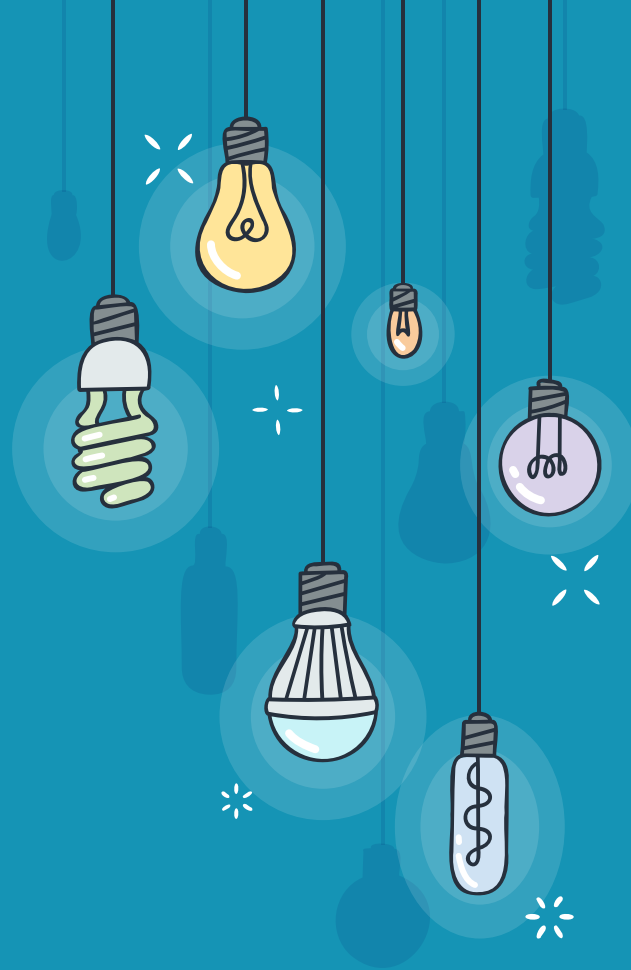


SPRING BOOT

1. FRAMEWORK 개요

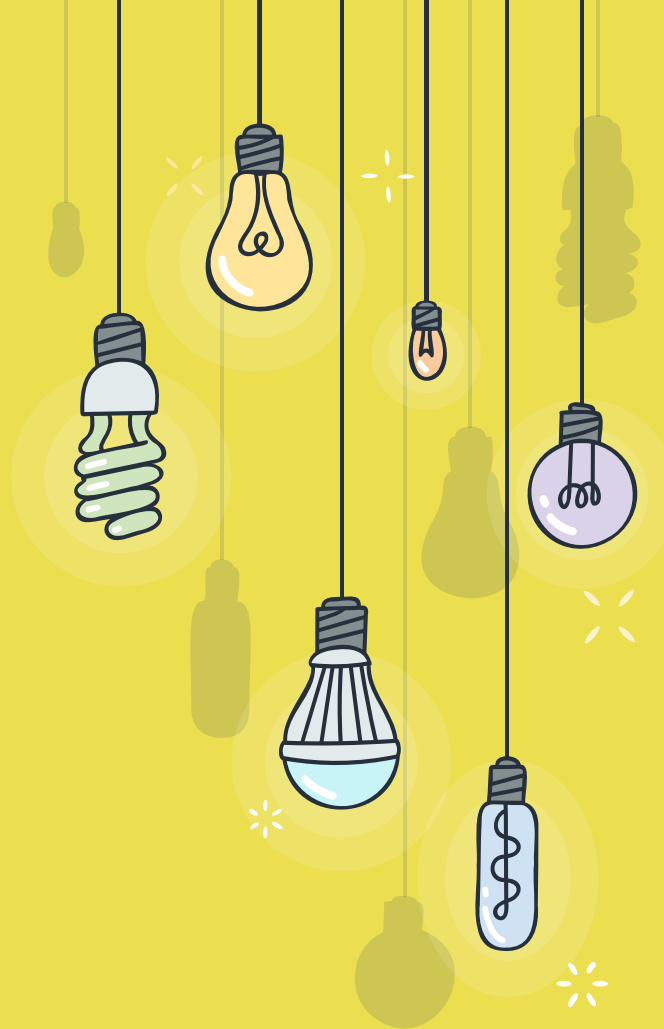
2. SPRING FRAMEWORK

3. SPRING BOOT



1

# FRAMEWORK 개요



# \* FRAMEWORK 개념[1/2]

## + 개발자가 소프트웨어 개발 시

- × 일련의 클래스 묶음이나 뼈대, 틀을 제공하는 라이브러리를 구현해 놓은 것

## + 특징

- × 개발자가 따라야 하는 가이드 제공
- × 개발할 수 있는 범위가 정해져 있음
- × 개발자를 위한 다양한 도구 및 플러그인 지원
- × 프로그램 개발 시간 단축
- × 프로그램의 재사용성 증가

