Windard's Python Book

Windard

Python-Book

About

Python

python

Python ,,Python

> Python | PythonImagingLibrary

python python cookbook

> Python | Python Cookbook 3rd Edition Documentation

Python

>

😁😄 😆

:

- pdf
- mobi
- epub

python python

os

1. -- `os.getcwd`
2. -- `os.chdir`
3. -- `os.mkdir`
4. -- `os.makedirs`
5. ,-- `os.rmdir`
6. ,-- `os.removedirs`
7. -- `os.listdir`

8. -- `os.remove`
9. -- `os.unlink`
10. -- `os.rename(old new)`
11. -- `os.path.getsize`
12. -- `os.stat`
13. -- `os.chmod`
14. -- `os.path.join(path,file)`
15. -- `os.path.split`
16. -- `os.path.splitext`
17. -- `os.path.dirname`
18. -- `os.path.basename`
19. -- `os.path.isdir`
20. -- `os.path.isfile`
21. ——— `os.path.exists`
22. -- `os.path.isabs`
23. -- `os.path.abspath` `os.path.realpath`

    `os.path.abspath(os.path.dirname(__file__))`

    `os.path.dirname(os.path.abspath(__file__))` `os.path.abspath(os.curdir)`
24. -- `os.path.relpath(path, start)`
25. -- `os.getenv` `os.putenv`
26. -- `os.path.expanduser('~')`
27. -- `os.name`

    Windows `nt` Linux/Unix/Mac `posix`
28. -- `os.sep`

    Windows `\` Linux `/` Mac OS `/`
29. -- `os.linesep`

    Windows `\r\n` ,Linux `\n` , Mac OS `\r` Mac `\n`
30. -- `os.pathsep`

    Windows `;` Linux `:` Mac OS `:`
31. shell-- `os.system`

> print

32. shell-- `os.popen`
> file openreadcommands

33. shell -- `os.execlp`
> `os.execlp('ls','')`

34. -- `os._exit`
> `os._exit(0)` `sys.exit`

35. -- `os.walk`

36. -- `os.environ`
> HTTP

37. -- `os.urandom`
> `os.urandom(num)` num ASCII

38. -- `os.access(filename, privilege)`
39. -- `os.fork`
> linux unix windows

40. pid -- `os.getpid`
41. pid -- `os.getppid`
42. -- `os.kill`
> `os.kill(os.getpid(), 9)`

```
# coding=utf-8

import os

currentpath = os.getcwd()
print currentpath
changedpath = 'C:\\Users\\dell\\Desktop'
os.chdir(changedpath)
currentpath = os.getcwd()
print currentpath
os.mkdir('hello')
changedpath = changedpath + '\\hello'
print changedpath
os.chdir(changedpath)
currentpath = os.getcwd()
print currentpath
os.makedirs('hello\\hello')
changedpath = changedpath + '\\hello\\hello'
print changedpath
os.chdir(changedpath)
currentpath = os.getcwd()
print currentpath
os.chdir('../')
currentpath = os.getcwd()
```

```
print currentpath
currentlist = os.listdir(currentpath)
print currentlist
os.rmdir('hello')
currentlist = os.listdir(currentpath)
print currentlist
os.chdir('../../')
currentpath = os.getcwd()
currentlist = os.listdir(currentpath)
print currentlist
os.removedirs('hello\\hello')
currentlist = os.listdir(currentpath)
print currentlist
FILE1 = open('test1.txt','w')
FILE1.close()
FILE2 = open('test2.txt','w')
FILE2.close()
currentlist = os.listdir(currentpath)
print currentlist
os.remove('test1.txt')
currentlist = os.listdir(currentpath)
print currentlist
os.rename('test2.txt','newtest.txt')
currentlist = os.listdir(currentpath)
print currentlist
FILE = open('newtest.txt','w')
FILE.write('THis is for test')
FILE.close()
FILESIZE = os.path.getsize('newtest.txt')
print FILESIZE
FILESTAT = os.stat('newtest.txt')
print FILESTAT
currentpath = currentpath + "\\newtest.txt"
print currentpath
(splitpath,splitfile) = os.path.split(currentpath)
print splitpath
print splitfile
(splitpath,splitfile) = os.path.splitext(currentpath)
print splitpath
print splitfile
splitpath = os.path.dirname(currentpath)
splitfile = os.path.basename(currentpath)
print splitpath
print splitfile
isdir = os.path.isfile(currentpath)
isfile = os.path.isdir(currentpath)
print isdir
print isfile
os.remove('newtest.txt')
currentpath = os.path.dirname(currentpath)
isdir = os.path.isfile(currentpath)
isfile = os.path.isdir(currentpath)
print isdir
```

```
print isfile
isexist = os.path.exists(currentpath)
print isexist
isabs = os.path.isabs(currentpath)
print isabs
osname = os.name
print osname
linesep = os.linesep
print linesep
os.system('dir')
```

`os_demo.py`

```
C:\Users\dell\.ssh\Python_Lib>python os.demo.py
C:\Users\dell\.ssh\Python_Lib
C:\Users\dell\Desktop
C:\Users\dell\Desktop\hello
C:\Users\dell\Desktop\hello
C:\Users\dell\Desktop\hello\hello\hello
C:\Users\dell\Desktop\hello\hello\hello
C:\Users\dell\Desktop\hello\hello
['hello']
[]
['desktop.ini', 'hello', 'SOMETHING']
['desktop.ini', 'SOMETHING']
['desktop.ini', 'SOMETHING', 'test1.txt', 'test2.txt']
['desktop.ini', 'SOMETHING', 'test2.txt']
['desktop.ini', 'newtest.txt', 'SOMETHING']
16
nt.stat_result(st_mode=33206, st_ino=0L, st_dev=0L, st_nlink=0, st_uid=0, st_gid=0, st_size=16L, st_atime=1444984570L, st_mtime=1444984
70L, st_ctime=1444984570L)
C:\Users\dell\Desktop\newtest.txt
C:\Users\dell\Desktop
newtest.txt
C:\Users\dell\Desktop\newtest
.txt
C:\Users\dell\Desktop
newtest.txt
True
False
False
True
True
True
nt


 驱动器 C 中的卷没有标签。
 卷的序列号是 32D7-EC2A

 C:\Users\dell\Desktop 的目录

2015/10/16  16:36    <DIR>          .
2015/10/16  16:36    <DIR>          ..
2015/10/16  16:05    <DIR>          SOMETHING
               0 个文件              0 字节
```

```
# coding=utf-8

import os

print "File Name: ",__file__
print "Exist File ? ",os.access(__file__,os.F_OK)
print "Read File ? ",os.access(__file__,os.R_OK)
print "Write File ? ",os.access(__file__,os.W_OK)
print "Execute File ? ",os.access(__file__,os.X_OK)
```

```
File Name:  /home/windard/github/Python_Lib/code/os_access.py
Exist File ?  True
Read File ?  True
Write File ?  True
Execute File ?  False
```

ubuntu

```
# coding=utf-8

import os

# pid = os.fork()

# if pid:
#     print "Child Pid : %s, Current Pid %s"%(pid,os.getpid())
# else:
#     print "I am the child,Current Pid %s"%(os.getpid())

def create_child():
    pid0=os.getpid()
    print ''   ,pid0

    try:
        pid1=os.fork()
    except OSError:
        print u'fork'
        exit()

    if pid1 < 0:
        print u''
    elif pid1==0:
        print ' pid:%d pid: %d: %d '                                    %(pid1,os.getpid(),o
s.getppid())
    else:
        print ' pid:%d :%d ,: %d '                                  %(pid1,os.getpid(),os.
getppid())

    print ','

create_child()
```

```
 29511
 pid:29512 :29511 ,: 9372
,
 pid:0 pid: 29512: 29511
,
```

shell

```
import os
shell = "dir"
print os.system(shell)
```

os_shell.py

```
 Directory of C:\Users\dell\.ssh\Python_Lib\code

2015/11/17  00:22    <DIR>          .
2015/11/17  00:22    <DIR>          ..
2015/11/15  22:58               588 163mail_smtp_demo.py
2015/11/11  19:26               748 163mail_smtp_html.py
2015/11/11  20:19             1,097 163mail_smtp_multipart.py
2015/10/16  17:26               413 argparse_add_argument.py
2015/10/17  14:01               157 argparse_chose.py
2015/10/16  20:43               628 argparse_count.py
2015/10/16  17:11                74 argparse_demo.py
2015/10/24  12:27               173 base64_demo.py
2015/10/24  12:24               141 base64_encode.py
2015/10/18  19:06               279 create.php
2015/11/11  20:57               644 envelopes_demo.py
2015/10/24  12:00               118 hash_sha1.py
2015/11/15  11:22               385 imageDraw_demo.py
2015/11/15  11:29               552 imageEnhance_demo.py
2015/11/15  15:39               164 imageFilter_demo.py
2015/11/15  12:19               215 imageFont_demo.py
2015/11/14  14:41               548 image_change.py
2015/11/15  14:52               953 image_code.py
2015/11/14  15:10               415 image_create.py
2015/11/13  21:29                64 Image_demo.py
2015/11/15  11:07               159 image_histogram.py
```

