PDF-XChange
Click to buy NOW!
www.docu-track.com

第 1 章

PyDay 0 时刻准备着!发布

发布!为了全人类!

- 因为一个人如果力求完善自己，他就会看到，为此也必须同时完善他人。
  一个人如果不关心别人的完善，自己便不可能完善。

1.1　从需求导出用户群

小白已经实现的功能：

1. 扫描光盘内容并存储为硬盘上的文本文件
   - 存储成*.cdc 的文本文件
   - 可以快速指定保存目录
   - 可以快速指定保存的文件名
2. 根据储存到硬盘上的光盘信息计件进行搜索
   - 可以搜索指定目录中所有*.cdc 文件
   - 可以指定关键字进行搜索
     - 列出所有含有关键字的信息行

对于大胆并逐渐习惯命令行界面的小白,现在的 PyCDC 基本可用了,那么这样的小工具,有什么哪的人愿意使用?

推想一下：

- 有很多光盘的人
- 知道并会安装 Python 的人
- 不惧怕命令行的人
- 勇于尝试的人

这样的人也必定是愿意分享的人,小白当然在行者多日的熏陶下，刚刚完成 可用的 PyCDC 就想汇报给列表中的先辈们，获得夸奖， 更加希望有人通过使用反馈意见，甚至于代码，可以将这首个作品发展下去；

所以要发布!

1.2　发布的本质

发布好象不是通告一嗓子就好的事儿

现在的 CDC 仅仅是接受光盘信息和查询关键字，如果从面向数据的开发考虑，完全可以跃迁成 基于 FP～函式型程序。

一切都是函式间的嵌套调用，一但发生问题就会自然退出，无模式，无状态，只是单纯的数据响应，整个结构可以更加扁平不易出错，好维护……

1.2.1 给人用

小白尝试将 PyCDC 整理了个压缩包发送给朋友，结果所有的朋友都说不会使用...郁闷是小白的真实写照；

> 询问列表:"怎么发布软件哪?!"
>
> 回复:"文档!文档!文档!"

文档!

分分分!学生的命根! 文档,文档,文档!软件的颜面!

- 软件是给人使用的,但是你无法跑到每个愿意尝试使用软件的人的身份进行演示,如何使用你的软件吧?!
- 所以,文档!友好的文档!清晰的文档!有效的文档! 就是作者的代言人,形象大使,作为尝试者的向导,来令他人学习使用你的软件。
- 对于源代码发布的自由软件,代码也是文档的一部分,乱七八糟的代码,是没有人有兴趣试用的。

> **丰富的文档是可以安抚我们面对未知的恐惧的,推荐以下深入阅读资料,但是不推荐现在就全面阅读** Dive Into Python **深入** Python **中文版本** Abyte Of Python -- **简明** Python **教程** Python Reference Python **参考资料汇编**

1.2.2 对比及推导

那么哪种比较好?

- 小白当然原意令自个儿的小工具发扬光大下去
- 那么改进不可避免!
- 其实记录的摘要文本已经升级过一次级

```
o  ^                    ^  ^
o  |                    |  |  +- files 列表，此目录的文件名
o  |                    |  +- 各个数据段使用";" 分隔
o  |                    +- dirs 列表，子目录名，如果没有就为空
```

- +- 当前目录
- os.walk() 输出的自然结构格式
- 使用专门的 formatCDinfo() 函式格式化为
- ...
- -f /media/cdrom0/RyokoHirosue/Ryoko Hirosue-files.files title.gif
- == =======
- 其中 d 代表目录;f 代表文件
- 虽然是自个儿随意设计的格式, 但是小白可以体会出有以下好处:
  - 格式自然, 可以直接人工查阅
  - 搜索匹配后输出行就是自然行, 并包含必要信息

