**IoC(Inversion of Control)이란 무엇인가?**

자바가 등장하고 자바기반으로 애플리케이션을 개발하기 시작하던 최초의 시기에는 자바 객체를 생성하고, 객체간의 의존관계를 연결하는 제어권을 개발자가 직접 가지고 있었다.

그러나 서블릿, EJB가 등장하면서 개발자가 독점적으로 가지고 있었던 제어권이

서블릿, EJB를 관리하는 컨테이너로 넘어가 버렸다

::

이는 객체생성의 주체가 바뀌었다는 것을 의미하는 것 뿐만이 아니라 객체관의 연관성(주입) 객체의 생명주기를 관리하는 권한 또한 컨테이너가 전담할 수 밖에 없는 구조가 되어버린다.

::

객체 생성에서부터 생명주기의 관리까지의 모든 것을 컨테이너가 직접 관리하게 되는 것을 제어의 역전, 즉 IOC(Inversion Of Controller)라고 한다.

**IOC 기본 개념**

IOC ( Inversion of Control ) : 제어의 역행

* 단일 기능이 아닌 일반적인 자바의 컴포넌트는 협업객체와의 공동작업으로 해당 기능을 구현 ( coupling )
* 컴포넌트와의 관계를 해당 작업 컴포넌트가 직접 찾는 것이 아니라 컨테이너에게 부여 받음.(getBean())
* 객체에 대한 의존성을 컨테이너로부터 부여받음
* 의존하는 객체를 역행적으로 취득한다는 의미에서

DI ( Dependency Injection )이라고도 함.

Spring IOC의 철학

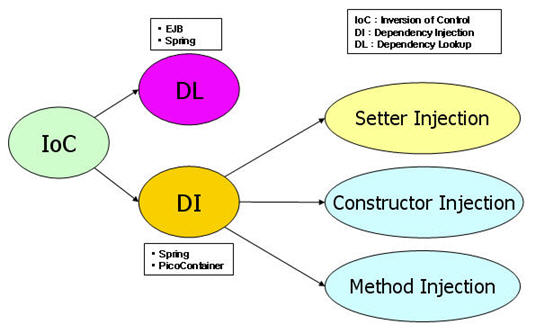
* **결합도를 낮추어 테스트와 재사용성을 높임.**
* **인터페이스를 이용하여** , 클라이언트 클래스에 영향을 주지 않고도 실제 구현 클래스를 유지보수 가능.

Bean 명명

* 모든 빈(bean) 은 하나 또는 그 이상의 id 값을 가진다.
* 모든 id값은 유일한 값이다.
* name도 사용가능

### IoC컨테이너의 분류

IoC 개념은 글을 쓰는 저자마다 다양한 방식으로 분류하고 있다. [Spring 프레임워크 워크북](http://kangcom.com/common/bookinfo/bookinfo.asp?sku=200601190003)에 따르면 다음과 같이 분류하고 있다.



#### DL(Dependency Lookup) 혹은 DP(Dependency Pull)

Dependency Lookup은 저장소에 저장되어 있는 빈(Bean)에 접근하기 위하여 개발자들이 컨테이너에서 제공하는 API를 이용하여사용하고자 하는 빈(Bean)을 Lookup하는 것을 말한다.

#### DI(Dependency Injection)

Dependency Injection은 Spring 프레임워크에서 지원하는 IoC의 형태이다.   
DI는 클래스 사이의 의존관계를 빈 설정 정보를 바탕으로 컨테이너가 자동적으로 연결해주는 것을 말한다.  
개발자들은 빈 설정 파일(저장소 관리 파일)에 의존관계가 필요하다는 정보를 추가하면 된다.  
Spring 프레임워크는 Setter Injection, Constructor Injection 의 두가지 유형으로 나타난다.

* **Setter Injection**  
  Setter Injection은 클래스 사이의 의존관계를 연결시키기 위하여 setter 메소드를 이용하는 방법이다. 예제코드는 다음과 같다.

class UserServiceImpl implements UserService{

private UserDAO dao;

public void setDao(UserDao dao){

this.dao = dao;

}

}

Class UserDAOImpl implements UserDAO{

Private DataSource ds;

Public void setDataSource(DataSource ds){

This.ds = ds;

}

* 위 코드를 보면 서비스 레이어는 퍼시스턴스 계층과의 통신을 위해 UserDAO 인터페이스와 의존관계가 형성된다.  
  소스코드 어디에도 UserDAO 구현 클래스의 정보와 인스턴스가 생성되는 곳이 없다.  
  그러나 위 클래스는 UserDAO 인스턴스가 존재한다는 가정하에 개발을 진행하고 있는데 이것이 가능한 이유는  
  아래와 같은 Spring 프레임워크의 빈 설정파일에 있다.

**<bean id=”service” class=”UserServiceImpl”>**

**<property name=”dao” ref=”dao” />**

**</bean>**

**Constructor Injection**Constructor Injection은 생성자를 이용하여 의존관계를 연결시킨다.  
위 소스를 Constructor Injection으로 변경하면 다음과 같다.

class UserServiceImpl implements UserService{

UserServiceImpl(UserDao dao){

}

빈 설정파일은 아래와 같이 작성된다.

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN"

"http://www.springframework.org/dtd/spring-beans.dtd">

**<bean id=”service” class=”UserServiceImp>**

**<contructor-args ref=”dao” />**

**</bean>**