一、控制反转：应用本身不负责依赖对象的创建及维护，依赖对象的创建及维护是由外部容器负责的。这样控制权就由应用转移到了外部容器，即控制权的转移。

依赖注入：在运行期，由外部容器动态地将依赖对象注入到组件中。

好处：

1、降低组件之间的耦合度，实现软件各层之间的解耦。

2、可以使容器提供众多服务如事务管理消息服务处理等等。

3、容器提供单例模式支持，开发人员不需要自己编写实现代码。

4、容器提供了AOP技术，利用它很容易实现如权限拦截，运行期监控等功能。

二、singleton：单例模式，Spring IoC容器中只会存在一个共享的Bean实例，无论有多少个Bean引用它，始终指向同一对象。Singleton作用域是Spring中的缺省作用域，也可以显示的将Bean定义为singleton模式

prototype:原型模式，每次通过Spring容器获取prototype定义的bean时，容器都将创建一个新的Bean实例，每个Bean实例都有自己的属性和状态，而singleton全局只有一个对象。根据经验，对有状态的bean使用prototype作用域，而对无状态的bean使用singleton作用域。

request：在一次Http请求中，容器会返回该Bean的同一实例。而对不同的Http请求则会产生新的Bean，而且该bean仅在当前Http Request内有效。

<bean id="loginAction" class="com.cnblogs.Login" scope="request"/>,针对每一次Http请求，Spring容器根据该bean的定义创建一个全新的实例，且该实例仅在当前Http请求内有效，而其它请求无法看到当前请求中状态的变化，当当前Http请求结束，该bean实例也将会被销毁。

三、

@Component取代<bean class="">

@Component(“id”) 取代 <bean id="" class="">

web开发，提供3个@Component注解衍生注解（功能一样）取代<bean class="">

@Repository ：dao层

@Service：service层

@Controller：web层。