

The background features abstract, flowing green and white lines that create a sense of movement and depth. A solid dark green horizontal bar is positioned at the bottom of the slide.

# **Java Web Application**

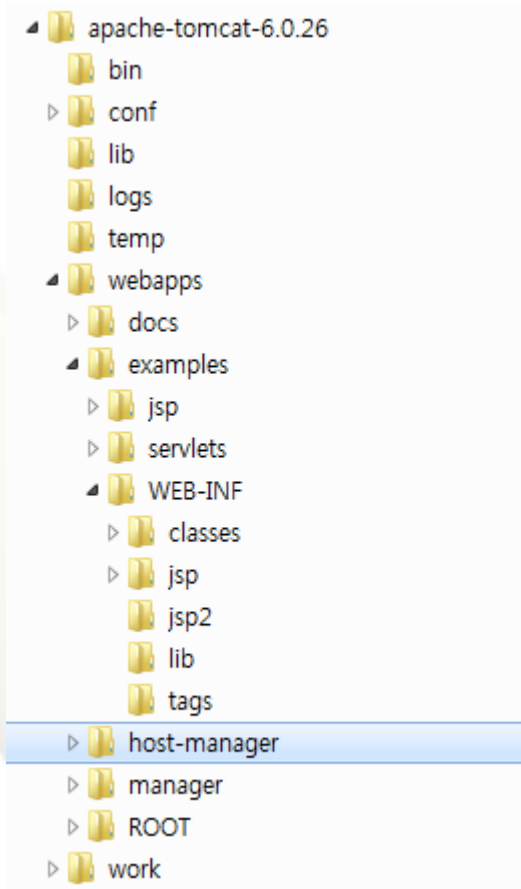
# 자바 웹 애플리케이션 구조

## ■ 웹 애플리케이션

- 톰캣 설치 경로의 webapps 디렉터리 각 하위 디렉터리들

## ■ 웹 애플리케이션의 구성

- 웹 애플리케이션 디렉터리
  - JSP(.jsp), HTML(.html) 등의 파일들이 위치
  - 웹을 통해 공개되는 영역
- WEB-INF 디렉터리
  - .class 파일, .jar 파일, web.xml 파일 등이 위치
  - .class 파일은 **classes** 디렉터리에 위치
  - .jar 파일은 **lib** 디렉터리에 위치
  - 웹을 통해 공개되지 않는 영역 (직접 접근 불가능)
- web.xml
  - 웹 애플리케이션의 환경설정 파일



# ROOT 웹 애플리케이션과 기본 페이지

## ■ ROOT 웹 애플리케이션

- 톰캣의 기본 애플리케이션으로 애플리케이션 이름을 지정하지 않고 접근
  - <http://localhost:8080/> 형식으로 접근
- 톰캣 설치 경로\webapps\ROOT 경로

## ■ 커스텀 웹 애플리케이션

- 애플리케이션 이름을 지정해서 접근
  - <http://localhost:8080/appname/> 형식으로 접근
- 톰캣 설치 경로\webapps\appname 경로

## ■ 기본 페이지

- 페이지 이름을 명시하지 않았을 경우 실행되는 페이지
- web.xml 파일에 등록

# 서블릿 작성

- HttpServlet 상속 클래스 정의
- doGet(HttpServletRequest, HttpServletResponse) 및 doPost(HttpServletRequest, HttpServletResponse) 재정의
- 컴파일 및 .class 파일을 WEB-INF\classes 디렉터리에 저장 (이클립스에서는 자동)

```
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class GreetingsServlet extends HttpServlet {
    public void doGet(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException {
        //Process Request
    }
    public void doPost(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException {
        //Process Request
    }
}
```

# 서블릿 등록

- web.xml 파일에 등록
- 특정 요청과 등록된 서블릿 클래스 바인딩

```
<?xml version="1.0" encoding="UTF-8"?>

<web-app>

    <servlet>
        <servlet-name>GreetingsServlet</servlet-name>
        <servlet-class>GreetingsServlet</servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>GreetingsServlet</servlet-name>
        <url-pattern>/GetGreetings</url-pattern>
    </servlet-mapping>

</web-app>
```

- 등록 후 <http://localhost:8080/appname/GetGreetings> 로 요청

# JSP 작성 및 실행

- 웹 애플리케이션 하위 경로에 .jsp 로 저장
- 요청시 파싱되어 클래스 파일로 변경된 후 컴파일되어 실행
- 컴파일 결과는 서블릿 클래스

```
<%@ page language="java"
      contentType="text/html; charset=EUC-KR"
      pageEncoding="EUC-KR"%>

<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
    <title>Insert title here</title>
</head>
<body>
    <%= "Hello, JSP Programming World !!!!!" %>
</body>
</html>
```

# HTTP 요청 (Link와 Submit)

- Browser를 통해 두 가지 형태의 요청을 서버로 전달
  - 주소직접입력 또는 하이퍼링크를 클릭해서 특정 페이지 요청 (Link)
  - 응답된 페이지에 사용자 데이터를 입력하고 데이터와 함께 서버로 전송 (Submit)
  - Submit 할 데이터는 form 태그 단위로 입력되고 전송됩니다.
  - form 태그에 action 속성으로 전송 대상을 지정
  - form 태그에 method 속성으로 데이터 전송방법을 지정
    - post : 메시지 본문에 데이터를 기록하고 전송
    - get : 주소 뒤에 ?이름=값&이름=값2의 형식으로 데이터 전송

