什么是YAML

YAML参考了其他多种语言，包括：XML、C语言、Python、Perl以及电子邮件格式RFC2822。 Clark Evans在2001年5月在首次发表了这种语言，另外Ingy döt Net与Oren Ben-Kiki也是这语言的共同设计者。

YAML是"YAML Ain't a Markup Language"（YAML不是一种置标语言）的递归缩写。 在开发的这种语言时，YAML 的意思其实是："Yet Another Markup Language"（仍是一种置标语言），

格式及示例

数据结构可以用类似大纲的缩排方式呈现，结构通过缩进来表示，连续的项目通过减号"-"来表示，map结构里面的key/value对用冒号":"来分隔。

示例如下：

```
开源优测信息:
    微信公众号:
        公众号: DeepTest
        中文名: 开源优测
        创建者: 苦叶子
        内容系列:
            - 接口测试
            - jmeter
            - selenium
            - 快学python3
            - 大数据测试
            - 杂谈系列
    web站点:
        中文名: 开源优测社区
        状态: 已暂停
        城市: 广州
        网址: www.testingunion.com
```

注意：

1. 字符串不一定要用双引号标识
2. 在缩排中空白字符的数目并不是非常重要，只要相同阶层的元素左侧对齐就可以了（*不过不能使用TAB字符*）
3. 允许在文件中加入选择性的空行，以增加可读性
4. 在一个档案中，可同时包含多个文件，并用"——"分隔
5. 选择性的符号"..."可以用来表示档案结尾（在利用串流的通讯中，这非常有用，可以在不关闭串流的情况下，发送结束讯号）

PyYaml

PyYAML是一个Python的YAML解析器。

如何安装? 请使用如下命令进行安装

```
pip install PyYaml
```

官方文档地址：

```
http://pyyaml.org/wiki/PyYAMLDocumentation
```

python yaml几个示例

下面先看一个如何将yaml格式的字符串转换成字典，再将字典转换成字符串的示例：

```
#-*- coding:utf-8 -*-

__author__ = "苦叶子"

import yaml

if __name__ == "__main__":
    print("python yaml基本示例")

    document = """
    公众号: 开源优测
    基本信息:
        创建人: 苦叶子
        id: DeepTest
    """

    # 将yaml格式内容 转换成 dict类型
    load = yaml.load(document)
    print(type(load))
    print(load)

    print("---" * 25)
    # 将python对象转换成为yaml格式文档
    output = yaml.dump(load)
    print(type(output))
    print(output)
```

说明： load：将yaml格式的字符串转换成Python对象 dump：将Python对象转换成yaml格式文档

多段yaml格式内容解析用用到load_all函数，示例如下：

```
#-*- coding:utf-8 -*-

__author__ = "苦叶子"

import yaml
```

```
import codecs

if __name__ == "__main__":
    print("python yaml基本示例")

    fp = codecs.open("yaml_data.yaml", "r", "utf-8")
    document = fp.read()
    fp.close()

    # 将yaml格式内容 转换成 dict类型
    load = yaml.load_all(document)

    # 遍历迭代器
    for data in load:
        print(type(data))
        print(data)

        print("---" * 25)
        # 将python对象转换成为yaml格式文档
        output = yaml.dump(data)
        print(type(output))
        print(output)
```

说明 load_all返回的是一个迭代器对象，需要自己去遍历获取每一个段的转换后才python对象。

请自己对比上述两个示例的一些细节区别，加强对yaml应用和基本解析的理解。

小结

当然了python和yaml的结合还有很多很多的特性，例如在yaml格式的内容里直接写python类型等等，这些大家都可以去自行研究。

就我们后续要用的先掌握这些暂时足够，对于更深入的，届时再结合实际的应用进行演示。

扫一扫关注微信公众号：