在Spring中，Spring IOC提供了一个基本的JavaBean容器，通过**IoC模式管理依赖关系**，并通过**依赖注入**和**AOP切面**增强了为JavaBean这样的POJO对象赋予**事务管理**、**生命周期管理**等基本功能。

在Spring IoC容器的设计中，我们可以看到两个主要的容器系列，一个是实现**BeanFactory接口**的简单容器系列，这系列容器只实现了容器的最基本功能；另一个是**ApplicationzContext**应用上下文，它作为容器的高级形态而存在。

在Spring提供的基本IoC容器的接口定义和实现的基础上，**Spring通过定义BeanDefinition来管理基于Spring应用中的各种对象以及它们之间的相互依赖关系**。**BeanDefinition抽象了我们对Bean的定义**，是让容器起作用的主要数据类型。IoC容器是用来管理对象依赖关系的，对IoC容器来说，BeanDefinition就是对依赖反转模式中管理的对象依赖关系的数据抽象，也是容器实现依赖反转功能的核心数据结构，依赖反转功能都是围绕对这个BeanDefinition的处理来完成的。

BeanFactory是一个Factory，也就是IoC容器。FactoryBean是一个接口，里面有一个方法：getObject()，当某个实现了FactoryBean接口的类交给IoC容器管理时，如果调用getBean()方法，返回的是FactoryBean中getObject()方法返回的实例，只有在调用getBean()方法的时候在Bean名称前加上“$”，返回的才是IoC容器原来返回的对象。

ClassPathResource res = new ClassPathResource("beans.xml");  
DefaultListableBeanFactory factory = new DefaultListableBeanFactory();  
XmlBeanDefinitionReader reader = new XmlBeanDefinitionReader(factory);  
reader.loadBeanDefinitions(res);

这样，我们就可以通过factory对象来使用DefaultListableBeanFactory这个IoC容器了。在使用IoC容器时，需要如下几个步骤：

1. 创建IoC配置文件的抽象资源，这个抽象资源包含了BeanDefinition的定义信息。
2. 创建一个BeanFactory，这里使用DefaultListableBeanFactory。
3. 创建一个载入BeanDefinition的读取器，这里使用XmlBeanDefinitionReader来载入XML文件形式的BeanDefinition，通过一个**回调**配置给BeanFactory。
4. 从定义好的资源位置读入配置信息，具体的解析过程由XMLBeanDefinitionReader来完成。完成整个载入和注册Bean定义之后，需要的IoC容器就建立起来了。

IoC容器的初始化时由refresh()方法来启动的，这个方法标志着IoC容器的正式启动。这个启动包括**BeanDefinition**的**Resource定位**、**载入**和**注册**三个基本过程。Spring把这三个过程分开，并使用不同的模块来完成，如使用相应的ResourceLoader、BeanDefinitionReader等模块，通过这样的设计方式，可以**让用户更加灵活地对这三个过程进行剪裁或扩展，定义出最适合自己的IoC容器的初始化过程**。

第一个过程是**Resource定位过程**。这个Resource定位值得是BeanDefinition的资源定位，它是由ResourceLoader通过统一的Resource接口来完成，这个Resource对各种形式的BeanDefinition的使用都提供了统一接口。

第二个过程是**BeanDefinition的载入**。这个载入过程是把用户定义好的Bean表示成IoC容器内部的数据结构，而这个容器内部的数据结构就是BeanDefinition。这个BeanDefinition实际上就是POJO对象在IoC容器中的抽象，通过这个BeanDefinition定义的数据结构，使IoC容器能够方便地对POJO对象也就是Bean进行管理。

第三个过程是**向IoC容器注册这些BeanDefinition的过程**。这个过程是调用BeanDefinitionRegistry接口的实现来完成的。这个注册过程把载入过程中解析得到的BeanDefinition向IoC容器进行注册。在IoC容器内部将BeanDefinition注入到一个HashMap中去，IoC容器就是通过这个HashMap来持有这些BeanDefinition数据的。

在IoC容器初始化过程中农，一般不包含Bean依赖注入的实现。在Spring IoC的设计中，**Bean定义的载入和依赖注入是两个独立的过程**。依赖注入一般发生在应用第一次通过getBean向容器索取Bean的时候。