[json简介 1](#_Toc2432615)

[yaml简介 2](#_Toc2432616)

[jinja2模版简介 6](#_Toc2432617)

# json简介

• json 是什么?

– json 是 JavaScript 对象表示法,它是一种基于文本,独立于语言的轻量级数据交换格式。

– JSON中的分隔符限于单引号 ' 、小括号 ()、中括号[ ]、大括号 { } 、冒号 : 和逗号 ,

• json 特性

– JSON 是纯文本

– JSON 具有"自我描述性"(人类可读)

– JSON 具有层级结构(值中存在值)

– JSON 可通过 JavaScript 进行解析

• json 语法规则

– 数据在名称/值对中

– 数据由逗号分隔

– 大括号保存对象

– 中括号保存数组

• json 数据的书写格式是:名称/值对。

– 名称/值对包括字段名称(在双引号中),后面写一个

冒号,然后是值,例如:

"漂亮姐" : "庞丽静"

• json 语法规则之数组

{ "讲师":

["牛犇", "丁丁", "静静","李欣"]

}

• 复合复杂类型

{ "讲师":

[ {"牛犇":"小逗逼", "负责阶段":"1"},

{"丁丁":"老逗逼", "负责阶段":"2"},

{"静静":"漂亮姐", "负责阶段":"3"},

{"李欣":"老司机", "负责阶段":"4"}

]

}

json 就是键值对{}和数组[]之间来回嵌套

# yaml简介

• yaml 是什么

– 是一个可读性高,用来表达数据序列的格式。

– YAML:YAML Ain't Markup Language

– YAML参考了其他多种语言,包括:C语言、Python、Perl,并从XML、电子邮件的数据格式(RFC 2822)中获得灵感。

Clark Evans在2001年首次发表了这种语言[1],另外Ingy döt Net与Oren Ben-Kiki也是这语言的共同设计者[2]。

目前已经有数种编程语言或脚本语言支持(或者说解析)这种语言。

• yaml 基础语法

– YAML的结构通过空格来展示

– 数组使用"- "来表示（都有个空格）

– 键值对使用": "来表示（都有个空格）

– YAML使用一个固定的缩进风格表示数据层级结构关系

– 一般每个缩进级别由两个以上空格组成

– # 表示注释

• 注意:

– 不要使用tab,缩进是初学者容易出错的地方之一

– 同一层级缩进必须对齐

• YAML的键值表示方法

– 采用冒号分隔

– : 后面必须有一个空格

– YAML键值对例子

"庞丽静": "漂亮姐"

– 或

"庞丽静":

"漂亮姐"

– 复杂YAML的键值对嵌套

"讲师":

"庞丽静": "漂亮姐" （这一行前面是2个空格）

– 或

"讲师":

"庞丽静": （这一行前面2个空格）

"漂亮姐" （这一行前面4个空格）

– 数组

["牛犇", "丁丁", "静静", "李欣"]

• YAML 数组表示方法

– 使用一个短横杠加一个空格

– YAML 数组例子

- "牛犇"

- "丁丁"

- "静静"

- "李欣"

– 哈希数组复合表达式

"讲师":

- "牛犇"

- "丁丁"

- "静静"

- "李欣"

– 高级复合表达式

"讲师":

-

"牛犇": "小逗比"

"阶段": 1

-

"丁丁": "老逗比"

"阶段": 2

-

"静静": "漂亮姐"

"阶段": 3

-

"李欣": "老司机"

"阶段": 4

区别table和空格的方法

给空格刷颜色，或者给table刷颜色就能看出来，

在命令模式下

/\t 给table刷颜色

/\ 给空格刷颜色

• yaml高级语法

– | 与 > 表示对应的值为多行字符, > 与 | 的区别是会把 \n 转换为空格

– ! 可以设置类型,!! 可以强制类型转换

– 为了维持文件的简洁,并避免数据输入的错误,YAML提供了结点参考(\*)和散列合并(<<)参考到其他

结点标签的锚点标记(&)。参考会将树状结构加入锚点标记的内容,并可以在所有数据结构中运作,合并

只有散列表可以使用,可以将键值自锚点标记复制到指定的散列表中

# jinja2模版简介

• jinja2 是什么

– Jinja2是基于python的模板引擎,包含变量和表达式两部分,这两者在模板求值的时候会被替换为值。

模板中还有标签,控制模板的逻辑。

• 为什么要学习 jinja2 模版

– 要使用 ansible 就要深入学习 playbook 配置及模板。playbook 的模板使用 python 的 jinja2 模块来处理的

• jinja2 模版基本语法

– 模板的表达式都是包含在分隔符 "{{ }}" 内的;

– 控制语句都是包含在分隔符 "{% %}" 内的;

– 另外,模板也支持注释,都是包含在分隔符 "{# #}"

内,支持块注释。

– 调用变量

{{varname}}

– 计算

{{2+3}}

– 判断

{{1 in [1,2,3]}}

• jinja2 模版控制语句

{% if name == '小逗逼' %}

讲故事,吹牛B

{% elif name == '老逗逼' %}

黑丝(82年的)

{% elif name == '漂亮姐' %}

约会

{% else %}

沉迷学习,无法自拔

{% endif %}

{% if name == ... ... %}

... ...

{% elif name == '漂亮姐' %}

{% for method in [约会, 逛街, 吃饭, 看电影, 去宾馆] %}

{{do method}}

{% endfor %}

... ...

{% endif %}

• jinja2 过滤器

– 变量可以通过 过滤器 修改。过滤器与变量用管道符号

( | )分割,并且也 可以用圆括号传递可选参数。多

个过滤器可以链式调用,前一个过滤器的输出会被作

为 后一个过滤器的输入。

– 例如:

– 把一个列表用逗号连接起来: {{ list|join(', ') }}

– 过滤器这里不一一列举,需要的可以查询在线文档

http://docs.jinkan.org/docs/jinja2/templates.html

#builtin-filters