`os.walk()`

```python
import os

dirlist = r"C:\Users\dell\Desktop\2048"
filenum = 0
dirnum  = 0
for i,j,k in os.walk(dirlist):
    print i
for i,j,k in os.walk(dirlist):
    for item in k:
        print item
    filenum = filenum + 1
    for index in range(len(k)):
        dirnum = dirnum + 1

print filenum
print dirnum
```

os_walk.py

```
C:\Users\dell\.ssh\Python_Lib\code>python os_walk.py
C:\Users\dell\Desktop\2048
C:\Users\dell\Desktop\2048\css
C:\Users\dell\Desktop\2048\js
index.html
README.md
2048.css
jquery.min.js
main2048.js
showanimation2048.js
support2048.js
3
7
```

```
# coding=utf-8
import os

def showall(path,leavel=0,filenum=0,show=True):
    newnum = filenum
    currentpath = path;
    dirandfile = os.listdir(path)
    for item in dirandfile:
        newpath = os.path.join(currentpath,item)
        if os.path.isdir(newpath):
            num = showall(newpath,leavel+1,newnum,show)
            newnum = num
        else:
            newnum = newnum + 1
            tab_stop = ""
            if show:
                for tab in range(leavel):
                    tab_stop = tab_stop + " "
            print tab_stop + newpath

    return newnum

if __name__ == '__main__':
    num = showall('./',show=False)
    print "File Number : " + str(num)
```

```
# coding=utf-8

import os

environment = os.environ

for i,j in environment.items():
    print "%s : %s "%(i,j)
```

```
TMP : C:\Users\dell\AppData\Local\Temp
COMPUTERNAME : YANGWENQIANG
USERDOMAIN : YANGWENQIANG
GOROOT : C:\Go\
PSMODULEPATH : C:\Users\dell\Documents\WindowsPowerShell\Modules;C:\Program Files\Windows
PowerShell\Modules;C:\WINDOWS\system32\WindowsPowerShell\v1.0\Modules\
COMMONPROGRAMFILES : C:\Program Files (x86)\Common Files
PROCESSOR_IDENTIFIER : Intel64 Family 6 Model 69 Stepping 1, GenuineIntel
PROGRAMFILES : C:\Program Files (x86)
PROCESSOR_REVISION : 4501
PATH : C:\Python27\Lib\site-packages\PyQt4;C:\ProgramData\Oracle\Java\javapath;C:\Program
 Files\Dell\DW WLAN Card;C:\Perl64\site\bin;C:\Perl64\bin;C:\Program Files (x86)\Common F
```

```
iles\Intel\Shared Files\cpp\bin\Intel64;C:\Windows\system32;C:\Windows;C:\Windows\System3
2\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Program Files\WIDCOMM\Bluetooth Sof
tware\;C:\Program Files\WIDCOMM\Bluetooth Software\syswow64;C:\Program Files (x86)\NVIDIA
 Corporation\PhysX\Common;D:\Program Files (x86)\QuickTime\QTSystem\;D:\Program Files\Tor
toiseSVN\bin;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\WINDOWS\System32\
WindowsPowerShell\v1.0\;C:\Python34;C:\Python27;C:\Perl64;C:\Program Files\Java\jdk1.8.0_
60\bin;D:\Program Files (x86)\Sublime text2\Sublime2\Sublime2\Sublime;C:\Users\dell\AppDa
ta\Local\Google\Chrome\Application;C:\mingw;C:\Python27\Scripts;C:\Ruby22-x64;C:\PHP;C:\c
url-7.33.0-win64-ssl-sspi;C:\curl-7.33.0-win64-ssl-sspi;C:\Program Files (x86)\MySQL\MySQ
L Server 5.7\bin;D:\Program Files (x86)\MySQL\MySQL Server 5.7\bin;C:\sqlite3;C:\Apache24
\bin;C:\gunwin32\GetGnuWin32\bin;D:\Program Files (x86)\Tesseract-OCR;C:\Users\dell\.ssh\
Python_Lib\project;C:\MinGW\bin;D:\Program Files\cmder\bin;D:\Program Files\cmder;C:\User
s\dell\.ssh\Python_Lib\project;C:\Go\bin;C:\Program Files (x86)\Google\Chrome\Application
;D:\Program Files\nodejs\;D:\Program Files\VMare\OVFTool;D:\Program Files (x86)\Git\cmd;D
:\Program Files (x86)\Git\bin;C:\Ruby22-x64\bin;D:\Program Files (x86)\Nmap;C:\Users\dell
\AppData\Roaming\npm
SYSTEMROOT : C:\WINDOWS
PROGRAMFILES(X86) : C:\Program Files (x86)
C_EM64T_REDIST11 : C:\Program Files (x86)\Common Files\Intel\Shared Files\cpp\
ASL.LOG : Destination=file
TESSDATA_PREFIX : D:\Program Files (x86)\Tesseract-OCR\
TEMP : C:\Users\dell\AppData\Local\Temp
WINDIR : C:\WINDOWS
COMMONPROGRAMFILES(X86) : C:\Program Files (x86)\Common Files
PROCESSOR_ARCHITECTURE : x86
ALLUSERSPROFILE : C:\ProgramData
LOCALAPPDATA : C:\Users\dell\AppData\Local
FPS_BROWSER_USER_PROFILE_STRING : Default
HOMEPATH : \Users\dell
USERDOMAIN_ROAMINGPROFILE : YANGWENQIANG
JAVA_HOME : C:\Program Files\Java\jdk1.8.0_60
PROGRAMW6432 : C:\Program Files
USERNAME : dell
LOGONSERVER : \\MicrosoftAccount
COMSPEC : C:\WINDOWS\system32\cmd.exe
PROGRAMDATA : C:\ProgramData
CLASSPATH : .;D:\Program Files (x86)\QuickTime\QTSystem\QTJava.zip;C:\Program Files\Java\
jdk1.8.0_60\bin;C:\Program Files\Java\jdk1.8.0_60\lib\tools.jar;C:\Users\dell\Desktop\pyt
hon\java
FPS_BROWSER_APP_PROFILE_STRING : Internet Explorer
AWE_DIR : D:\Program Files (x86)\Khrona LLC\Awesomium SDK\1.6.6\
SESSIONNAME : Console
PATHEXT : .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC;.PY;.RB;.RBW;.CPL
CONFIGSETROOT : C:\WINDOWS\ConfigSetRoot
FP_NO_HOST_CHECK : NO
QTJAVA : D:\Program Files (x86)\QuickTime\QTSystem\QTJava.zip
OPENSSL_CONF : C:\OpenSSL-Win32\bin\openssl.cfg
MOZ_PLUGIN_PATH : C:\Program Files (x86)\Foxit Software\Foxit Reader Plus\plugins\
HOMEDRIVE : C:
SYSTEMDRIVE : C:
NUMBER_OF_PROCESSORS : 4
APPDATA : C:\Users\dell\AppData\Roaming
PROCESSOR_LEVEL : 6
```

```
PROCESSOR_ARCHITEW6432 : AMD64
COMMONPROGRAMW6432 : C:\Program Files\Common Files
OS : Windows_NT
PUBLIC : C:\Users\Public
USERPROFILE : C:\Users\dell
```

