# Yaml文件

相比较ini、csv等文件，用Yaml存储数据的效率更高，特别是结构化的数据。Yaml中数据结构是怎么样的，那么读取出来就是怎么样的。Yaml主要支持字典、数组、还有纯量，字典用：冒号表示，数组用-横杠表示。Yaml的基本语法：1.大小写敏感；2.使用缩进表示层级关系；3.不允许使用 TAB 键来缩进，只允许使用空格键来缩进；4.缩进的空格数量不重要；5.使用"#"来表示注释。Python处理 Yaml 格式的数据需要先下载pyyaml（pip install pyyaml），然后import yaml进行读取yaml文件。例如：

import yaml  
with open('./readyaml.yaml','r',encoding='utf8') as file:  
 data=yaml.load(stream=file,Loader=yaml.FullLoader)

## 字典

Yaml中用：冒号代表字典，其中冒号与值之间有个空格。例如：

name: boy  
address: 浙江省杭州市  
age: 18

用python读取出来的值为：{'name': 'boy', 'address': '浙江省杭州市', 'age': 18}

也可以在字典中套用字典：

message:  
 name: boy  
 address: 浙江省杭州市  
 age: 18

（除了第一行下面的几行需要缩进一个空格）或者

message: {'name': 'boy', 'address': '浙江省杭州市', 'age': 18}

结果都是：{'message': {'name': 'boy', 'address': '浙江省杭州市', 'age': 18}}

当然也可以：

message1:  
 name: boy  
 address: 浙江省杭州市  
 age: 18  
message2:  
 name: tom  
 address: 浙江省温州市  
 age: 20

结果为：{'message1': {'name': 'boy', 'address': '浙江省杭州市', 'age': 18}, 'message2': {'name': 'tom', 'address': '浙江省温州市', 'age': 20}}。未缩进的的空格为同一级，缩进一个空格降一级，以此类推。

## 数组

Yaml中用-横杠代表数组，其中横杠与值之间有个空格。例如：

- a  
- b  
- c

用python读取出来的值为：['a', 'b', 'c']。

列表中的值可以放其他结构的数据：

-  
 - a  
 - b  
 - c  
-  
 - d  
 - e  
 - f

（除了第一行下面的几行需要缩进一个空格）或者

- [a,b,c]  
- [d,e,f]

结果都是：[['a', 'b', 'c'], ['d', 'e', 'f']]。也可以跟字典结合起来使用：

-  
 a: 1  
 b: 2  
- {'c':3,'d':4}  
- e: 5

结果为：[{'a': 1, 'b': 2}, {'c': 3, 'd': 4}, {'e': 5}]。

如果[]或者{}仅仅表示字符，那么需要用引号括起来，要不然会报错：

- [1,2,3,4,'[a]']  
- [{1,2,3,4,'a{}'}]

## 纯量

纯量是最基本的、不可再分的值，以下数据类型都是常见的纯量：字符串、布尔值、整数、浮点数、null、时间、日期。

number: 15.01  
string: hi  
bool: true  
nothing: ~  
date: 2018-01-01  
time: 2018-01-01 12:12:12

转化为Python为：

{'number': 15.01,   
 'string': 'hi',   
 'bool': True,   
 'nothing': None,   
 'date': datetime.date(2018, 1, 1),   
 'time': datetime.datetime(2018, 1, 1, 12, 12, 12)}