# 섹션3-1 DI/IoC 개념 이해하기

## 단어적 의미

**D**ependency **I**njection - 의존성 주입 => 주입을 받는다.

**I**nversion **o**f **C**ontrol - 제어의 역전 => 제어의 주체가 바뀜

**D**

- Dependency 만 있는 구조
- 내가 Control 하는 구조

필요할 때 마다 객체를 생성함

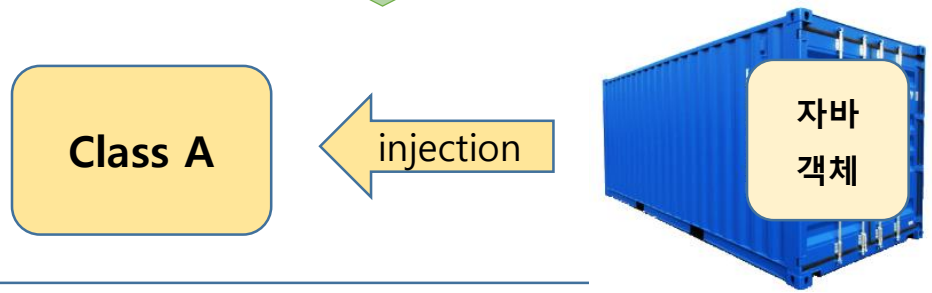
Classname name = new Cassname();


**DI**

- D도 있고 I도 있고
- 내가 Control 당하는 느낌

스프링이 구동될 때 이미 객체가 생성되어

필요로 하는 Class 가 수동적으로 연결을 받음



# 섹션3-2 View : Jsp vs Thymeleaf 어떤걸 선택할까?

## https://docs.spring.io/spring-boot/docs/2.1.5.RELEASE/reference/htmlsingle/#boot-features-jsp-limitations

### 29.4.5 JSP Limitations

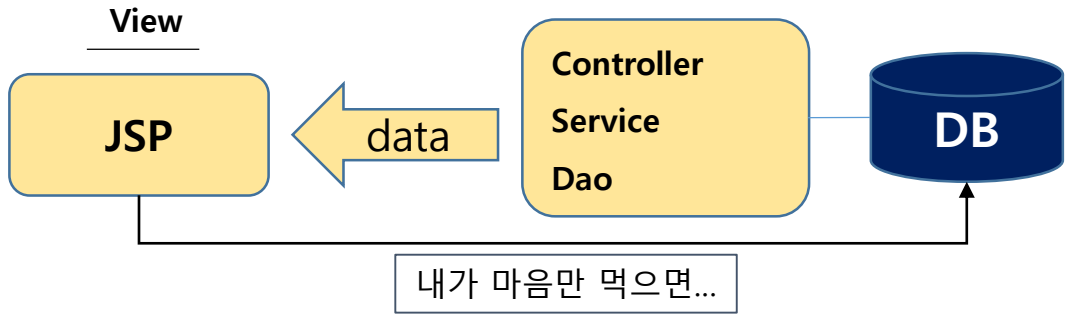
When running a Spring Boot application that uses an embedded servlet container (and is packaged as an executable archive), there are some limitations in the JSP support.

- With Jetty and Tomcat, it should work if you use war packaging. An executable war will work when launched with `java -jar`, and will also be deployable to any standard container. JSPs are not supported when using an executable jar.
- Undertow does not support JSPs.
- Creating a custom `error.jsp` page does not override the default view for error handling. Custom error pages should be used instead.

There is a JSP sample so that you can see how to set things up.

## https://docs.spring.io/spring-boot/docs/1.2.3.RELEASE/reference/html/boot-features-developing-web-applications.html

JSPs should be avoided if possible, there are several known limitations when using them with embedded servlet containers.



# 섹션3-3 Jar vs War 어떤걸 선택할까? (내장톰캣 vs 외장톰캣)