cgi header

```
# coding=utf-8

import os

environment = os.environ

print "Content-type:text/html\r\n\r\n"

for i,j in environment.items():
    print "%s : %s <br />"%(i,j)
```

```
ALLUSERSPROFILE:C:\ProgramData
APPDATA:C:\Users\dell\AppData\Roaming
ASL.LOG:Destination=file
AWE_DIR:D:\Program Files (x86)\Khrona LLC\Awesomium SDK\1.6.6\
C_EM64T_REDIST11:C:\Program Files (x86)\Common Files\Intel\Shared Files\cpp\
CLASSPATH:.;D:\Program Files (x86)\QuickTime\QTSystem\QTJava.zip;%JAVA_HOME%\bin;%JAVA_HO
ME%\lib\tools.jar;C:\Users\dell\Desktop\python\java
COMMONPROGRAMFILES:C:\Program Files (x86)\Common Files
COMMONPROGRAMFILES(X86):C:\Program Files (x86)\Common Files
COMMONPROGRAMW6432:C:\Program Files\Common Files
COMPUTERNAME:YANGWENQIANG
COMSPEC:C:\WINDOWS\system32\cmd.exe
CONFIGSETROOT:C:\WINDOWS\ConfigSetRoot
Content-type:text/html
CONTENT_LENGTH:
CONTENT_TYPE:text/plain
Date:Thu, 18 Aug 2016 14:05:03 GMT
FP_NO_HOST_CHECK:NO
FPS_BROWSER_APP_PROFILE_STRING:Internet Explorer
FPS_BROWSER_USER_PROFILE_STRING:Default
GATEWAY_INTERFACE:CGI/1.1
GOROOT:C:\Go\
HOMEDRIVE:C:
HOMEPATH:\Users\dell
HTTP_ACCEPT:text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
HTTP_COOKIE:_ga=GA1.1.2013049555.1445440760
HTTP_REFERER:
HTTP_USER_AGENT:Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Geck
o) Chrome/52.0.2743.116 Safari/537.36
JAVA_HOME:C:\Program Files\Java\jdk1.8.0_60
LOCALAPPDATA:C:\Users\dell\AppData\Local
LOGONSERVER:\\MicrosoftAccount
```

```
MOZ_PLUGIN_PATH:C:\Program Files (x86)\Foxit Software\Foxit Reader Plus\plugins\
NUMBER_OF_PROCESSORS:4
OPENSSL_CONF:C:\OpenSSL-Win32\bin\openssl.cfg
OS:Windows_NT
PATH:C:\Python27\Lib\site-packages\PyQt4;C:\ProgramData\Oracle\Java\javapath;C:\Program F
iles\Dell\DW WLAN Card;C:\Perl64\site\bin;C:\Perl64\bin;C:\Program Files (x86)\Common Fil
es\Intel\Shared Files\cpp\bin\Intel64;C:\Windows\system32;C:\Windows;C:\Windows\System32\
Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Program Files\WIDCOMM\Bluetooth Softw
are\;C:\Program Files\WIDCOMM\Bluetooth Software\syswow64;C:\Program Files (x86)\NVIDIA C
orporation\PhysX\Common;D:\Program Files (x86)\QuickTime\QTSystem\;D:\Program Files\Torto
iseSVN\bin;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\WINDOWS\System32\Wi
ndowsPowerShell\v1.0\;C:\Python34;C:\Python27;C:\Perl64;C:\Program Files\Java\jdk1.8.0_60
\bin;D:\Program Files (x86)\Sublime text2\Sublime2\Sublime2\Sublime;C:\Users\dell\AppData
\Local\Google\Chrome\Application;C:\mingw;C:\Python27\Scripts;C:\Ruby22-x64;C:\PHP;C:\cur
l-7.33.0-win64-ssl-sspi;C:\curl-7.33.0-win64-ssl-sspi;C:\Program Files (x86)\MySQL\MySQL
Server 5.7\bin;D:\Program Files (x86)\MySQL\MySQL Server 5.7\bin;C:\sqlite3;C:\Apache24\b
in;C:\gunwin32\GetGnuWin32\bin;D:\Program Files (x86)\Tesseract-OCR;C:\Users\dell\.ssh\Py
thon_Lib\project;C:\MinGW\bin;D:\Program Files\cmder\bin;D:\Program Files\cmder;C:\Users\
dell\.ssh\Python_Lib\project;C:\Go\bin;C:\Program Files (x86)\Google\Chrome\Application;D
:\Program Files\nodejs\;D:\Program Files\VMare\OVFTool;D:\Program Files (x86)\Git\cmd;D:\
Program Files (x86)\Git\bin;C:\Ruby22-x64\bin;D:\Program Files (x86)\Nmap;C:\Users\dell\A
ppData\Roaming\npm
PATH_INFO:
PATH_TRANSLATED:C:\Users\dell\Desktop\python
PATHEXT:.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC;.PY;.RB;.RBW;.CPL
PROCESSOR_ARCHITECTURE:x86
PROCESSOR_ARCHITEW6432:AMD64
PROCESSOR_IDENTIFIER:Intel64 Family 6 Model 69 Stepping 1, GenuineIntel
PROCESSOR_LEVEL:6
PROCESSOR_REVISION:4501
PROGRAMDATA:C:\ProgramData
PROGRAMFILES:C:\Program Files (x86)
PROGRAMFILES(X86):C:\Program Files (x86)
PROGRAMW6432:C:\Program Files
PSMODULEPATH:C:\Users\dell\Documents\WindowsPowerShell\Modules;C:\Program Files\WindowsPo
werShell\Modules;C:\WINDOWS\system32\WindowsPowerShell\v1.0\Modules\
PUBLIC:C:\Users\Public
QTJAVA:D:\Program Files (x86)\QuickTime\QTSystem\QTJava.zip
QUERY_STRING:
REMOTE_ADDR:127.0.0.1
REMOTE_HOST:YangWenqiang
REQUEST_METHOD:GET
SCRIPT_NAME:/cgi-bin/environment.py
Server:SimpleHTTP/0.6 Python/2.7.10
SERVER_NAME:YangWenqiang
SERVER_PORT:8001
SERVER_PROTOCOL:HTTP/1.0
SERVER_SOFTWARE:SimpleHTTP/0.6 Python/2.7.10
SESSIONNAME:Console
SYSTEMDRIVE:C:
SYSTEMROOT:C:\WINDOWS
TEMP:C:\Users\dell\AppData\Local\Temp
TESSDATA_PREFIX:D:\Program Files (x86)\Tesseract-OCR\
```

```
TMP:C:\Users\dell\AppData\Local\Temp
USERDOMAIN:YANGWENQIANG
USERDOMAIN_ROAMINGPROFILE:YANGWENQIANG
USERNAME:dell
USERPROFILE:C:\Users\dell
WINDIR:C:\WINDOWS
```

sys

-
-

argparse

1. `sys.argv`

```
# coding=utf-8
import sys

#0
print sys.argv[1]

#
print sys.argv[2] + sys.argv[3]

#
print int(sys.argv[2])+int(sys.argv[3])

#
num = 0
for i in sys.argv[2:]:
    num = num+int(i)
print num
```

`sys_argv.py`

```
C:\Users\dell\.ssh\Python_Lib\code>python sys_argv.py first 3 5 9 7
first
35
8
24

C:\Users\dell\.ssh\Python_Lib\code>
```

