

# 개발환경 구성

---

558280-1  
2026년 봄학기  
3/5/2026  
박경신

# 인터넷과 웹 서비스

## □ 인터넷과 WWW(World Wide Web)

- 인터넷은 TCP/IP 기반의 네트워크가 전세계적으로 확대되어 하나로 연결된 '네트워크의 네트워크'
- WWW(World Wide Web)는 인터넷 기반의 서비스 중 하나

이름	프로토콜	포트	기능
WWW	http	80	웹 서비스
Email	SMTP/POP3/IMAP	25/110/114	이메일 서비스
FTP	ftp	21	파일 전송 서비스
telnet	telnet	23	원격 호스트 접속 서비스
DNS	DNS	83	도메인 이름 변환 서비스
News	NNTP	119	인터넷 뉴스 서비스

# 인터넷과 웹 서비스

## □ HTTP (Hyper Text Transfer Protocol)

- Protocol이란 네트워크에 연결된 컴퓨터가 서로 통신(대화)하기 위한 규약
- HTTP는 www 서비스를 위한 프로토콜
- 웹 서버와 클라이언트는 HTTP를 이용해 통신

## □ URL (Uniform Resource Locator)

- URL은 웹 어플리케이션에 요청을 구분하기 위한 용도로 사용됨
- 예: `http://www.google.com/search?hl=en&q=jsp&aq=f&oq=`
  - 프로토콜: http
  - 호스트: www.google.com
  - 포트: 80 (http 프로토콜의 기본 포트)
  - 경로: /search
  - 쿼리문자열: hl=en&q=jsp&aq=f&oq=

# 웹 프로그래밍 기술

## □ HTML과 웹 프로그래밍 기술

- HTML(HyperText Markup Language): www 서비스를 표현하기 위해 사용하는 언어
- HTML은 텍스트 파일로 정적인 정보만 처리 가능
- 동적인 콘텐츠 처리하기 위해 CGI(Common Gateway Interface), Fast CGI, PHP(Personal Home Page), ASP(Active Server Page), JSP(Java Server Page) 등의 기술이 사용

## □ 클라이언트 스크립트

- Javascript가 대표적. 웹 브라우저가 스크립트 해석의 주체
- 웹 브라우저 핸들링은 가능하지만 서버 연동은 불가능

## □ 서버 스크립트

- ASP.NET, PHP, Servlet, JSP 등. HTML과 스크립트 언어를 함께 사용할 수 있는 기술로 웹 서버에서 해석.
- 데이터베이스 연동 처리 등 다양한 구현이 가능
- 별도의 컴파일 과정없이 HTML 태그 수정 가능

