestbookDAO.java

Package : guestbook.bean

Class : GuestbookDTO.java

[테이블 작성]

```
create table guestbook(  
seq    number primary key, -- 시퀀스 객체로부터 값을 얻어온다  
name   varchar2(30),  
email  varchar2(30),  
homepage varchar2(35),  
subject varchar2(500) not null,  
content varchar2(4000) not null,  
logtime date);
```

[시퀀스]

```
create sequence seq_guestbook nocycle nocache;
```