1. `sys.platform()` WindowsLinux
2. `sys.exit(n)` n00

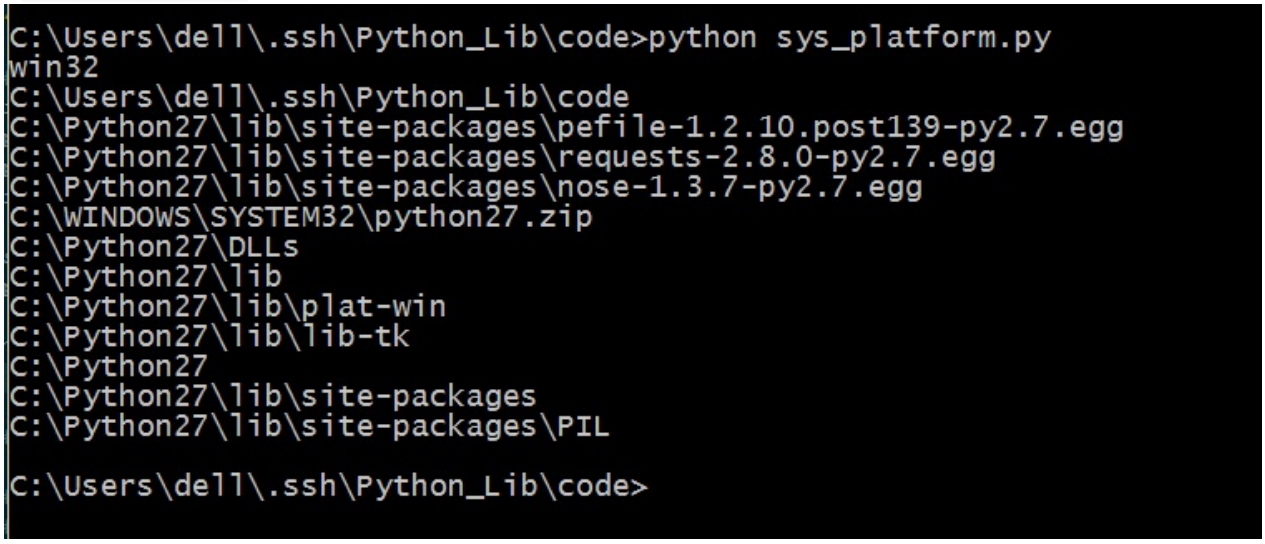
   > `sys.exit()` `os._exit()` `exit()/quit()` Python
   >
   > `sys.exit()` Python
   >
   > `os._exit()`
   >
   > `exit()/quit()` shell

3. `sys.path` `python -m site`

```
import sys
print sys.platform
path = sys.path
```

```
for i in path:
    print i
sys.exit(0)
print "This won't run"
```

`sys_platform.py`



1. `sys.modules()` python
2. `sys.version` Python
3. `sys.hexversion` Python
4. `sys.maxint` int
5. `sys.maxunicode` Unicode
6. `sys.copyright` Python
7. `sys.version_info` Python
8. `sys.api_version` CAPI
9. `sys.exec_prefix` Python
10. `sys.byteorder` bigbig-endianlittlelittle-endian
11. `sys.getdefaultencoding()` Python z.xASCIIPython 3.xUnicode
12. `sys.setdefaultencoding()`
13. `sys.executable` Python
14. `sys.getwindowsversion()` Windows

```
import sys
print sys.version
print sys.version_info
print sys.hexversion
print sys.api_version
print sys.exec_prefix
print sys.executable
print sys.maxint
print sys.maxunicode
print sys.byteorder
print sys.getdefaultencoding()
```

```
print sys.getwindowsversion()
print sys.getfilesystemencoding()
modules =  sys.modules
for i in modules.keys():
    print i
print sys.copyright
```

`sys_modules.py`

```
C:\Users\dell\.ssh\Python_Lib\code>python sys_modues.py
2.7.10 (default, May 23 2015, 09:40:32) [MSC v.1500 32 bit (Intel)]
sys.version_info(major=2, minor=7, micro=10, releaselevel='final', serial=0)
34015984
1013
C:\Python27
C:\Python27\python.exe
2147483647
65535
little
ascii
sys.getwindowsversion(major=6, minor=2, build=9200, platform=2, service_pack='')
mbcs
copy_reg
sre_compile
locale
_sre
functools
encodings
site
__builtin__
```

1. `sys.platform`
2. `sys.stdout`
3. `sys.stdin`
4. `sys.stderr`

```
import sys
data = sys.stdin
print data
sys.stdout.write("hello,world")
```

`sys_std.py`

```
C:\Users\dell\.ssh\Python_Lib\code>python sys_std.py heh
<open file '<stdin>', mode 'r' at 0x0234D020>
hello,world
C:\Users\dell\.ssh\Python_Lib\code>
```

1. `sys.getrecursionlimit()` python
2. `sys.setrecursionlimit(15000)` Mac 1000

re

-
-
-
-
-
-
-
  - nginx
  -
  -
-

python

`\`	
`\d`	[0-9]
`\D`	
`\s`	`[<>\t\n\r\f\v ]`
`\S`	
`\w`	[a-zA-Z0-9]
`\W`	`[^\w]`
`\A`	
`\Z`	
`\b`	
`\B`	
`\\`	`\`
`.`	`\n`
`*`	
`+`	
`{m}`	m
`{m,n}`	mn

`^`	
`$`	
`[……]`	
`(……)`	

pythonmatchsearch

matchsearchMatch,

- `match(pattern, string, flags=0)`
- `search(pattern, string, flags=0)`

python `\` `r'XXX'`

```
>>> a = re.match(r"he","hello , world")
>>> a
<_sre.SRE_Match object at 0xb70afdb0>
>>> a.group()
'he'
>>> b = re.search(r"wo","hello , world")
>>> b
<_sre.SRE_Match object at 0xb70afdb0>
>>> b.group()
'wo'
>>> c = re.match(r"wo","hello , world")
>>> c
>>> type(c)
<type 'NoneType'>
>>> re.match(r'(hello) , (world)', a).group(0)
'hello , world'
```

findall,

- `findall(pattern, string, flags=0)`

```
>>> d = re.findall(r"wo","hello , world")
>>> d
['wo']
>>> e = re.findall(r"\w","hello , world")
>>> e
['h', 'e', 'l', 'l', 'o', 'w', 'o', 'r', 'l', 'd']
>>> re.findall(r"\d+", "2333abc3uio890da123")
['2333', '3', '890', '123']
```

subsplit

- `sub(pattern, repl, string, count=0, flags=0)`

- `split(pattern, string, maxsplit=0, flags=0)`

> repl

```
>>> a = "hello , world"
>>> b = re.sub(r"o","0","hello , world")
>>> b
'hell0 , w0rld'
>>> a
'hello , world'
>>> c = re.split(r"\s","hello , world")
>>> c
['hello', ',', 'world']
>>> a
'hello , world'
>>> re.sub(r'(hello) , (world)', r'\2 , \1', a)
'world , hello'
>>> '%s , %s' % re.match(r'(hello) , (world)', a).groups()
'hello , world'
>>> '%s , %s' % (re.match(r'(hello) , (world)', a).group(2), re.match(r'(hello) , (world)
', a).group(1))
'world , hello'
>>> re.search(r"(.+?)\1+", 'dxabcabcyyyydxycxcxz').group()
'abcabc'
```

subsubnsub

```
>>> b = re.subn(r"o","0","hello , world")
>>> b
('hell0 , w0rld', 2)
>>> a
'hello , world'
```

, `\1`

>

```
In [4]: re.search(r"(.+?)\1+", 'dxabcabcyyyydxycxcxz').group()
Out[4]: 'abcabc'

In [5]: re.search(r"(.+?)\1+", 'dxabcabcyyyydxycxcxz').groups()
Out[5]: ('abc',)
```

sublime `$1` `$2`

python `\1` `\2` `\0`

- match search
- group groups , groupdict

abb

```
>>> re.compile(r'(a)(b)\2*').search('abbbb').groups()
('a', 'b')
>>> re.compile(r'(a)(b)\2*').search('abbbb').group()
'abbbb'
```

PatternPatternMatchMatch

```
# coding=utf-8

import re

pattern = re.compile(r"he")

match = pattern.match("hello , world")

if match:
    print match.group()
```

re_demo.py

```
windard@windard:~/github/Python_Lib/code$ python re_demo.py
he
```

compile `compile(pattern, flags=0)` flagsflags `|`

1. `re.I(re.IGNORECASE)` :
2. `re.M(re.MULTILINE)` :'^''$'
3. `re.S(re.DOTALL)` :'.'
4. `re.L(re.LOCALE)` : `\w \W \b \B \s \S`
5. `re.U(re.UNICODE)` : `\w \W \b \B \s \S \d \D` unicode
6. `re.X(re.VERBOSE)` :

Match

1. `string` :
2. `re` : Pattern
3. `pos` :
4. `endpos` :
5. `lastindex` : None
6. `lastgroup` : None

