



链滴

不限层级目录模板

作者: [crowds21](#)

原文链接: <https://ld246.com/article/1632302122435>

来源网站: [链滴](#)

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

看了这位老哥的帖子想到的: [分享一个思源笔记模板功能的高级用法——工作周日报模板 - 链滴 \(Id246.com\)](#)

把循环语句全部写在同一行,来控制缩进,即可输出不同层级.真是一叶障目.

之后等凑齐一波再整理到Organized中,比较懒.近期准备更新集市模板的朋友直接拿去用也无妨.

2021年9月22日 20点30分 更新: 解决只能在笔记本下第一层级的文档中使用的问题

```
.action{$docid:=.id}
.action{$docPath:= ""}
.action{$tab:= " "}
.action{$line:= "- "}

.action{$getdocInfo := (queryBlocks "SELECT * FROM blocks WHERE id='?' and type='d' " $docid)}
.action{range $v:= $getdocInfo}
    .action{$docPath =$v.Path}
.action{end}

.action{$temp:= (splitList "/" $docPath)}

.action{$block:= (queryBlocks "SELECT * FROM blocks WHERE id !=?' 'AND type= 'd' AND path like '%?%' Order BY path" $docid $docid)}
- ((.action{$docid}))
.action{range $v:= $block}
.action{$a:=splitList "/" $v.Path}
.action{range $p:= $temp}.action{$a = rest $a }.action{end}
.action{range $i:= $a}.action{$tab}.action{end}.action{$line}((.action{$v.ID}))
.action{end}
```

实际效果



0 请从这里开始

- [0 请从这里开始](#)
 - [编辑器](#)
 - [排版元素](#)
 - [内容块](#)
 - [0 什么是内容块](#)
 - [2 引用内容块](#)
 - [3 在内容块中遨游](#)
 - [1 内容块类型](#)
 - [5 嵌入内容块](#)
 - [4 文档块和标题块的转换](#)
 - [6 内容块属性](#)
 - [搜索进阶](#)
 - [0 数据库表](#)
 - [1 类型过滤](#)
 - [云端](#)
 - [2 数据备份和恢复](#)
 - [0 存储空间](#)
 - [1 数据同步](#)

(教程文档不就是为了测试的嘛)

其他

因为我没有系统性的学习并编写过golang程序.模板是完全依照sprig提供的API来实现的.

下面的内容只是我几次尝试后瞎总结的,可能存在错误.

对于[package builtin](#)标准库中的函数,按照以下方式即可直接调用.函数名在前,参数在后.

```
.action{len $temp}
```

输出结果为2. 发完贴子才试出来,不然能写的再简单,以后再改.

而对于其他库中的函数,如 [package time](#) 中的函数,即函数前标注了 `type xxx`. 貌似意思是函数作为结构体成员.

- func (t *Location) String() string
- type Time
 - func Date(year int, month Month, day, hour, min, sec, nsec int, loc *Location) Time
 - func Parse(layout, value string) (Time, error)
 - func ParseInLocation(layout, value string, loc *Location) (Time, error)
 - func Now() Time
 - func Unix(sec int64, nsec int64) Time
 - func (t Time) Location() *Location
 - func (t Time) Zone() (name string, offset int)

调用时如下

```
(toDate "2006-01-02" "2021-06-28").Sub now)
Time.Sub Value
```

toDate "2006-01-02" "2021-06-28" 返回的是Time类型的变量.