



SPRING BOOT

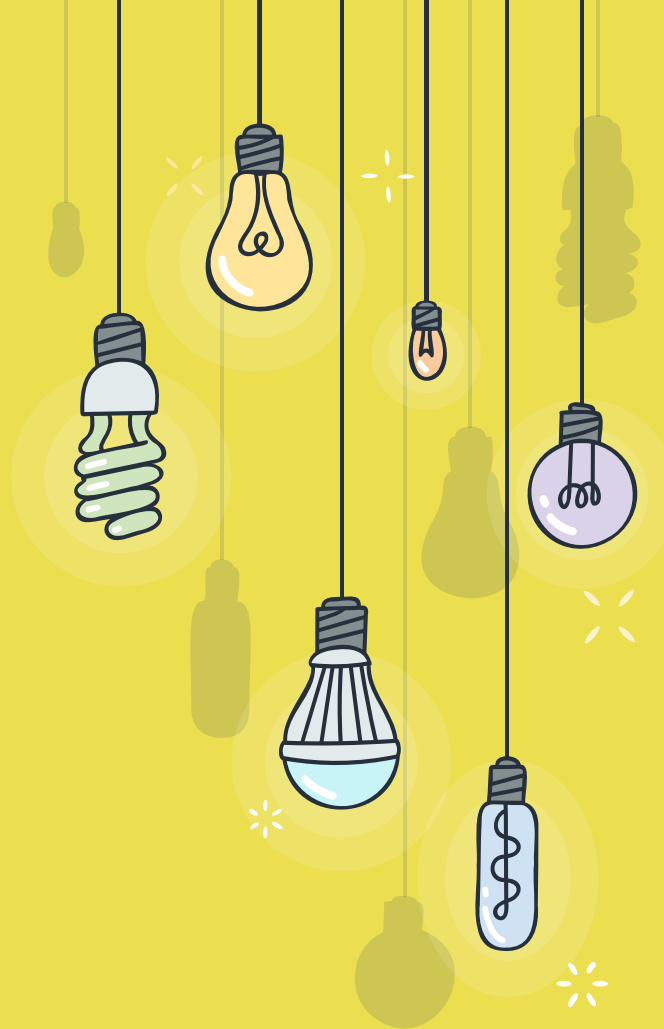
1. 의존성 주입(DI)

2. 자동 설정



1

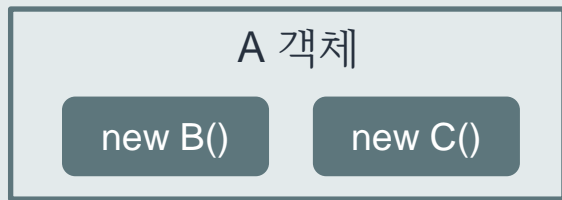
익존성주임



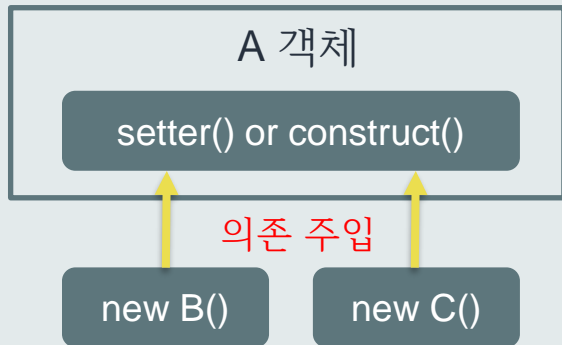
* 의존성 주입(DI)

+ 객체를 사용하는 2가지 방법

1. A객체가 B, C의 객체를 직접 생성



2. B, C객체가 외부에 생성되어 A객체에 주입



* IoC 컨테이너

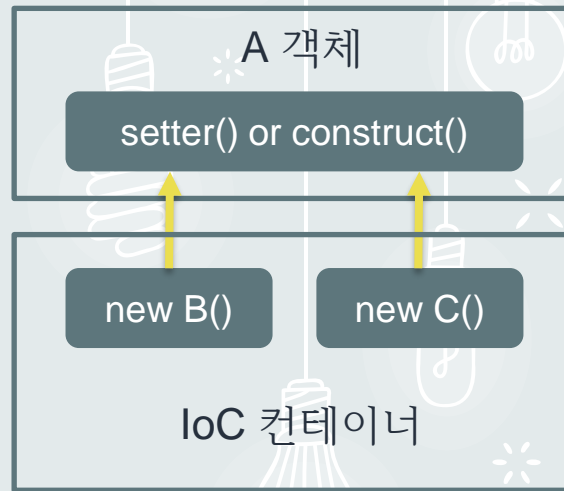
+ 컨테이너

- × 객체에 의존 주입을 하는 라이브러리 집합

+ Inversion of Control (제어의 역전)