1. `group([group1, …])` : group10group(0),None
2. `groups([default])` : group(1,2,…last)defaultNone
3. `groupdict([default])` : default
4. `start([group])` : stringgroup0
5. `end([group])` : string+1group0
6. `span([group])` : (start(group), end(group))
7. `expand(template)` : templatetemplate `\id` `\g<id>` `\g<name>` 0 `\id` `\g<id>` `\10` 10 `\1` `'0'` `\g<1>0`

```
# coding=utf-8

import re

pattern = re.compile(r"(\w{1,6})(\s)(\,)(\s)(\w*)$")
m = re.match(pattern,"hello , world")

print "m.string:", m.string
print "m.re:", m.re
print "m.re.pattern:", m.re.pattern
print "m.pos:", m.pos
print "m.endpos:", m.endpos
print "m.lastindex:", m.lastindex
print "m.lastgroup:", m.lastgroup

print "m.group():", m.group()
print "m.group(1,2):", m.group(1, 2)
print "m.groups():", m.groups()
print "m.groupdict():", m.groupdict()
print "m.start(2):", m.start(2)
print "m.end(2):", m.end(2)
print "m.span(2):", m.span(2)
```

re_complex.py

```
windard@windard:~/github/Python_Lib/code$ python re_complex.py
m.string: hello , world
m.re: <_sre.SRE_Pattern object at 0xb748b260>
m.re.pattern: (\w{1,6})(\s)(\,)(\s)(\w*)$
m.pos: 0
m.endpos: 13
m.lastindex: 5
m.lastgroup: None
m.group(): hello , world
m.group(1,2): ('hello', ' ')
m.groups(): ('hello', ' ', ',', ' ', 'world')
m.groupdict(): {}
m.start(2): 5
m.end(2): 6
m.span(2): (5, 6)
windard@windard:~/github/Python_Lib/code$
```

`+`

/	
`*?`	
`+?`	1
`??`	01
`{n, m}?`	nm
`{n,}?`	n

```
import re

m = re.match(r'<html>(.*)', "<html><body><title>this is title</title></body></html>>")

print m.group()

m = re.match(r'<html>(.*?)', "<html><body><title>this is title</title></body></html>>")

print m.group()
```

```
<html><body><title>this is title</title></body></html>>
<html>
```

/	
`(?:)`	
`(?P<name>)`	
`(?P=<name>)`	

```
In [1]: import re

In [2]: re.match(r"(?P<key>\w+):(?P<value>\d+)", "haha:1").groups()
Out[2]: ('haha', '1')

In [3]: re.match(r"(?P<key>\w+):(?P<value>\d+)", "haha:1").groupdict()
Out[3]: {'key': 'haha', 'value': '1'}

In [4]: re.search(r"((?P<key>\w+):(?P<value>\d+);)*", "haha:1;laal:2;").groups()
Out[4]: ('laal:2;', 'laal', '2')

In [5]: re.search(r"((?P<key>\w+):(?P<value>\d+);)*", "haha:1;laal:2;").groupdict()
Out[5]: {'key': 'laal', 'value': '2'}

In [6]: re.search(r"(?:(?P<key>\w+):(?P<value>\d+);)*", "haha:1;laal:2;").groups()
```

```
Out[6]: ('laal', '2')
```

regex

nginx

nginx

```
log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                  '$status $body_bytes_sent "$http_referer" '
                  '"$http_user_agent" "$http_x_forwarded_for"';
```

```
obj = re.compile(r'(?P<ip>.*?)- - \[(?P<time>.*?)\] "(?P<request>.*?)" (?P<status>.*?) (?P<bytes>.*?) "(?P<referer>.*?)" "(?P<ua>.*?)" "(?P<forwarded>.*?)"')
```

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

import re


def convert_camel_to_snake(hump_str):
    """

    :param hump_str:
    :return:
    """
    #
    p = re.compile(r'([a-z]|\d)([A-Z])')
    #
    sub = re.sub(p, r'\1_\2', hump_str).lower()
    return sub


if __name__ == '__main__':
    print(convert_camel_to_snake("AdId"))
    print(convert_camel_to_snake("CampaignId"))
    print(convert_camel_to_snake("coreUserID"))
    print(convert_camel_to_snake("FromWhereYouGo"))
    print(convert_camel_to_snake("PackageId"))
    print(convert_camel_to_snake("SocketServer"))
    print(convert_camel_to_snake("Python_Lib"))
```

> How to split but ignore separators in quoted strings, in python?

```
# coding=utf-8

import re

data = """part 1;"this is ; part 2;";'this is ; part 3';part 4;this "is ; part" 5"""
print(re.split(r'''((?:[^;"']|"[^"]*"|'[^']*')+)''', data)[1::2])
print(re.split(r''';(?=(?:[^'"]|'[^']*'|"[^"]*")*$)''', data))
```

30

json

- 2016-01-13
- 2017-03-18
- 2017-10-22
- 2018-06-21
- 2020-09-09

pythonjson `json.dumps()` `json.loads()` distPythonjsonjsonPython

> ujson simplejson

```
#  python  json
dump(obj, fp, skipkeys=False, ensure_ascii=True, check_circular=True, allow_nan=True, cls
=None, indent=None, separators=None, encoding='utf-8', default=None, sort_keys=False, **k
w)
#  python  json
dumps(obj, skipkeys=False, ensure_ascii=True, check_circular=True, allow_nan=True, cls=No
ne, indent=None, separators=None, encoding='utf-8', default=None, sort_keys=False, **kw)
#  json python
load(fp, encoding=None, cls=None, object_hook=None, parse_float=None, parse_int=None, par
se_constant=None, object_pairs_hook=None, **kw)
#  json  python
loads(s, encoding=None, cls=None, object_hook=None, parse_float=None, parse_int=None, par
se_constant=None, object_pairs_hook=None, **kw)
```

```
# coding=utf-8

import json

data = {
    'name' : 'ACME',
    'shares' : 100,
    'price' : 542.23,
    'others': ["first thing","second thing","third thing"]
}

json_str = json.dumps(data)
print json_str

python_str = json.loads(json_str)
print python_str
print python_str["name"]
print python_str["price"]
print python_str["others"][0]
```

json_demo.py

```
C:\Users\dell\.ssh\Python_Lib\code>python json_demo.py
{"price": 542.23, "name": "ACME", "shares": 100, "others": ["first thing", "second thing", "third thing"]}
{u'price': 542.23, u'name': u'ACME', u'shares': 100, u'others': [u'first thing', u'second thing', u'third thing']}
ACME
542.23
first thing

C:\Users\dell\.ssh\Python_Lib\code>
```

jsonPythondist jsonPython `pprint` `print` key

```
# coding=utf-8

import json
from pprint import pprint

data = {
    'name' : 'ACME',
    'shares' : 100,
    'price' : 542.23,
    'others': ["first thing","second thing","third thing"]
}

json_str = json.dumps(data)

python_str = json.loads(json_str)
pprint(python_str)
```

json_demo_2.py,

```
C:\Users\dell\.ssh\Python_Lib\code>python json_demo_2.py
{u'name': u'ACME',
 u'others': [u'first thing', u'second thing', u'third thing'],
 u'price': 542.23,
 u'shares': 100}

C:\Users\dell\.ssh\Python_Lib\code>
```

jsonPython

```
# coding=utf-8

import json

class JSONObject:
    def __init__(self,d):
        self.__dict__=d

data = {
    'name' : 'ACME',
    'shares' : 100,
    'price' : 542.23,
    'others': ["first thing","second thing","third thing"]
}
```

```
json_str = json.dumps(data)

python_str = json.loads(json_str, object_hook=JSONObject)
print isinstance(python_str,object)
print python_str.name
print python_str.price
print python_str.others[1]
```

json_object.py

```
C:\Users\dell\.ssh\Python_Lib\code>python json_object.py
True
ACME
542.23
second thing

C:\Users\dell\.ssh\Python_Lib\code>
```

json `pprint` json `dumps()` `indent=X`

2016-01-13

Python `eval` `str(unicode)` json

```
>>> data = {
...     'name' : 'ACME',
...     'shares' : 100,
...     'price' : 542.23,
...     'others': ["first thing","second thing","third thing"]
... }
>>> json_str = str(data)
>>> json_str
"{'price': 542.23, 'name': 'ACME', 'shares': 100, 'others': ['first thing', 'second thing
', 'third thing']}"
>>> eval(json_str)
{'price': 542.23, 'name': 'ACME', 'shares': 100, 'others': ['first thing', 'second thing'
, 'third thing']}
```

json true Python True json false Python False