# 웹 컨테이너

## □ 웹 컨테이너

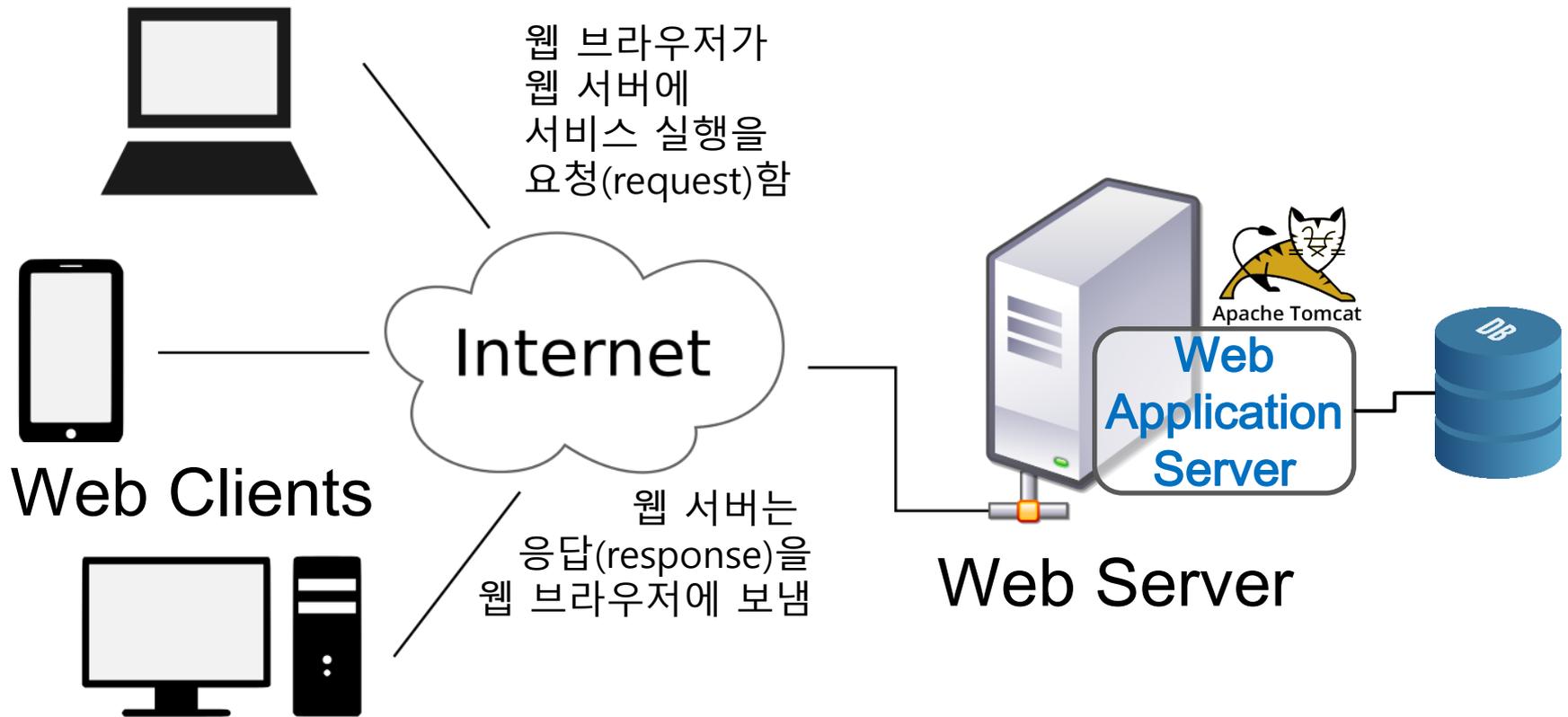
- Servlet 실행 환경을 제공하며, WAS (Web Application Server)라고 함
- 웹 브라우저의 요청에 대한 응답을 웹 서버에 전달하는 기능을 제공하며 웹 응용 프로그램의 보안, 관리, 라이프 사이클관리 등의 서비스를 제공