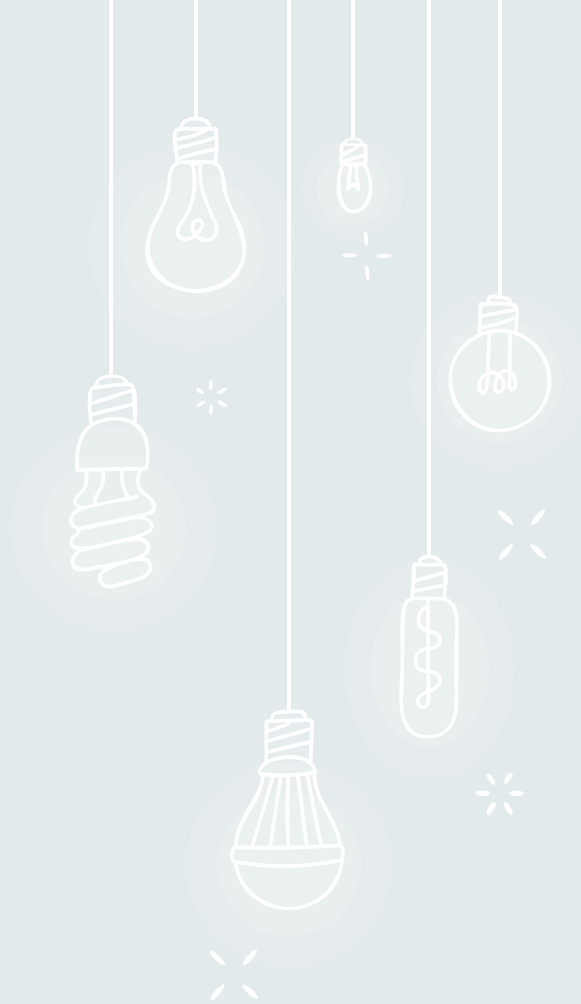
# \* FRAMEWORK 개념[2/2]

## + 장점

- × 프로그램 개발 시간 단축
- × 관리의 용이성 증가
- × 개발자들의 역량 획일화
- × 검증된 아키텍처의 재사용
- × 아키텍처의 일관성 유지

## + 단점

- × 지나친 의존 시 개발자들의 능력이 떨어져 스스로 개발하는 것이 어려워짐
- × 습득 시간이 오래 걸림

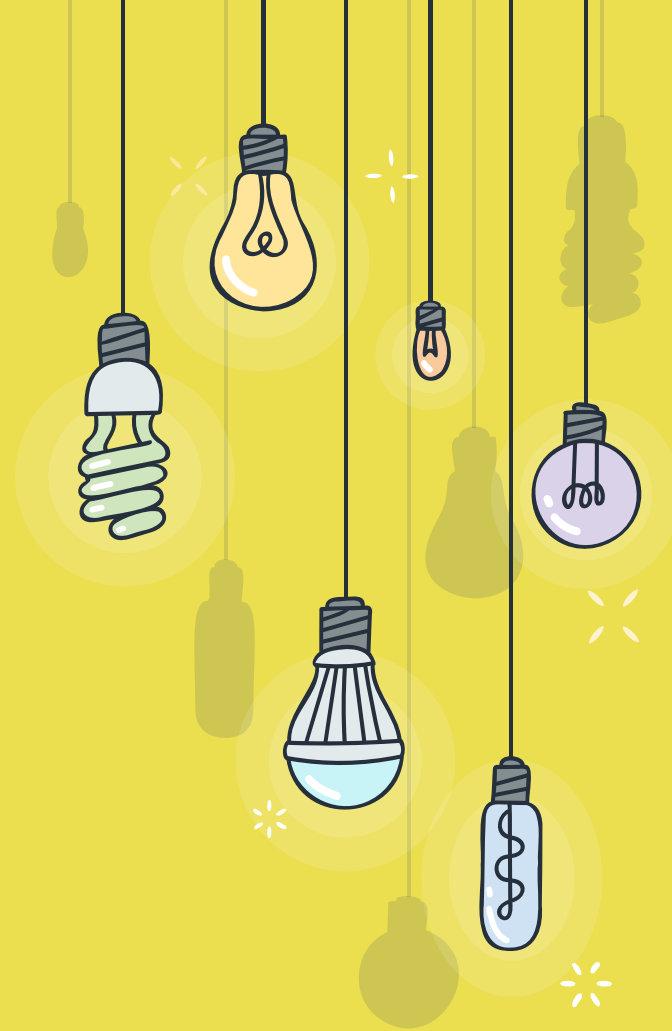


# \* FRAMEWORK 종류

구분	종류	설명
영속성 Framework	MyBatis, Hibernate	데이터의 저장, 조회, 변경, 삭제를 다루는 클래스 및 설정 파일을 라이브러리화하여 구현
자바 Framework	Spring Framework, 전자정부표준 Spring, Struts	Java EE를 통한 웹 애플리케이션 개발에 초점을 맞춰 필요한 요소들을 모듈화하여 제공
화면구현 Framework	Bootstrap, Foundation, MDL	Front-End를 보다 쉽게 구현할 수 있게 틀 제공
기능 지원 Framework	Log4j, JUnit 5, ANT	특정 기능이나 업무 수행에 도움을 줄 수 있는 기능 제공