- drwxr-xr-x 3 zoomq zoomq 4096 2007-08-18 22:43 apidocs
- -rwxr-xr-x 1 zoomq zoomq 87 2007-07-31 14:58 AUTHORS
- drwxr-xr-x 3 zoomq zoomq 4096 2007-07-31 14:58 cdc
- -rwxr-xr-x 1 zoomq zoomq 3874 2007-08-18 22:42 cdctools.py
- -rw-r--r-- 1 zoomq zoomq 3758 2007-08-18 22:42 cdctools.pyc
- -rwxr-xr-x 1 zoomq zoomq 1152 2007-08-18 22:02 ChangeLog
- -rwxr-xr-x 1 zoomq zoomq 246 2007-08-18 22:01 PKG-INFO
- -rwxr-xr-x 1 zoomq zoomq 3273 2007-08-18 22:28 pycdc.py
- -rw-r--r-- 1 zoomq zoomq 4634 2007-08-18 22:28 pycdc.pyc
  - 是以空行来区分不同目录下的信息
  - 每节是固定顺序的信息:
  - ./apidocs: <-- 当前目录
  - -rw-r--r-- 1 zoomq zoomq 760 2007-08-18 22:52 api-objects.txt
  - ^ ^ ^ ^ ^ ^ ^
  - | | | | | | +- 文件/目录名
  - | | | | | +- 创建/修订时间
  - | | | | +- 体积
  - | | | +- 用户组
  - | | +- 用户
  - | +- 权限组

- +- 文件(-)或是目录(d)
  - 包含了丰富且规范的信息，但是作为 PyCDC 其中有关时间/权限的信息都是不必要的，而且不是跨平台兼容的，在 M$平台中，权限信息就根本用不上。

- CD Catalog Expert 的文本数据格式
- [Info]
- DriveType=5
- ImagePath=L:
- Volume=MCollec.39
- Serial=723840260
- FAT=CDFS
- DiskSize=676020224

epydoc

Easy Py Documentor——轻松的 Py 文档生成器!

- *http://wiki.woodpecker.org.cn/moin/CodeCommentingRule*
- *http://epydoc.sourceforge.net/api/epydoc-module.html* epydoc 自个儿的文档输出，是多么专业和规范哪

图 1-1

文件?是 os 的

- 文件大小信息... f 开头的模块都不象哪...
- 突然小白想起来,这文件和目录可都是文件系统,操作系统管理的哪
- 光盘扫描的 walk() 函式就是在 os 模块中的,文件这个信息也应该在的说?
- Files and Directories -- 文及目录 ;果然! 有个 os.stat() 系统状态函式!
- 照例 在 iPython 中快速尝试一下
- Bingo! 就它了!

1.3 完成重构

简单的说就是将原先 oa.walk() 生成的信息利用 基础配置处理机模块，组织成类 ini 的文本保存下来：

好了,组织好了,接下来就是表单组织,最后连接上即有功能就得!

HTMLTags

http://karrigell.sourceforge.net/en/htmltags.htm 有说,Karrigell 提供一个非常 OOP 的支持模块可以快速的函式样儿的组织生成 HTML 代码:

```
print HTML( HEAD(TITLE('Test document')) +
       BODY(H1('This is a test document')+
           TEXT('First line')+BR()+
           TEXT('Second line')))
```

将生成:

```
<HTML>
<HEAD>
<TITLE>Test document</TITLE>
</HEAD>
<BODY>
<H1>This is a test document</H1>
First line
<BR>
Second line
</BODY>
</HTML>
```

但是,可以进行交互提交的表单,好象,貌似,无法函式化的生成哪...

有点郁,小白不想自个儿折腾,根据 -1 day 的经验:

```
def cdWalker(cdrom,cdcfile):
    '''光盘扫描主函式
    @param cdrom: 光盘访问路径
    @param cdcfile: 输出的光盘信息记录文件(包含路径,绝对,相对都可以)
    @return: 无,直接输出成*.cdc 文件
    @attention: 从 v0.7 开始不使用此扫描函式,使用 iniCDinfo()
    '''
    export = ""
    for root, dirs, files in os.walk(cdrom):
        print formatCDinfo(root,dirs,files)
        export+= formatCDinfo(root,dirs,files)
    open(cdcfile, 'w').write(export)

def formatCDinfo(root,dirs,files):
    '''光盘信息记录格式化函式
    @note: 直接利用 os.walk() 函式的输出信息进行重组
    @param root: 当前根
    @param dirs: 当前根中的所有目录
    @param files: 当前根中的所有文件
    @return: 字串,组织好的当前目录信息
    '''
    export = "\n"+root+"\n"
    for d in dirs:
        export+= "-d "+root+_smartcode(d)+"\n"
    for f in files:
        export+= "-f %s %s \n" % (root,_smartcode(f))
    export+= "="*70
    return export
{/old}[e]
```

1.4. 重构

先来个 20 行:

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

```
 2 def _htmhead(title):
 3     '''默认页面头声明
 4     @note: 为了复用，特别的组织成独立函式,根据 Karrigell 非页面访问约定，函
式名称前加"_"
 5     @param title: 页面标题信息
 6     @return: 标准的 HTML 代码
 7     '''
 8     htm = """<html><HEAD>
 9 <meta http-equiv="Content-Type" content="text/html;charset=utf-8"/>
10 <title>%s</title></HEAD>
11 <body>"""%title
12     return htm
13 ## 默认页面尾声明
14 htmfoot="""
15 <h5>design by:<a href="mailto:Zoom.Quiet@gmail.com">Zoom.Quiet</a>
16     powered by : <a href="http://python.org">Python</a> +
17  <a href="http://karrigell.sourceforge.net"> KARRIGELL 2.3.5</a>
18 </h5>
19 </body></html>"""
20
21 def index(**args):
22     print _htmhead("PyCDC WEB")
23     print htmfoot
```

1.5　引发的改进

-5 PyDay 初体验和原始需求

Use it! do not learnning -- 用之,不学!

剧本背景

所谓实例故事,就是设计一个具体情景,让代表读者的菜鸟,跟着代表作者的老鸟,完成一件事儿,在操作过程中引导读者学习 Python 编程语言;

人物

> 小白：读者一方,没有或是仅有一点编程体验的好奇宝宝,想使用 Python 解决一个实际的问题。
>
> 行者：哈！啄木鸟社区的一个或是一群热心的先行学习过 Python 的好人,说话可能有些颠三倒四,但是绝对是好心人哪。

列表：

指**邮件列表**——一种仅仅通过邮件进行群体异步交流的服务形式,是比 BBS 更加古老和有效的沟通方式（邮件列表的规范和礼节 CPyUG 社区列表资源）。

触发事件

怎么回事儿呢? 原来是小白买了台刻录机,这一下, eMule 的下载更加是没日没夜了,但是才一个月刻录出来的光盘就有上百张了,结果想找回一个专辑的 MP3,简直不可...得粉碎。

想要一种工具：

可以不用插入光盘就可以搜索所有光盘的内容。

行者：Python!

小白：OK!你们都说 Python 好用,那么来尝试一下吧！我是菜鸟我怕谁?!

运行环境：

推荐 ActivePython——一个商业产品,但是有自由使用版权的,一个完善的 Python 开始应用环境,关键是文档齐备; GNU/Linux 环境中,当然是原生的 python.org

增加 python 环境的设置说明前后文链接

```
Python 2.4.3 (#69, Mar 29 2006, 17:35:34) [MSC v.1310 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print "Hello World!"
Hello World!
>>>
```

图 1-2

再 Show 一个类似的,但是推荐的体验环境 [1]

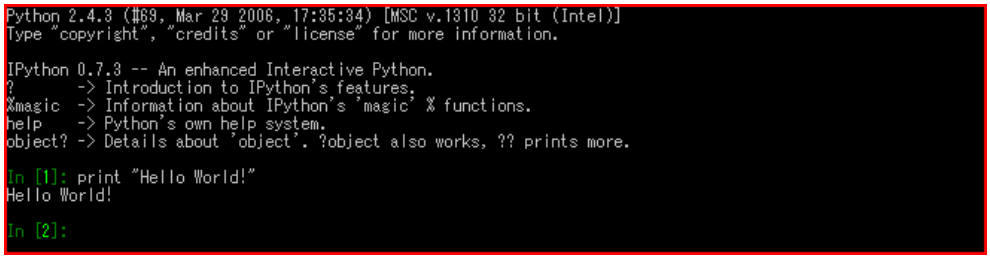

图 1-3

行者：是也乎,就是这么简单,告诉 Python 打印"Hello World!" 就好。所以说,对于 Python, 勿学,即用就好!

-5 PyDay 小结

今天小白决定使用 Python 来解决光盘内容管理,这一实际问题; 安装了 python 环境,运行了 "Hello World!" 实例.

> **列表指邮件列表——一种仅仅通过邮件进行群体异步交流的服务形式,是比 BBS 更加古老和有效的沟通方式。**

OK！非常轻松的开始，但是，你知道吗？你同时获得了免费的绝对强大的科学计算器！

1.6 练习

1、计算今年是闰年嘛?

2、计算...

3、::-- ZoomQuiet [2007-04-13 16:14:09]