## □ 주요 웹 컨테이너

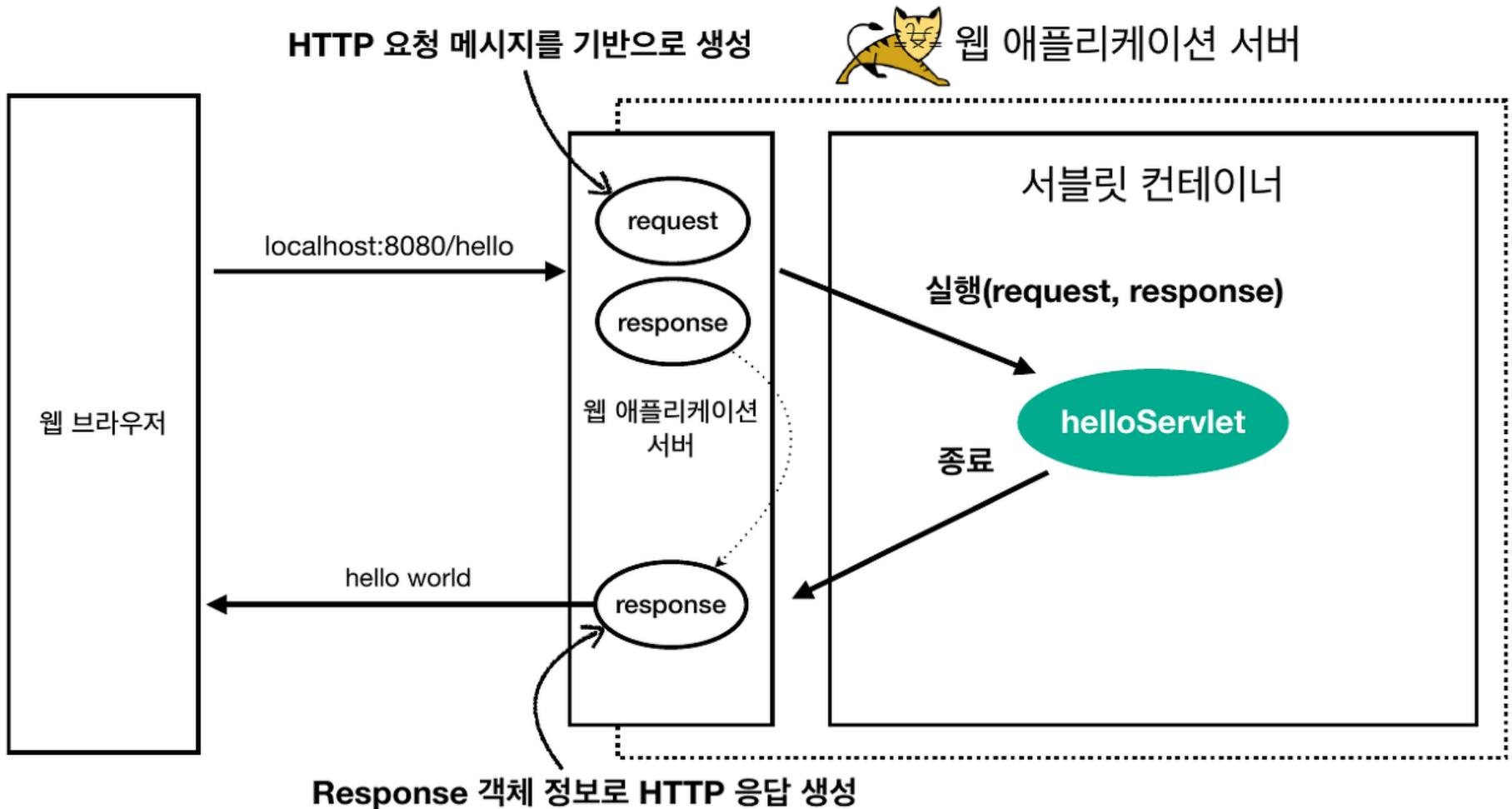
- 톰캣(Tomcat) : <http://tomcat.apache.org/>
- 제티(Jetty) : <http://www.eclipse.dev/jetty/>
- 엔진엑스(Nginx) : <http://www.nginx.com/>
- JBoss : <https://www.redhat.com/ko/technologies/jboss-middleware/web-server>
- WebLogic : <https://www.oracle.com/kr/java/weblogic/>
- WebSphere : <https://www.ibm.com/kr-ko/products/websphere-application-server>