```
>>> data = "{'name':'ACMA','status':false}"
>>> eval(data)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "<string>", line 1, in <module>
NameError: name 'false' is not defined
```

json

2017-03-18

json Python dict json Python

```
+------------------+---------------+
| Python           | JSON          |
+==================+===============+
| dict             | object        |
+------------------+---------------+
| list, tuple      | array         |
+------------------+---------------+
| str, unicode     | string        |
+------------------+---------------+
| int, long, float | number        |
+------------------+---------------+
| True             | true          |
+------------------+---------------+
| False            | false         |
+------------------+---------------+
| None             | null          |
+------------------+---------------+
```

```
JSONEncoder().encode({"foo": ["bar", "baz"]}) #  json
JSONDecoder().decode('{"foo": ["bar", "baz"]}') #  json
```

2017-10-22

json dict

- dict json
- dict json

2018-06-21

- `json.dumps(obj, indent=4)`
- `json.dumps(obj, separators=(',',':'))` . `(', ', ': ')`
- `json.dumps(obj, ensure_ascii=False)` utf-8 Unicode `\uXXXX`

2020-09-09

json `'{"price": 542.23, "name": "ACME", "shares": 100, "others": ["first thing", "second thing", "third thing"]}'` json key value `Invalid Control Character` `ValueError`

ACSII `\t` , `\n` , `\r`

ASCII

json `'{"price": 542.23, "name": "ACME", "sh\rares": 100, "others": ["first thing", "second\t thing", "third\n thing"]}'`

`strict=False`

```
In [28]: s = '{"price": 542.23, "name": "ACME", "shares": 100, "others": ["first thing",
"second thing", "third thing"]}'

In [29]: json.loads(s)
Out[29]:
{u'name': u'ACME',
 u'others': [u'first thing', u'second thing', u'third thing'],
 u'price': 542.23,
 u'shares': 100}

In [30]: s = '{"price": 542.23, "name": "ACME", "sh\rares": 100, "others": ["first thing"
, "second\t thing", "third\n thing"]}'

In [31]: json.loads(s)
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-31-48280973ea66> in <module>()
----> 1 json.loads(s)

/Users/bytedance/miniconda/envs/byted/lib/python2.7/json/__init__.pyc in loads(s, encodin
g, cls, object_hook, parse_float, parse_int, parse_constant, object_pairs_hook, **kw)
    337             parse_int is None and parse_float is None and
    338             parse_constant is None and object_pairs_hook is None and not kw):
--> 339         return _default_decoder.decode(s)
    340     if cls is None:
    341         cls = JSONDecoder

/Users/bytedance/miniconda/envs/byted/lib/python2.7/json/decoder.pyc in decode(self, s, _
w)
    362
    363         """
--> 364         obj, end = self.raw_decode(s, idx=_w(s, 0).end())
    365         end = _w(s, end).end()
    366         if end != len(s):

/Users/bytedance/miniconda/envs/byted/lib/python2.7/json/decoder.pyc in raw_decode(self,
s, idx)
    378         """
    379         try:
--> 380             obj, end = self.scan_once(s, idx)
    381         except StopIteration:
    382             raise ValueError("No JSON object could be decoded")

ValueError: Invalid control character at: line 1 column 38 (char 37)

In [32]: json.loads(s, strict=False)
```

```
Out[32]:
{u'name': u'ACME',
 u'others': [u'first thing', u'second\t thing', u'third\n thing'],
 u'price': 542.23,
 u'sh\rares': 100}
```

1. json key value
2. json json key `'\n{"price": 542.23,\n "name": "ACME", \t"shares": 100, "others": ["first thing", "second thing",\n "third thing"]}'`
3. json

```
In [34]: s = '\n{"price": 542.23,\n "name": "ACME", \t"shares": 100, "others": ["first th
ing", "second thing",\n "third thing"]}'

In [35]: json.loads(s)
Out[35]:
{u'name': u'ACME',
 u'others': [u'first thing', u'second thing', u'third thing'],
 u'price': 542.23,
 u'shares': 100}

In [37]: s= """{"price": 542.23, "name": "ACME", "shares": 100, "others": ["first thing",
 "second
    ...: thing", "third thing"]}"""

In [38]: s
Out[38]: '{"price": 542.23, "name": "ACME", "shares": 100, "others": ["first thing", "sec
ond \nthing", "third thing"]}'

In [39]: json.loads(s)
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-39-48280973ea66> in <module>()
----> 1 json.loads(s)

/Users/bytedance/miniconda/envs/byted/lib/python2.7/json/__init__.pyc in loads(s, encodin
g, cls, object_hook, parse_float, parse_int, parse_constant, object_pairs_hook, **kw)
    337             parse_int is None and parse_float is None and
    338             parse_constant is None and object_pairs_hook is None and not kw):
--> 339         return _default_decoder.decode(s)
    340     if cls is None:
    341         cls = JSONDecoder

/Users/bytedance/miniconda/envs/byted/lib/python2.7/json/decoder.pyc in decode(self, s, _
w)
    362
    363         """
--> 364         obj, end = self.raw_decode(s, idx=_w(s, 0).end())
    365         end = _w(s, end).end()
    366         if end != len(s):
```

```
/Users/bytedance/miniconda/envs/byted/lib/python2.7/json/decoder.pyc in raw_decode(self,
s, idx)
    378         """
    379         try:
--> 380             obj, end = self.scan_once(s, idx)
    381         except StopIteration:
    382             raise ValueError("No JSON object could be decoded")

ValueError: Invalid control character at: line 1 column 84 (char 83)
```

logging

-
-
-
-
  -
  -

print

```
# coding=utf-8

import logging

logging.info("hi, hello world")
logging.error("oops, what's wrong?")

logger = logging.getLogger(__name__)
logger.info("I'm forgot")
logger.error("What's see?")

print(logger.level)
print(logging.root.level)
```

```
ERROR:root:oops, what's wrong?
ERROR:__main__:What's see?
0
30
```

logging warning

```
#coding=utf-8

import logging
import sys

LEVELS = {'debug': logging.DEBUG,
          'info': logging.INFO,
          'warning': logging.WARNING,
          'error': logging.ERROR,
          'critical': logging.CRITICAL}
```

```
if len(sys.argv) > 1:
    level_name = sys.argv[1]
    level = LEVELS.get(level_name, logging.NOTSET)
    logging.basicConfig(level=level)

logging.debug('This is a debug message')
logging.info('This is an info message')
logging.warning('This is a warning message')
logging.error('This is an error message')
logging.critical('This is a critical error message')
```

logging log Level warning Level

`logging` warning

`logging.info` `root_logger` `logging.getLogger(__name__)` `logger`

`logging.basicConfig()`

> `basicConfig`

```
logging.basicConfig(
    level=logging.INFO,
    format='%(name)-25s %(asctime)s %(levelname)-8s %(lineno)-4d %(message)s',
    datefmt='[%Y-%m-%d %H:%M:%S]'
)
```

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

import logging
from logging.config import dictConfig


logging_config = {
    "version": 1,
    "disable_existing_loggers": False,
    "formatters": {"default": {"format": "%(asctime)s %(levelname)-8s %(filename)s:%(line
no)d %(message)s"}},
    "handlers": {
        "console": {"level": "INFO", "class": "logging.StreamHandler", "formatter": "defa
ult"},
        "file_logger": {"level": "INFO", "class": "logging.FileHandler", "formatter": "de
fault", "filename": "douban_spider.log"}
    },
    "root": {"handlers": ["file_logger", "console"], "level": "INFO"},
}
dictConfig(logging_config)
```

```
logger = logging.getLogger(__name__)
```

logging

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

import sys
import logging

logger = logging.getLogger("Socket Logging")
formatter = logging.Formatter('%(name)-12s %(asctime)s %(levelname)-8s %(lineno)-4d %(mes
sage)s', '%Y %b %d %a %H:%M:%S',)

file_handler = logging.FileHandler("SocketServer.log")
file_handler.setFormatter(formatter)

logger.addHandler(file_handler)

stream_handler = logging.StreamHandler(sys.stderr)
stream_handler.setFormatter(formatter)

logger.addHandler(stream_handler)

logger.setLevel(logging.DEBUG)
```

```
# coding=utf-8

import logging

def main():
    # Configure the logging system
    logging.basicConfig(
        filename='app.log',
        level=logging.ERROR,
        format='%(levelname)s:%(asctime)s:%(message)s'
    )

