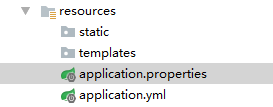
# 配置文件

## 全局的配置文件（properties，yml）

Spring Boot使用一个全局的配置文件：application.properties；application.yml

配置文件的作用：修改SpringBoot自动配置的默认值；SpringBoot在底层都给我们自动配置好；

# YAML标记语言

## Yaml

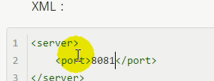
YAML（YAML Ain't Markup Language）

​ YAML A Markup Language：是一个标记语言

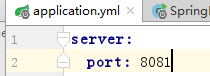
​ YAML isn't Markup Language：不是一个标记语言；

标记语言：

​ 以前的配置文件；大多都使用的是 \*\*xxxx.xml\*\*文件；

​ YAML：以数据为中心，比json、xml等更适合做配置文件；配置例子



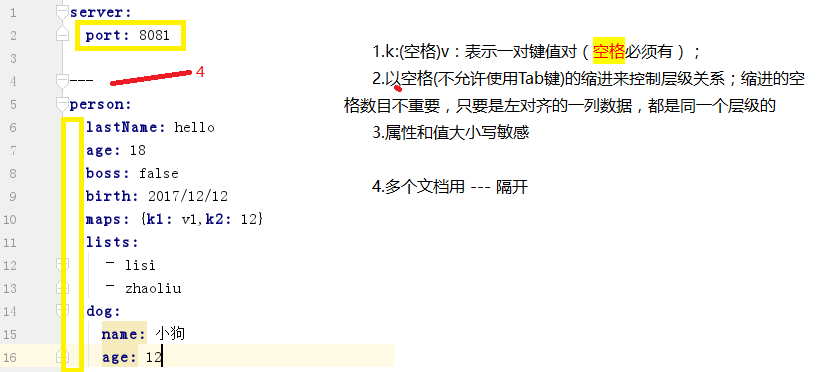
## YAML基本语法

k:(空格)v：表示一对键值对（空格必须有）；

以空格(不允许使用Tab键)的缩进来控制层级关系；缩进的空格数目不重要，只要是左对齐的一列数据，都是同一个层级的

属性和值大小写敏感

多个文档用 --- 隔开



## 属性值的写法

### 字面量

**普通的值（数字，字符串，布尔）**

k: v：字面直接来写；

字符串默认不使用引号；

""：双引号；不会转义字符串里面的特殊字符；特殊字符会作为本身想表示的意思

name: "zhangsan \n lisi"：输出；zhangsan 换行 lisi

''：单引号；会转义特殊字符，特殊字符最终只是一个普通的字符串数据

name: ‘zhangsan \n lisi’：输出；zhangsan \n lisi

### 对象、Map（属性和值）（键值对的集合）：

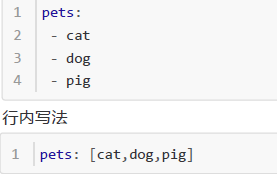
k: v：在下一行来写对象的属性和值的关系；对象的一组键值对，使用冒号分隔。如：username: admin，冒号后面跟空格来分开键值；注意缩进

{k: v}是行内写法



### 数组（List、Set）：

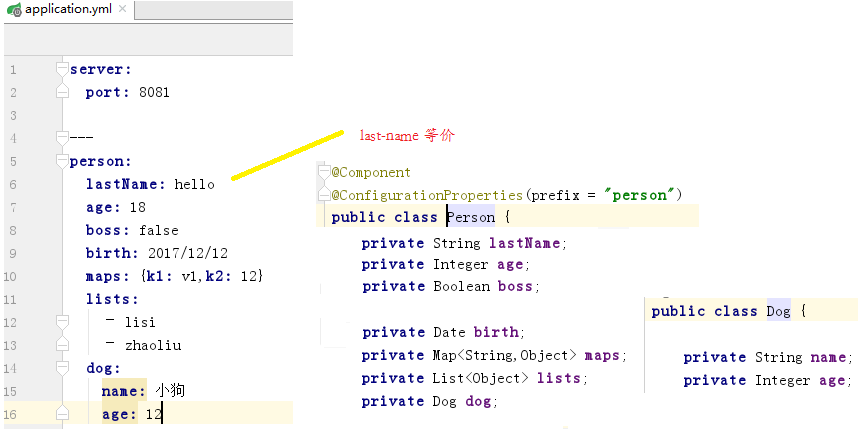
用- 值表示数组中的一个元素，[]为行内写法



可复合使用：https://nodeca.github.io/js-yaml/

## application.yml

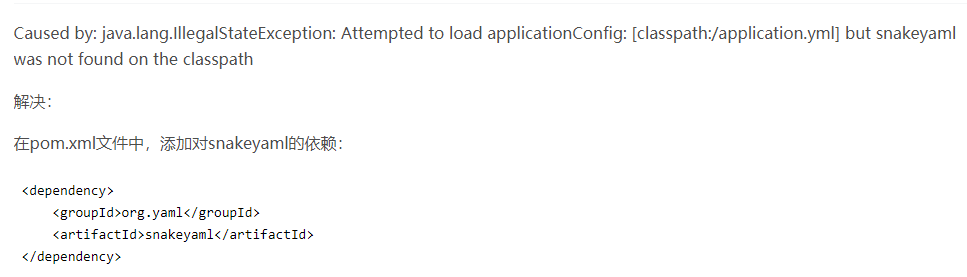
看文件：application.yml



## 问题

### Yml运行报错报错：[classpath:/application.yml] but snakeyaml was not found on the classpath

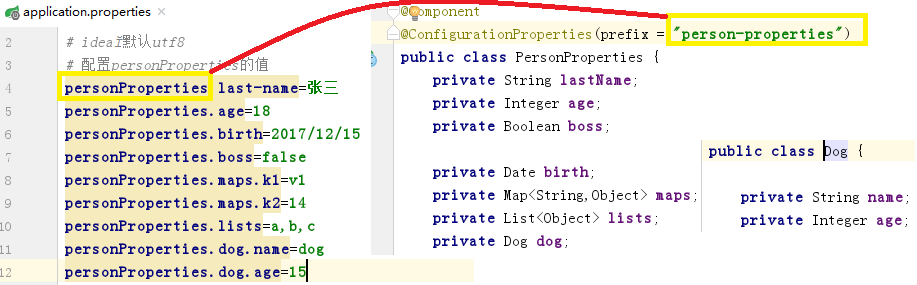
https://blog.csdn.net/jingjing9823/article/details/98439646



# .properties文件

## application.properties

看文件：application.propreties



## properties配置乱码

properties配置文件在idea中默认utf-8可能会乱码

