作者：赵扶摇  
链接：https://www.zhihu.com/question/41253282/answer/119857880  
来源：知乎  
著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

HOCON(Human-Optimized Config Object Notation)  
是由typesafe（开发scala和play framework的公司）主导的项目：  
[GitHub - typesafehub/config: Configuration library for JVM languages](https://link.zhihu.com/?target=https%3A//github.com/typesafehub/config)

在Google干过的同学可以参考GCL，会发现很多设计上的类似点。

我觉得比较完美的配置文件格式需有这些特性

1. 语法要简单，灵活

简单大家都差不多

HOCON是JSON和java property的超集，最为灵活，你可以写

a: {

b: {

c: 3

d: 4

}

}

也可以单独写

a.b.c=5

":"号也可以换成"="或者就完全省略

2. 能够写注释，这点json简直可悲，不过可以考虑加预处理的过程

3. 能够比较方便的覆盖参数值（方便书写或者debug），比如说在config文件中定义了  
a=1

可以在运行的时候，通过类似 program -Da=2或者a=2 program的的方式来覆盖参数值，而不需要跑去修改配置文件本身

这一点HOCON完爆其他的几种

4. 能够重用配置片段，比较大一点的project中，经常有很多地方的配置需要保持一致，最好的办法就是引入变量和引用的概念，比如可以类似

db\_connection: {host: a, password: b, db\_name: c, ..... }

service\_a: {

host: yyy

db: $db\_connection

}

service\_b: {

host: yyy

db: $db\_connection

}

这样最大的好处是避免了copy and paste，在修改配置文件的时候能搞保证不出问题

这点yaml和hocon基本上都是做的不错的，json没有，ini我用的不多，好像是没有。  
yaml的实现其实比较简单，就是单纯的文本替换，这样导致我要说的下一点被HOCON完爆。

5，可以继承  
这是HOCON完爆其他语言的地方。还是上面那个例子，假设service\_b.db不仅仅是是要是用全局db\_connection的值，要稍微修改其中host的值，可以

service\_b: {

host: yyy

db: $db\_connection {

host: abc

}

}

而且不需要重新copy and paste之前所有的内容

它的继承非常强大，语义可以说基本上和普通OO语言没有太大区别

另外HOCON可以包含文件，比如说你可以写一个基础的配置文件base.conf，然后再针对dev，staging和production分别做一份不同的文件，这样可以很轻松地做到在不同的环境下，用不同的配置而且没有重复的配置代码，比如说

include 'base.conf'

// 覆盖默认值

db.connection = "product\_machine:2000"

....

另外提一下twitter之前尝试用scala语言本身来当config language，但是好像过于笨重