    # Variables (to make the calls that follow work)
    hostname = 'www.python.org'
    item = 'spam'
    filename = 'data.csv'
    mode = 'r'

    # Example logging calls (insert into your program)
    logging.critical('Host %s unknown', hostname)
    logging.error("Couldn't find %r", item)
    logging.warning('Feature is deprecated')
    logging.info('Opening file %r, mode=%r', filename, mode)
    logging.debug('Got here')
```

```
if __name__ == '__main__':
    main()
```

```
#coding=utf-8

import sys
import logging

logger = logging.getLogger("Test Logging")
formatter = logging.Formatter('%(name)-12s %(asctime)s %(levelname)-8s %(lineno)-4d %(mes
sage)s', '%Y%b%d %a %H:%M:%S')
file_handler = logging.FileHandler("test.log")
file_handler.setFormatter(formatter)
file_handler.setLevel(logging.DEBUG)

stream_handler = logging.StreamHandler(sys.stderr)
stream_handler.setFormatter(formatter)
stream_handler.setLevel(logging.WARNING)

logger.addHandler(file_handler)
logger.addHandler(stream_handler)
logger.setLevel(logging.DEBUG)

logger.debug('This is a debug message')
logger.info('This is an info message')
logger.warning('This is a warning message')
logger.error('This is an error message')

logger.removeHandler(stream_handler)
logger.critical('This is a critical error message')
```

log log log

log

`%(name)s`	Logger
`%(levelno)s`	
`%(levelname)s`	
`%(pathname)s`	
`%(filename)s`	
`%(module)s`	
`%(funcName)s`	
`%(lineno)d`	

`%(created)f`	UNIX
`%(relativeCreated)d`	Logger
`%(asctime)s`	“2003-07-08 16:49:45,896”
`%(thread)d`	ID
`%(threadName)s`	
`%(process)d`	ID
`%(message)s`	

`logger.exception` `logger.error` `error` ,

> logger `logger.info` `logger.error`

```
# coding=utf-8

import sys
import logging

logger = logging.getLogger(__name__)
formatter = logging.Formatter('%(name)-12s %(asctime)s %(levelname)-8s %(lineno)-4d %(mes
sage)s', '%Y %b %d %a %H:%M:%S',)

stream_handler = logging.StreamHandler(sys.stderr)
stream_handler.setFormatter(formatter)

logger.addHandler(stream_handler)
logger.setLevel(logging.DEBUG)

if __name__ == '__main__':
    logger.info("main start...")
    try:
        1 / 0
    except Exception as e:
        logger.exception("error %s", e)
    logger.info("main end.")
```

`logger.error`

```
__main__     2019 May 25 Sat 22:23:36 INFO     16   main start...
__main__     2019 May 25 Sat 22:23:36 ERROR    20   error integer division or modulo by z
ero
__main__     2019 May 25 Sat 22:23:36 INFO     21   main end.
```

`logger.exception`

```
__main__     2019 May 25 Sat 22:23:52 INFO     16   main start...
__main__     2019 May 25 Sat 22:23:52 ERROR    20   error integer division or modulo by z
ero
Traceback (most recent call last):
  File "code/logging_exception.py", line 18, in <module>
    1 / 0
ZeroDivisionError: integer division or modulo by zero
__main__     2019 May 25 Sat 22:23:52 INFO     21   main end.
```

`logging.handlers.TimedRotatingFileHandler`

ast

-
-
-

ast python python

```
+---------+   +------------+   +------------------------+   +----------------------+   +-
--------------+
|    +-->+    +-->+                +-->+               +-->+         +

|         |   | Parse Tree |   |  Abstract Syntax Tree  |   |  Control Flow Graph  |   |
 Code Object  |
+---------+   +------------+   +------------------------+   +----------------------+   +-
--------------+
```

ast ast

```
# coding=utf-8

import ast

code = """
def add(a, b):
    return a + b

add(1, 2)
"""

print "hello world"
print ast.parse(code)
print ast.dump(ast.parse(code))
```

`ast.parse` `compile`

```
def parse(source, filename='<unknown>', mode='exec'):
    """
    Parse the source into an AST node.
    Equivalent to compile(source, filename, mode, PyCF_ONLY_AST).
    """
    return compile(source, filename, mode, PyCF_ONLY_AST)
```

`_ast.Module` , `_ast.Assign` , `_ast.Str` , Module

```
Module(body=[FunctionDef(name='add', args=arguments(args=[Name(id='a', ctx=Param()), Name
(id='b', ctx=Param())], vararg=None, kwarg=None, defaults=[]), body=[Return(value=BinOp(l
eft=Name(id='a', ctx=Load()), op=Add(), right=Name(id='b', ctx=Load())))], decorator_list
=[]), Expr(value=Call(func=Name(id='add', ctx=Load()), args=[Num(n=1), Num(n=2)], keyword
s=[], starargs=None, kwargs=None))])
```

```
# coding=utf-8

import ast


code = """
def add(a, b):
    return a + b

print add(1, 2)
"""


class CrazyTransformer(ast.NodeTransformer):

    def visit_BinOp(self, node):
        print node.__dict__
        node.op = ast.Mult()
        print node.__dict__
        return node


def main():
    module = ast.parse(code)
    exec compile(module, '<string>', 'exec')
    transformer = CrazyTransformer()
    multi = transformer.visit(module)
    exec compile(multi, '<string>', 'exec')


if __name__ == '__main__':
    main()
```

ast

unparse.py

`codegen` , `astunparse` , `astor`

```
# coding=utf-8

import ast
import unparse

code = """
def add(a, b):
    return a + b

print add(1, 2)
"""


class CrazyTransformer(ast.NodeTransformer):

    def visit_BinOp(self, node):
        node.op = ast.Mult()
        return node


def back():
    module = ast.parse(code)
    transformer = CrazyTransformer()
    multi = transformer.visit(module)
    unparse.Unparser(multi)


if __name__ == '__main__':
    back()
```

```
# coding=utf-8

import ast
import codegen
import astunparse
import astor


code = """
data = {
    "key": "value",
    "list": [1,2,3]
}

def add(a, b):
    return a + b

print add(1, 2)
"""
```

```
module = ast.parse(code)

print codegen.to_source(module)
print astunparse.unparse(module)
print astor.to_source(module)
```

3.9 `ast.unparse`

Python `eval` , `eval` `ast.literal_eval`

`(•̀ᴗ•́)` Python `None`

> `1+2j`

```
def literal_eval(node_or_string):
    """
    Safely evaluate an expression node or a string containing a Python
    expression.  The string or node provided may only consist of the following
    Python literal structures: strings, numbers, tuples, lists, dicts, booleans,
    and None.
    """
    _safe_names = {'None': None, 'True': True, 'False': False}
    if isinstance(node_or_string, basestring):
        node_or_string = parse(node_or_string, mode='eval')
    if isinstance(node_or_string, Expression):
        node_or_string = node_or_string.body
    def _convert(node):
        if isinstance(node, Str):
            return node.s
        elif isinstance(node, Num):
            return node.n
        elif isinstance(node, Tuple):
            return tuple(map(_convert, node.elts))
        elif isinstance(node, List):
            return list(map(_convert, node.elts))
        elif isinstance(node, Dict):
            return dict((_convert(k), _convert(v)) for k, v
                        in zip(node.keys, node.values))
        elif isinstance(node, Name):
            if node.id in _safe_names:
                return _safe_names[node.id]
        elif isinstance(node, BinOp) and \
             isinstance(node.op, (Add, Sub)) and \
             isinstance(node.right, Num) and \
             isinstance(node.right.n, complex) and \
             isinstance(node.left, Num) and \
             isinstance(node.left.n, (int, long, float)):
            left = node.left.n
            right = node.right.n
            if isinstance(node.op, Add):
```