# Web Architecture

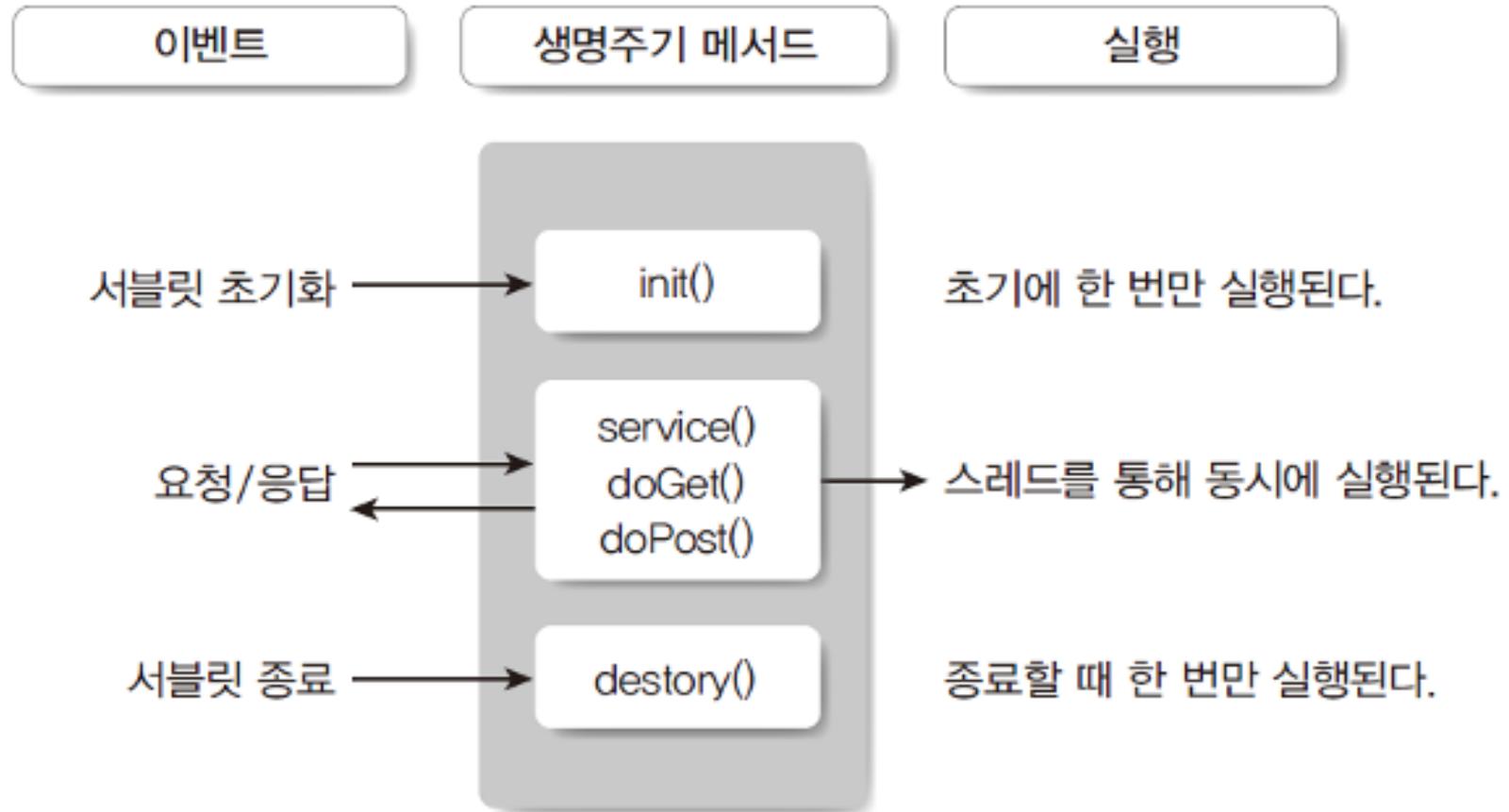
## □ 웹 어플리케이션 구조



# Web Architecture



# Servlet Lifecycle



# 개발 환경 설정

---

- JDK 설치 (**OpenJDK 25.0.2** <https://jdk.java.net/25/>)
- IDE 설치 (**Visual Studio Code**  
<https://code.visualstudio.com/download#>)
- 환경설정
  - VSCode Java Extension Pack:  
<https://marketplace.visualstudio.com/items?vscjava.vscode-java-pack>
  - Spring Boot Extension Pack - VMware:  
<https://marketplace.visualstudio.com/items?itemName=vmware.vscode-boot-dev-pack>
  - Lombok Annotations Support for VS Code - Microsoft:  
<https://marketplace.visualstudio.com/items?vscjava.vscode-lombok>

