



链滴

用 Python 批量创建思源笔记的 DailyNote

作者: [Syngo](#)

原文链接: <https://ld246.com/article/1645957014539>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

起因

经过一个多月的使用，已逐渐适应并喜欢上写思源的dailynote，除了待办、摘录和总结外，当天剪的文章会直接用 `<kbd>Alt + [</kbd>` 保存为子文档，书写格式和流程基本固化。

因我的某象已经沦为“垃圾场”，不适合批量导出导入，如想把之前年度的笔记进行手工“清洁”迁，需要新建大量dailynote文档。

过程

研究思源笔记本地文件

思源笔记的本地文件结构

打开本地目录发现，主要结构为笔记本文件夹——文档，和文档文件夹——子文档两种。

```
data
├── .siyuan
├── yyyyymmddhhmmss-随机7位字母和数字组合 # 笔记本文件夹，如对应daily note笔记本
│   ├── .siyuan
│   ├── yyyyymmddhhmmss-随机7位字母和数字组合 # 一级文档文件夹，如对应2020子文档文件夹
│   └── yyyyymmddhhmmss-随机7位字母和数字组合.sy # 一级文档，如对应2020文档
│       ├── yyyyymmddhhmmss-随机7位字母和数字组合 # 二级文档文件夹，如对应01月子文档文件夹
│       └── yyyyymmddhhmmss-随机7位字母和数字组合.sy # 二级文档，如对应01月文档
│           └── yyyyymmddhhmmss-随机7位字母和数字组合.sy # 三级文档，如对应2020-01-01文档
├── assets
├── emojis
├── icon
├── templates
└── widgets
```

思源笔记的本地文件

打开几个对应仅录入了文档名的本地文件，发现不同的主要为文档的ID、文档名、段落的ID和更新时间戳。其中删改文档内容时，Children对象的也会变化，考虑是否可先省略。

```
{
  "ID": "yyyyymmddhhmmss-随机7位字母和数字组合", # 文档的ID
  "Type": "NodeDocument",
  "Properties": {
    "id": "yyyyymmddhhmmss-随机7位字母和数字组合", # 文档的ID
    "title": "文档名",
    "updated": "yyyyymmddhhmmss" # 更新时间戳
  },
  "Children": [
    {
      "ID": "yyyyymmddhhmmss-随机7位字母和数字组合", # 段落的ID
```

```

        "Type": "NodeParagraph",
        "Properties": {
            "id": "yyyyymmddhhmmss-随机7位字母和数字组合" # 段落的ID
            "updated": "yyyyymmddhhmmss" # 更新时间戳
        }
    }
]
}

```

ID的差异化

当创建2020-01-01文档时，实际需要分别先创建2020和01文档和文件夹，3个文档ID的前缀yyyyymm都是20200101。为了有所区分，对秒数进行了递进，同时方便后续按ID排序时不乱序。

文档名	ID前缀
2020	2020010101010101
01	2020010101010102
2020-01-01	2020010101010103

研究Python代码

随机字符串相关

```

string.ascii_lowercase # 小写字母
string.digits # 数字
random.sample(population, k) # 无重复的随机抽样
str.join(sequence) # 将序列中的元素以指定的字符连接生成一个新的字符串

```

日期时间相关

```

datetime.date(year, month, day) # 一个理想化的简单型日期
datetime.datetime.now() # 返回当前时间
strftime(format) # 根据给定的格式将对象转换为字符串
    %Y # yyyy, 年份
    %m # mm, 补零月份
    %d # dd, 补零日
    %H # hh, 补零24小时制
    %M # mm, 补零分钟
    %S # ss, 补零秒
dateutil.relativedelta.relativedelta() # 计算时间间隔，支持年月日等参数

```

文件相关

```

f = open(file, mode)
    "w" # 写入，不存在则新建
    "x" # 创建，已存在则报错
f.write() # 文件写入
f.close() # 文件关闭
os.mkdir(path) # 创建文件夹

