# spring

springboot 其实很简单，就是整合我们常用的哪些框架

微服务比较方便。

2008年，遇到基建问题 断电。火车延迟 过年

基础很重要 web基础<-java基础<-servlert+jdbc<-SSM框架<-springboot

是以ioc/aop起家的

1000页

spring书籍 300-500页面

但是看的时间很中

比较难理解

A a=new A()

B b=new B()

自己要去构造对象

这些对象让spring容器来管理 创建销毁

我们去容器中获取就可以了

A a=spring.get(“a”);

B b=spring.get(“b”);

控制对象的创建，交由spring容器来管理

反转了、

ioc/di

控制反转/依赖注入、

从某种角度来看，确实是一个东西，但是还是有很大区别

ioc/aop

对应的java

单例模式 / 工厂模式 / 代理模式

这两种模式的提现

通过反射机制+接口模式 实现工厂模式

工厂模式的目的是什么

为了以后更换业务更加方便

通过配置文件

自己要手写 工厂模式，比较繁琐，还有很多东西需要考虑的

spring容器就帮我们做了这个事情

ioc 这个容器

通过书写简单的配置文件即可

下载jar包

基础框架中，自己导入jar

maven通过依赖导入

5个jar 一个个找

依赖一个就可以了，会自动去找其他4个依赖的jar

强调 工厂模式，配合接口一起使用

spring容器控制的话。变得更加简单了。

软件春天的到来

spring 春天的意思

ioc aop springmvc springdata springboot 等等

获取bean的其他方式

单例模式

只有一个对象

注入数据 依赖注入的概念 di

什么是注入？

注入的类型有哪些？

注入方式有哪些？

ioc/di

注入的是对象，通过属性中的set方法注入

组装

对象中的各个组件自己从容器中获取，再自己组装

还是不方便

买车

买车架

买发动机

买变速箱

买轮胎

买刹车总成

买底盘

等等

买车，这个车已经帮我们组装好了。方便了

买车的时候自己定制

车->配件通过注入的方式

发动机注入到这个车

轮胎注入车

刹车注入车

其它等注入到车中

拿车就Ok了

什么是依赖注入

和控制反转还是有很大区别的。但是有共同的概念

## 注入的方式

属性注入，并不是属性，而是通过set方法注入的。

严格按照规范

mybatis 通过属性的set/get操作，但是新的版本直接访问属性，设置反射访问控制 true

默认会调用一个无参数的构造函数的

类设计 有一个有参的构造函数。那么jvm不会自动生成一个无参数构造函数的。

外部注入 配置构造函数注入或者是属性注入方式

主要是用在对象上面

基本类型数据，可以直接value

内部注入 需要配合构造注入还有一个属性注入

属性

构造

如果是引用类型，分外部注入，内部注入方式

下课的时候，可以自己休息一会，如果遇到老师还没有开始上课，那么自己把上课的内容自己练习一下。

有问题上课的时候就可以反馈

可能一部分同学没有看到录播中讲的这个事情。

休息5分钟，复习复习，把上课的代码练练。

注解 就是代码的注释，这个注释是给java去看，不是给程序员看的。

java通过反射机制 获取这个类中的所有东西，包括注解

注解充当配置文件

目前是半注解功能。

配置对象 也用注解 先不用 @component 组件

获取对象 也是用注解 先获取对象使用注解@autowired

不管是配置文件也好，还是注解也好，对象都是在容器中管理的

@Autowired

注解的意思就是从容器中获中获取对象，赋值给当前这个属性

@component

定义组件 可以设置名字，如果没有名字，可以和类名一样，小写，还可以通过类型获取这个对象

注解配置，配置的是对象引用

而且非数据层的对象

实体层，是不应该用IOC管理

new 构造

service

dao

纠正一下，应该是在service dao中用到ioc

## 注入的类型

字符串类型

对象类型