- × 프레임워크가 요구하는 대로 객체 만들면
 - ◆ 프레임워크가 해당 객체들을 생성하고, 메서드를 호출하고, 소멸시킨다
- × 장점
 - ◆ 구현체 사이의 변경이 용이하다
 - ◆ 객체 간 의존성이 낮아진다

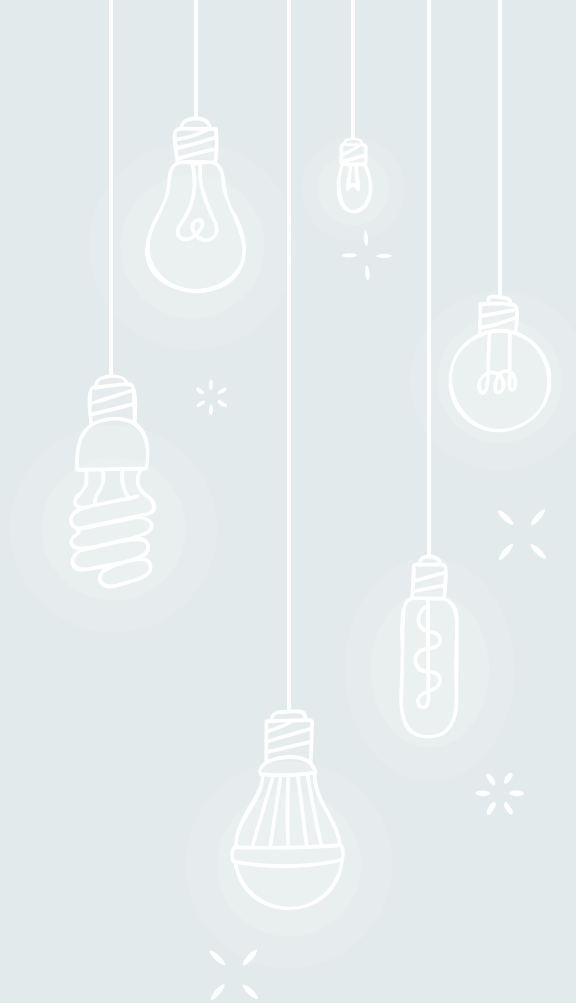
+ 스프링은 IoC 컨테이너로 빈을 관리



* 스프링 의존주입

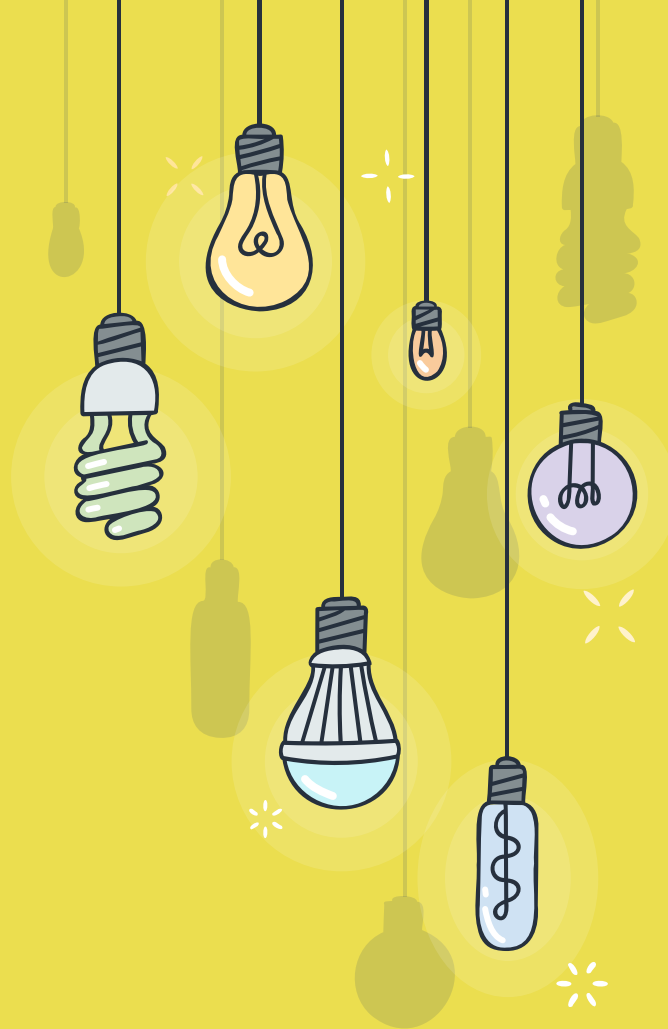
+ 스프링에서의 의존 주입 방법

- × bean configuration xml을 이용한 의존 주입
- × 자바 코드를 이용한 의존 주입
- × 어노테이션을 이용한 의존 주입



2

자동 설정



* 자동 설정

+ 자동 설정

- × 스프링 부트에서 복잡한 XML 설정 없이 웹 애플리케이션을 작성할 수 있는 이유는 내부적으로 자동 설정이 동작하기 때문

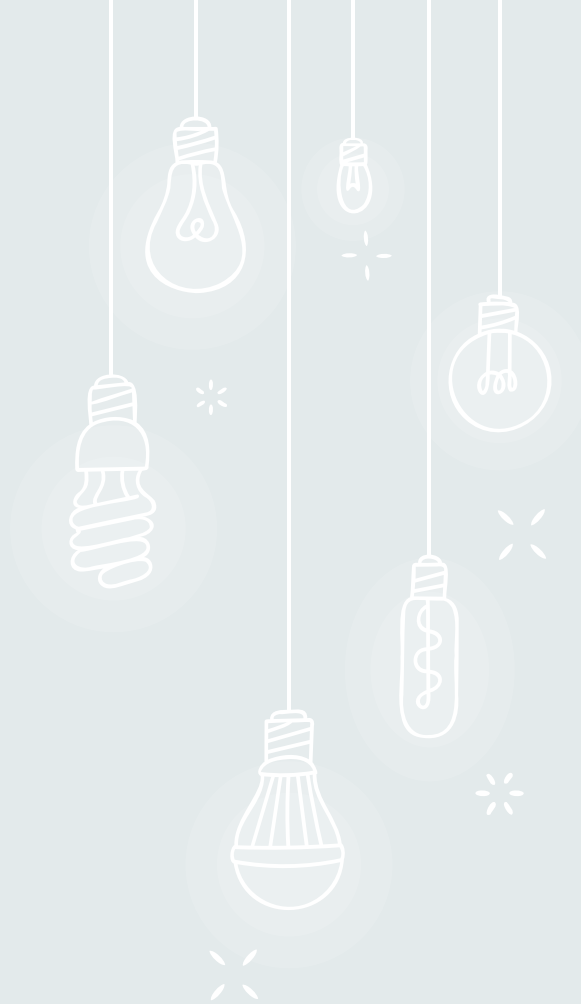
+ 자동 설정이란

- × 스프링 기반의 애플리케이션은 2종류의 객체 사용
 - ◆ 스프링이 제공하는 객체
 - ◆ 사용자가 작성한 객체

* 자동 설정 개요

+ 어노테이션 사용

- × @SpringBootApplication
- × 다음 어노테이션들을 포함
 - ◆ @Configuration
 - ◆ @EnableAutoConfiguration
 - ◆ @ComponentScan



* 어노테이션

- × @Configuration
 - ◆ 스프링 컨테이너는 설정 클래스를 로딩하고 설정 클래스에 @Bean으로 등록된 객체를 생성하고 관리
- × @ComponentScan
 - ◆ @ Configuration, @ Repository, @Service, @ Controller, @RestController 가 붙은 클래스들을 자동으로 scan하여 Bean으로 등록 해주는 역할
 - ◆ @ComponentScan이 설정된 클래스의 하위 패키지에 위치해야 함

* 어노테이션

- × @EnableAutoConfiguration
 - ◆ 스프링부트 자동 구성을 활성화 한다.
 - ◆ 스프링부트는 의존성을 기반으로 자동으로 프로젝트를 구성한다.
 - ◆ 즉 자동환경설정이라고 볼수있다.
 - ◆ spring.factories 안에 들어있는 수많은 자동 설정들이 조건에 따라 적용이 되어 수 많은 Bean들이 생성되고, 스프링 부트 어플리케이션이 실행되는 것

THANKS!

+ Any questions?