2

## SPRING FRAMEWORK



# \* SPRING FRAMEWORK 개요

## + Spring Framework 개요

- × 애플리케이션 운용에 필요한 객체를 생성하고 객체들 사이에 의존성(Dependency)을 관리하는 컨테이너 제공
- × 자바 플랫폼을 위한 오픈소스 애플리케이션 프레임워크
- × 전자정부 표준 프레임워크의 기반 기술

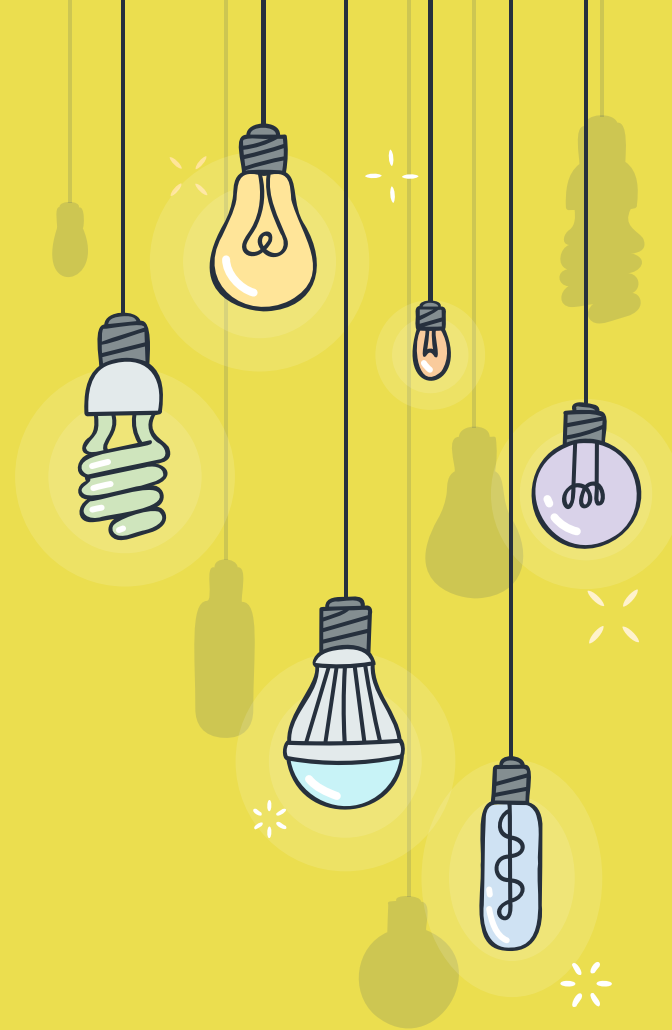
## + Spring의 변화

- × 초기에는 단순히 객체를 관리하는 컨테이너 기능만 제공
- × 현재는 개발에 필요한 모든 분야를 지원하는 하나의 플랫폼으로 사용



3

SPRING BOOT



# \* SPRING BOOT 개요

## + Spring Boot의 등장

### × 스프링의 복잡도 증가

- ◆ 본래 웹 애플리케이션만을 목적으로 만든 것이 아님
- ◆ 그러나 대부분 웹 애플리케이션에서 사용하고, 점점 웹 애플리케이션이 복잡해 지면서 라이브러리 관리 및 xml환경설정 등 설정에 많은 노력이 필요해짐

### × Spring Boot 등장

- ◆ Spring과 달리 웹 애플리케이션을 목적으로 함
- ◆ Spring 처럼 많은 설정이 필요 없음(개발자들이 쉽게 접근)

# \* SPRING BOOT의 특징

- + 라이브러리 관리 자동화
  - × Starter를 통한 라이브러리 의존성문제 해결
- + 설정의 자동화
  - × 추가된 라이브러리와 환경 설정을 자동으로 처리
- + 라이브러리 버전 자동 관리
  - × 스프링 라이브러리 뿐만 아니라 관련된 서드파티 라이브러리도 함께 다운로드
- + 테스트 환경과 내장 서버
  - × Junit이나 톰캣 간은 웹 서버 내장
- + 독립적으로 실행 가능한 JAR
  - × 빠른 개발과 빠른 배포 가능

# THANKS!

+ Any questions?

