

23주차 발표_민서연

어노테이션

01 어노테이션

개념

어노테이션은 다른 프로그램에게 유용한 정보를 제공하기 위해 사용 되는 것으로 주석과 같은 의미를 가진다.

자바의 어노테이션은 소스코드에 추가해서 사용할 수 있는 메타 데이터의 일종이다.

보통 골뱅이(@) 기호를 앞에 붙여서 사용하며 JDK 1.5 버전 이상부터 사용이 가능하다.

메타 데이터 : 애플리케이션이 처리해야할 데이터가 아니라 컴파일 과정과 실행과정에서 코드를 어떻게 처리해야할지를 알려주기 위한 추가정보

01 어노테이션

역할

- 컴파일러에게 문법 에러를 체크하도록 정보를 제공한다.
- 프로그램을 빌드할 때 코드를 자동으로 생성할 수 있도록 정보를 제공한다.
- 런타임에 특정 기능을 실행하도록 정보를 제공한다.

01 어노테이션

종류

표준 어노테이션

@Override

컴파일러에게 메서드를 오버라이딩하는 것이라고 알림

@Deprecated

앞으로 사용하지 않을 대상임을 알림

@FunctionalInterface

함수형 인터페이스라는 것을 알림

@SuppressWarnings

컴파일러가 경고 메시지를 나타내지 않음

@SafeVarargs

제네릭과 같은 가변 인자의 매개변수를 사용할 때의
경고를 나타내지 않음

01 어노테이션

종류

메타 어노테이션

@Target

어노테이션을 정의할 때 적용 대상을 지정하는데 사용

@Documented

어노테이션 정보를 javadoc으로 작성된 문서에 포함

@Inherited

어노테이션이 하위 클래스에 상속 되도록 함

@Retention

어노테이션이 유지되는 기간을 정하기 위해 사용

@Repeatable

어노테이션을 반복해서 적용할 수 있도록 함

01 어노테이션

종류

사용자 정의 어노테이션 (커스텀 어노테이션)

```
public @interface CustomAnnotation {  
    int age() default 20;  
    String name();  
}
```

사용자가 개발의 편의를 위해 정의하는 어노테이션

어노테이션은 내부에 값을 가질 수 있으며, 값을 설정 할 수 있음.
값을 설정하기 위해서는 default 값 형태로 설정한다.

02 스프링 어노테이션

스프링 빈이란

스프링 컨테이너가 관리하는 자바 객체를 의미한다.

IoC가 적용이 되기 때문에 사용자는 직접 new 를 이용해 생성한 객체를 사용하지 않고, 스프링에 의거하여 관리 당하는 자바객체를 사용한다.
이때 이 객체가 bean이다.

빈 사용시 Configuration 어노테이션을 꼭 사용해야 한다.

제어의 역전(IoC)