```

JSON相关

```
json.dumps(obj, indent=4) # 输出 JSON 格式, 并缩进四个空格
```

消息框相关

```
tkinter.messagebox.showinfo(title, message) # 弹出消息提示框
```

输出代码

```
import string
import random
import os
import json
import tkinter
import tkinter.messagebox
import sys

from datetime import *
from dateutil.relativedelta import *

start_date = date(2020, 1, 1) # 需修改起始日期, 不能跨年
end_date = date(2020, 12, 31) # 需修改结束日期, 不能跨年

# 确定随机数长度为 7
r_num = 7
# 随机数组合为小写字母和数字
token = string.ascii_lowercase + string.digits
# 生成年度文档随机数
token_y = ".join(random.sample(token,r_num))

# 生成年度文档 ID, 时间默认为当年 1 月 1 日 01 点 01 分 01 秒
id_y = start_date.strftime('%Y') + "0101010101-" + token_y

# 生成年度文档内容
data_y = {
    "ID": id_y,
    "Type": "NodeDocument",
    "Properties": {
        "id": id_y,
        # 标题格式 yyyy
        "title": start_date.strftime('%Y'),
        # 更新时间为当下时间, 格式 yyymmdhhmmss
        "updated": datetime.now().strftime('%Y%m%d%H%M%S')
    }
}
path_k = "results"
if not os.path.exists(path_k):
    os.mkdir(path_k)
else:
    tkinter.messagebox.showinfo("提示", "请先删除 " + path_k + " 文件夹后再重新运行")
    sys.exit()
```

```

# 创建年度文档
filename_y = path_k + "/" + id_y + ".sy"
f = open(filename_y, "x")
# 输出 JSON 格式, 并缩进四个空格
f.write(json.dumps(data_y, indent=4))
# 关闭文件
f.close()
# 创建年度文档文件夹
os.mkdir(path_k + "/" + id_y)

# 计算起始和结束日期月份差
delta_m = end_date.month - start_date.month + 1

# 月份循环 i 次
for i in range(delta_m):
    day_m = start_date + relativedelta(months=+i)
    token_m = ''.join(random.sample(token, r_num))
    # 生成月度文档 ID, 时间默认为当月 1 日 01 点 01 分 02 秒
    id_m = day_m.strftime('%Y%m%d') + "010102-" + token_m
    data_m = {
        "ID": id_m,
        "Type": "NodeDocument",
        "Properties": {
            "id": id_m,
            # 标题格式 mm
            "title": day_m.strftime('%m'),
            "updated": datetime.now().strftime('%Y%m%d%H%M%S')
        }
    }
}
# 在年度文件夹下, 创建月度文档
filename_m = path_k + "/" + id_y + "/" + id_m + ".sy"
f = open(filename_m, "x")
f.write(json.dumps(data_m, indent=4))
f.close()
# 在年度文件夹下, 创建月度文件夹
os.mkdir(path_k + "/" + id_y + "/" + id_m)

# 计算当月总天数
delta_d = day_m + relativedelta(months=+1) + relativedelta(days=-1)

# 日循环 j 次
for j in range(delta_d.day):
    day_d = day_m + relativedelta(days=+j)
    token_d = ''.join(random.sample(token, r_num))
    # 生成当日文档 ID, 时间默认为当日 01 点 01 分 03 秒
    id_d = day_d.strftime('%Y%m%d') + "010103-" + token_d
    data_d = {
        "ID": id_d,
        "Type": "NodeDocument",
        "Properties": {
            "id": id_d,
            # 标题格式 yyyy-mm-dd
            "title": day_d.strftime('%Y-%m-%d'),
            "updated": datetime.now().strftime('%Y%m%d%H%M%S')
        }
    }
}

```

```
    }  
  }  
  # 在年度文件夹 / 月度文件夹下, 创建当日文档  
  filename_d = path_k + "/" + id_y + "/" + id_m + "/" + id_d + ".sy"  
  f = open(filename_d, "x")  
  f.write(json.dumps(data_d, indent=4))  
  f.close()
```

```
tkinter.messagebox.showinfo("提示", "感谢等待, 创建已完成")
```

操作

环境

- win10
- 思源笔记1.8.4

步骤

1. 选择daily note笔记本——...更多——打开文件位置;
2. 关闭思源笔记;
3. 将上方代码保存为py文件, 修改第11和12行的起始结束日期, **不跨年**;
4. 文件运行完成后会出现成功的提示弹窗;
5. 将生成的result文件夹下所有文件及文件夹复制到daily note笔记本对应的本地文件夹下。 **不含result文件夹**;
6. 打开思源笔记, 可能需要等待几分钟, 再点击文档树——...更多——重建索引, 完成;

风险提示

本人几乎无Python基础, 代码为搜索+摸索产出, 此贴主要是分享探讨, 不排除使用时引发其他问题
建议新建工作目录测试成功后再用于正式目录。

有更好的优化方案也欢迎大家指教, 谢谢! ☺