# 프로젝트 생성

---

- 스프링부트 Hello 프로그램을 작성한다.
  - Ctrl+Shift+P를 눌러 커맨드 팔레트(Command palette)를 열어 'Spring initializr: Create a Maven Project' 선택. Dependencies: Lombok, Thymeleaf, Spring Web, Spring Boot Dev Tools

# 프로젝트 생성

The image shows a screenshot of the Visual Studio Code (VS Code) editor interface. The left sidebar displays the Explorer view with a project structure for 'spring-boot-hello'. The main editor area shows the code for 'SpringBootHelloApplication.java'.

**Project Structure (Left Sidebar):**

- 탐색기 (Search)
- 열려 있는 편집기 (Open Editors)
  - index.html src\main\resources\static
  - SpringBootHelloApplication.java src\main\ja...
- SPRING-BOOT-HELLO
  - .mvn
  - .vscode
  - src
    - main
      - java\kr\ac\dankook\ace\springboothello
        - SpringBootHelloApplication.java
      - resources
        - static
          - index.html
        - templates
      - application.properties
      - test
    - target
    - .gitignore
    - HELP.md
  - 개요 (Overview)
  - 타임라인 (Timeline)
  - JAVA PROJECTS
  - MAVEN

**Code Editor (Main Area):**

File: SpringBootHelloApplication.java

```
src > main > java > kr > ac > dankook > ace > springboothello > SpringBootHelloApplication.java > ...
1 package kr.ac.dankook.ace.springboothello;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5
6 @SpringBootApplication
7 public class SpringBootHelloApplication {
8
9     Run | Debug
10    public static void main(String[] args) {
11        SpringApplication.run(primarySource:SpringBootHelloAppli
12    }
13 }
14
```

**Bottom Status Bar:**

줄 2, 열 1 Tab 크기: 4 UTF-8 LF {} Java

# 프로젝트 생성

The image shows a Visual Studio Code editor window with a dark theme. The title bar at the top reads "spring-boot-hello". The interface is divided into three main sections: a sidebar on the left, a central editor area, and a status bar at the bottom.

**Sidebar (Left):** Contains a search bar, a list of open files, and a file explorer. The file explorer shows a project structure for "SPRING-BOOT-HELLO" with folders like ".mvn", ".vscode", "src", and "target". The "src" folder is expanded to show "main", which contains a "java" folder. The "java" folder is further expanded to show "kr", "ac", "dankook", "ace", and "springboothello". The "springboothello" folder contains "SpringBootApplication.java", "resources", and "static". The "static" folder is selected, showing "index.html".

**Editor Area (Center):** Displays two files: "index.html" and "SpringBootApplication.java". The "index.html" file is active and shows the following code:

```
src > main > resources > static > <> index.html > ...
1  <!DOCTYPE html>
2  <html xmlns:th="http://www.thymeleaf.org">
3  <head>
4  <meta charset="UTF-8">
5  <title>Insert title here</title>
6  </head>
7  <body>
8  |   Hello World! This is a first springboot web project!
9  </body>
10 </html>
11 |
```

**Status Bar (Bottom):** Shows the current file encoding as "UTF-8", line endings as "CRLF", and the file type as "HTML". It also displays "줄 11, 열 1" (Line 11, Column 1), "공백: 4" (Spaces: 4), and "Java: Ready".

# 프로그램 실행

- src/main/java/kr/ac/dankook/ace/springboothello/Spring BootHelloApplication.java 실행하면 톰캣이 실행됨
- 웹브라우저에서 localhost:8080/ 실행하면 src/main/resources/static/index.html 내용이 나타남

