**02. 개발에 앞서 알면 좋은 기초 지식**

**1. 서버 간 통신**

**①** 스프링의 핵심 가치

: 애플리케이션 개발에 필요한 기반을 제공해서 개발자가 비즈니스 로직 구현에 집중하게 함.

**②** 제어 역전(IoC: Inversion of Control)

: 사용할 객체를 직접 생성하지 않고 생명 주기를 외부에 위임. 여기서 외부란 스프링 컨테이너 또는 IoC 컨테이너를 의미 -> 의존성 주입(DI), 관점 지향 프로그래밍(AOP)

**③** 의존성 주입(DI: Dependency Injection)

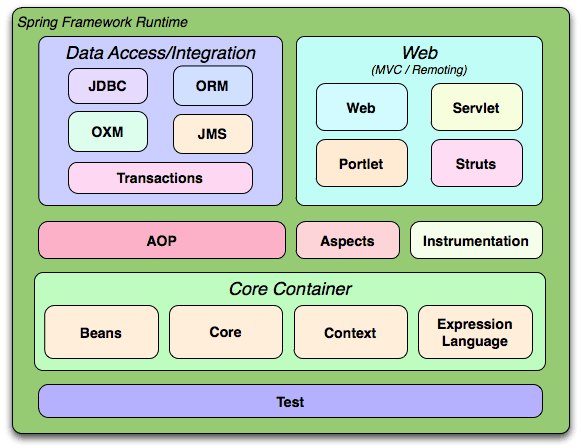
: 사용할 객체를 외부 컨테이너가 직접 생성 후 주입하여 사용하는 방식

방법) 생성자(권장), 필드 객체 선언, setter 매서드 이용

**④** 관점 지향 프로그래밍(AOP: Aspect Oriented Programming)

방법) 컴파일 과정에 삽입, 바이트코드를 메모리에 로드, 프락시 패턴 이용(스프링 제공)

**⑤** 스프링 프레임워크의 다양한 모듈



**2. 스프링 부트 (프레임워크에서 업그레이드)**

**①** 의존성 관리

: 호환되는 버전을 명시해야 했지만 spring-boot-starter를 통해 서로 호환되는 모듈 조합을 제공함.

**spring-boot-starter 라이브러리)**

-web: 스프링 MVC를 사용하는 RESTful 앱을 위한 의존성.

-test: Junit Jupiter, Mockito 등의 테스트용 라이브러리

-jdbc: JDBC(Java Data Base Connection) 기능 제공

-security: 인증, 권한, 인가 등의 기능 제공

-data-jpa: 하이버네이트를 활용한 JPA 기능 제공

-cache: 캐시 기능 제공

**②** 자동 설정

: 추가된 라이브러리를 이용하는데 필요한 환경 설정을 알아서 찾아준다.

**③** 내장 WAS(Web Application Server)

: 톰캣이 내장되어 있어 별 다른 설정 없이도 바로 실행 가능

**④** 모니터링

: 스프링 부트 액추에이터 도구 이용