

# 의존성 주입(DI)

김영래

테스트에 용이하다

결합도를 낮추면서 유연성과 확장성은 향상시킬 수 있다

재사용성이 높다

## 장점?

코드의 사용 이유 파악이 수월하며  
코드 가독성이 높아진다

객체간의 의존관계를 설정할 수 있다

코드를 단순화 시켜준다

종속성이 감소한다

종속적이던 코드의 수를 줄여준다

?

# 의존성 주입(Dependency Injection)

- Dependency(의존성)
- 의존성 주입이란?
- Android에서의 의존성 주입
- 처음 말했던 의존성 주입의 장점을 알아보시다!

**의존성 (Dependency)**

```
public class Car {  
  
    private Engine engine = new Engine();  
  
    public void start() {  
        engine.start();  
    }  
}
```

**Engine 클래스가 Car 클래스에 의존관계를 가진다!**

```
public class Main {  
  
    public static void main(String[] args) {  
        Car car = new Car();  
        car.start();  
    }  
}
```

# Dependency Injection

외부에서 객체를 생성해 주입시키는 것!



```
public class Car {  
  
    private final Engine engine;  
  
    public Car(Engine engine) {  
        this.engine = engine;  
    }  
  
    public void start() {  
        engine.start();  
    }  
}
```

## 생성자를 이용한 방법

```
public class Main {  
  
    public static void main(String[] args) {  
        Engine engine = new Engine();  
        Car car = new Car(engine);  
        car.start();  
    }  
}
```

```
public class Car {  
    private Engine engine;  
  
    public void setEngine(Engine engine) {  
        this.engine = engine;  
    }  
  
    public void start() {  
        engine.start();  
    }  
}
```

## setter를 이용한 방법

```
public class Main {  
  
    public static void main(String[] args) {  
        Car car = new Car();  
        car.setEngine(new Engine());  
        car.start();  
    }  
}
```

# DL in *Android*

# Dagger



테스트에 용이하다

결합도를 낮추면서 유연성과 확장성은 향상시킬 수 있다

재사용성이 높다

## 장점?

코드의 사용 이유 파악이 수월하며  
코드 가독성이 높아진다

객체간의 의존관계를 설정할 수 있다

코드를 단순화 시켜준다

종속성이 감소한다

종속적이던 코드의 수를 줄여준다



# Q & A

출처: <https://developer.android.com/training/dependency-injection>