: 객체의 생성 및 제어권을 사용자가 아닌 스프링에게 맡기는 것

02 스프링 어노테이션

어노테이션과 bean

bean을 사용하는 경우

유지 보수 과정에서 자주 변경되는 객체 등록 할 때

외부 라이브러리가 제공하는 객체를 사용할 때

어노테이션 (@Component)을 사용할 경우

유지 보수 과정에서 자주 변경되지 않는 객체 등록 할 때

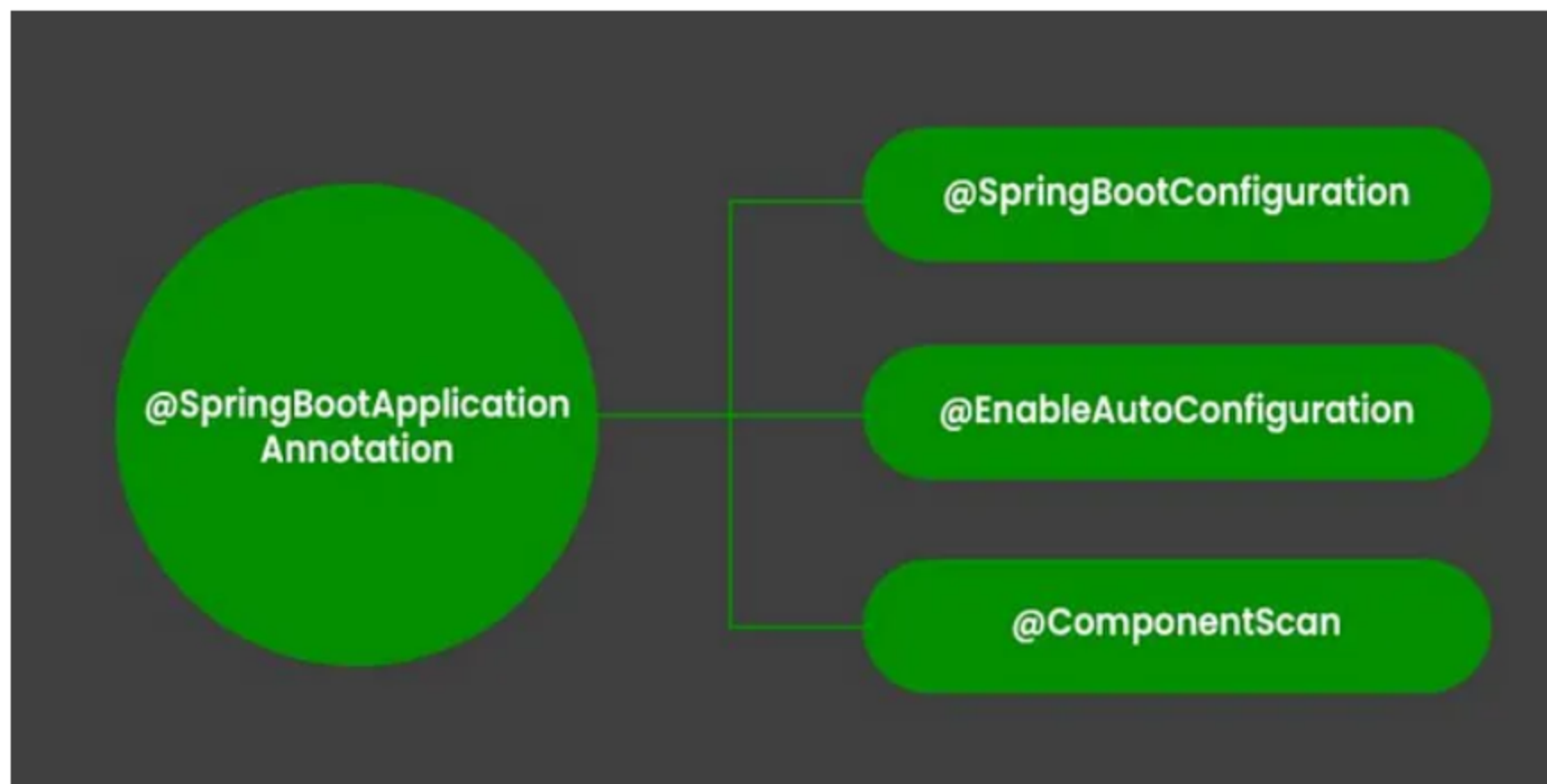
의존성 주입 할 때

@Component: 개발자가 직접 입력한 클래스를 bean으로 등록할 수 있는 방법

02 스프링 어노테이션

@SpringBootApplication @Configuration + @EnableAutoConfiguration + @ComponentScan

스프링부트를 자동으로 실행시켜주는 어노테이션



스프링 IOC Container에게 해당 클래스가 Bean 구성 클래스임을 알려주는 어노테이션

스프링 어플리케이션 컨텍스트를 만들때 자동으로 설정하는 기능들을 켜준다.

@Component, @Service, @Repository, @Controller, @Configuration 이 붙은 bean을 찾아서 context에 bean을 등록해 주는 어노테이션

02 스프링 어노테이션

@Resource, @Autowired, @Inject

모두 의존 관계를 자동으로 연결해서 bean을 주입해주는 기능을 가진 어노테이션

@RestController

@Controller + @ResponseBody 결합 형태의 어노테이션

기존 @Controller와 차이점

- @RestController: 주로 데이터를 리턴
- @Controller: 주로 view를 리턴

03 출처

https://velog.io/@falling_star3/Spring-Boot-%EC%8A%A4%ED%94%84%EB%A7%81-%EB%B9%88bean%EA%B3%BC-%EC%9D%98%EC%A1%B4%EA%B4%80%EA%B3%84
<https://sddev.tistory.com/225>
<https://prinha.tistory.com/entry/%EC%9E%90%EB%B0%94-%EC%8A%A4%ED%94%84%EB%A7%81-%EC%96%B4%EB%85%B8%ED%85%8C%EC%9D%B4%EC%85%98-annotation%EC%9D%98-%EC%A0%95%EC%9D%98%EC%99%80-%EC%A2%85%EB%A5%98>
<https://hbase.tistory.com/169>
<https://ittrue.tistory.com/156>
<https://bangu4.tistory.com/199>