*프로젝트 소스 폴더와 컴파일 디렉토리

src/main

java/	컴파일 →		*.class	→	bin/main/*
resources/	*.properties	→			↑
	*.xml				classpath

편의성을 위해 java / resources가 나뉘지만 컴파일하면 같은 classpath로 들어감.

*Monolitic vs microservies

Monolithic	한 개의 Web Application	MicroServers	서비스당 한 개 서버
Server		Server1	
카페서비스		카페서비스	
메뉴서비스		Server2	
검색서비스		메뉴서비스	
블로그서비스		Server3	
		검색서비스	
		Server4	
		블로그서비스	

*서비스별로 관리한다. → 운영, 유지보수가 쉽다.
단, 한번의 로그인으로 여러 서버 접속이 가능해야 한다.

*SpringBoot와 라이브러리, 그리고 설정

예전 방식	■ 라이브러리 ■ Spring Freamwork	+개발자가 설정	■ 클래스 작성 ■ 설정 값 할당		yml
스프링부트 방식	[Starter] ■ 라이브러리 정보 ■ 설정	+application.properties		■ 기본 설정 변경 ■ 함수 항목의 값을 설정	

스프링부트에서 뭔터를 쓰고 싶다면 뭔터 인터페이스를 만들고 @컴포넌트를 붙이면
디스패쳐 서블릿(프론트컨트롤러에) 스프링부트가 자동 등록함.

*SpringBoot의 context path와 servlet path
http://localhost:8888/app/board/list

8888 = server.port
app = server.servlet.context-path
/board/list = servlet path

```
System.out.println(httpRequest.getContextPath());
String servletPath = httpRequest.getServletPath();
System.out.println(servletPath);
System.out.println(httpRequest.getPathInfo());
```

*089 Thymeleaf 사용

	/app/board/list context path	page controller path			
1.요청	<<Front Controller>> DispathcerServlet		2.list() →	<<page controller>> BoardController	작업수행
→			← 4."board/list"		
←	↑ HTML	↓5.실행			
응답	ThymeleafView				
	6.파싱				
	board/list.html	←	HTML + Thymeleaf 명령 > HTML 컨텐트로 변경		

https://www.thymeleaf.org/doc/tutorials/3.0/usingthymeleaf.html