# 配置文件四个位置

优先级高到低

* **file:./config/ （项目根目录下config文件夹下）**
* **file:./ （项目根目录下）**

**以上两个打包后消失**

* **classpath:/config/ （类路径下config文件夹下）**
* **classpath:/ （类路径下）**

四个位置的配置文件都会被加载，可以形成**互补配置**

相同配置项，高优先级会覆盖低优先级

# 改变默认配置文件位置

使用spring.config.location来改变默认位置

一般在命令行中使用，当项目被打包发布后，希望修改一些配置的时候，就可以通过命令行参数的方式，启动项目的时候来指定配置文件的新位置，**指定配置文件和默认加载的这些配置文件形成互补配置，指定配置文件优先级最高。**



**java -jar springboot-2-quick-0.0.1-SNAPSHOT.jar --spring.config.location=**

**D:/application.yml**

**这里路径的斜线要用/正斜线，因为\反斜线表示转义符**

# 外部配置文件加载顺序

优先级高到低，**相同项，高覆盖低，不同项，互补，共同起作用**

1. **命令行参数**

直接指定某一项，多项用空格间隔，

**如 --server.port --server.servlet.context-path=/**



1. 来自java:comp/env的JNDI属性
2. Java系统属性（System.getProperties()）
3. 操作系统环境变量
4. RandomValuePropertySource配置的random.\*属性值

**如果指定了profile，带profile的优先（多profile文件）**

**如果jar包外有配置文件，jar包外的优先**

**jar包外指：和jar包同一路径下的配置文件，在包外部，相当于工程下**

**如，文件夹下 springboot.jar application.yml（该配置文件会被加载）**

1. **jar包外部的application-{profile}.properties或application.yml(带spring.profile)配置文件**
2. **jar包内部的application-{profile}.properties或application.yml(带spring.profile)配置文件**
3. **jar包外部的application.properties或application.yml(不带spring.profile)配置文件**
4. **jar包内部的application.properties或application.yml(不带spring.profile)配置文件**
5. @Configuration注解类上的@PropertySource
6. 通过SpringApplication.setDefaultProperties指定的默认属性