```
                return left + right
            else:
                return left - right
        raise ValueError('malformed string')
    return _convert(node_or_string)
```

> Python 3.2

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

import ast


if __name__ == '__main__':
    #
    # print ast.literal_eval("a=1")
    # print eval("a=1")
    # a = 1
    #
    # print ast.literal_eval("1+1")
    # print eval("1+1")
    # print ast.literal_eval("1==1")
    print eval("1==1")
    print ast.literal_eval("1")
    print ast.literal_eval("None")
    #
    # print ast.literal_eval("{1,2,4}")
    # print ast.literal_eval("set([1])")
    # print ast.literal_eval("[1,2,{'1', 2, '2,3,4'}, [4,5,'6']]")
    # print [1,2,{'1', 2, '2,3,4'}, [4,5,'6']]
    print ast.literal_eval("[1,2,3,{2:3}]")
    # list
    # print ast.literal_eval("list([1,2,3])")
    print list([1, 2, 3])
    # print ast.literal_eval("[1,2+3]")
    #
    print ast.literal_eval("1+2j")
    print ast.literal_eval("-2")
    # print ast.literal_eval("--2")
    #
    # print ast.literal_eval("+2")
    # print ast.literal_eval("++2")
```

ply

Python Lex-Yacc，Python Lex-Yacc

> `*nix` Lex Yacc try it youself .

if,else

if,else

yacc

ast

if,else yacc

Lex&Yacc

Lex Yacc

Lex Yacc Lex Yacc

Lex

lex_start.lt

```
%{
#include <stdio.h>
%}

%%
stop printf("Stop command received");
start printf("Start command reveived");
%%
```

lex

1. `printf` `stdio.h` C `%{` `%}`
2. `%%` `%%` lex `stop` `printf("Stop command received");` , `start`

`lex lex_start.lt` `lex.yy.c` C `cc lex.yy.c -o lex_start -ll` `lex_start`

`start` `Start command reveived` , `stop` `Stop command received` , `start` `stop`

```
$ ./lex_start
start
Start command reveived
stop
Stop command received
yes
yes
no
no
!
!
^C
```

lex C

```
[0123456789]+         #
[0-9]+                 #
[a-z]                 # a-z
[a-z]*                 # 0
```

lex_regex.lt

```
%{
#include <stdio.h>
%}

%%
[0-9]+                        printf("NUMBER\n");
[a-zA-Z][a-zA-Z0-9]*        printf("WORD\n");
%%
```

```
$ lex lex_regex.lt
$ cc lex.yy.c -o lex_regex -ll
$ ./lex_regex
2334
```

```
NUMBER

dervr
WORD

str1
WORD

123avc
NUMBER
WORD

-12
-NUMBER

12.43
NUMBER
.NUMBER

a@a
WORD
@WORD

^C
```

.

Lex/YACC

threading

-
  -
  -
-
- -
  -
-
-
-
-
-
-
-
-

`threading` `Java`

1. `threading.Thread` `start`
2. `threading.Thread` `run` `run`
   `threading.Thread`

```
# coding=utf-8

import threading
from time import ctime,sleep

def loop(nloop,nsec):
    print "loop",nloop," start at: ",ctime()
    sleep(nsec)
    print "loop",nloop,"end    at: ",ctime()

print "all start at: ",ctime()
loops = [4,2]
threads = []
nloops = range(len(loops))

#
for i in nloops:
    t = threading.Thread(target=loop,args=(i,loops[i]))
    threads.append(t)

#
```

```
for i in nloops:
    threads[i].start()

print "all end   at: ",ctime()
```

`threading_start.py`

```
$ python threading_start.py
all start at:  Wed Oct  7 15:13:32 2020
loop 0  start at:  Wed Oct  7 15:13:32 2020
 loop 1  start at:  Wed Oct  7 15:13:32 2020
 all end   at:  Wed Oct  7 15:13:32 2020

loop 1 end    at:  Wed Oct  7 15:13:34 2020
loop 0 end    at:  Wed Oct  7 15:13:36 2020
```

244

```

```

```
# coding=utf-8

import threading
from time import ctime,sleep

class ThreadFunc(object):
    def __init__(self, func, args, name=""):
        self.args = args
        self.func = func
        self.name = name

    def __call__(self):
        apply(self.func, self.args)

def loop(nloop,nsec):
    print "loop",nloop," start at: ",ctime()
    sleep(nsec)
    print "loop",nloop,"end    at: ",ctime()

print "all start at: ",ctime()

loops = [4,2]
threads = []
nloops = range(len(loops))

for i in nloops:
    t = threading.Thread(target=ThreadFunc(loop,(i,loops[i]),loop.__name__))
    threads.append(t)

for i in nloops:
```

```
    threads[i].start()

for i in nloops:
    threads[i].join()

print "all end   at: ",ctime()
```

`threading_class.py`

```
C:\Users\dell\.ssh\Python_Lib\code>python threading_class.py
all start at:  Thu Dec 10 22:52:05 2015
loooploop 1  start at:  0 Thu Dec 10 22:52:05 2015  start at:
 Thu Dec 10 22:52:05 2015
loop 1 end    at:  Thu Dec 10 22:52:07 2015
loop 0 end    at:  Thu Dec 10 22:52:09 2015
all end   at:  Thu Dec 10 22:52:09 2015
```

`start` `join`

> `join` `join`
> `jion` `join`
> `join`

```
#coding=utf-8

import threading
from time import ctime,sleep

class ThreadFunc(object):
    def __init__(self, func ,args,name=""):
        self.args = args
        self.func = func
        self.name = name

    def __call__(self):
        apply(self.func,self.args)

def loop(nloop,nsec):
    print "loop",nloop," start at: ",ctime()
    sleep(nsec)
    print "loop",nloop,"end    at: ",ctime()

print "all start at: ",ctime()

loops = [4,2]
threads = []
nloops = range(len(loops))

for i in nloops:
    t = threading.Thread(target=ThreadFunc(loop,(i,loops[i]),loop.__name__))
    threads.append(t)

for i in nloops:
```

```
    threads[i].start()

# for i in nloops:
#     threads[i].join()
#     print i, "is joined at: ", ctime()

#  join
threads[0].join()
print "0 is joined at: ", ctime()
threads[1].join()
print "1 is joined at: ", ctime()

print "all end   at: ",ctime()
```

`join` `join`

```
$ python code/threading_class_2.py
all start at:  Wed Oct  7 15:34:05 2020
loop 0loop 1  start at:  Wed Oct  7 15:34:05 2020
  start at:  Wed Oct  7 15:34:05 2020
loop 1 end    at:  Wed Oct  7 15:34:07 2020
loop 0 end    at:  Wed Oct  7 15:34:09 2020
0 is joined at:  Wed Oct  7 15:34:09 2020
1 is joined at:  Wed Oct  7 15:34:09 2020
all end   at:  Wed Oct  7 15:34:09 2020
```

```
# coding=utf-8

import threading
from time import ctime,sleep

class MyThread(threading.Thread):
    def __init__(self, func, args, name=""):
        threading.Thread.__init__(self)
        self.name = name
        self.func = func
        self.args = args

    def run(self):
        apply(self.func, self.args)

def loop(nloop,nsec):
    print "loop",nloop," start at: ",ctime()
    sleep(nsec)
    print "loop",nloop,"end    at: ",ctime()

print "all start at: ",ctime()

loops = [4,2]
```

```
threads = []
nloops = range(len(loops))

for i in nloops:
    t = MyThread(loop,(i,loops[i]),loop.__name__)
    threads.append(t)

for i in nloops:
    threads[i].start()

for i in nloops:
    threads[i].join()

print "all end   at: ",ctime()
```

threading_class_MyThread.py

```
C:\Users\dell\.ssh\Python_Lib\code>python threading_class_MyThread.py
all start at:  Thu Dec 10 23:43:52 2015
looploop 0 1  start at:   start at:  Thu Dec 10 23:43:52 2015 Thu Dec 10 23:43:52 2015

loop 1 end    at:  Thu Dec 10 23:43:54 2015
loop 0 end    at:  Thu Dec 10 23:43:56 2015
all end   at:  Thu Dec 10 23:43:56 2015
```

```
#coding=utf-8

import threading
from time import ctime,sleep


class ThreadDemo(threading.Thread):
    def __init__(self, nloop, nsec, name=""):
        threading.Thread.__init__(self)
        self.name = name
        self.nloop = nloop
        self.nsec = nsec

    def run(self):
        print "loop", self.nloop, "start  at: ", ctime()
        sleep(self.nsec)
        print "loop", self.nloop, "end    at: ", ctime()


print "all start at: ",ctime()

loops = [4,2]
threads = []
nloops = range(len(loops))

for i in nloops:
    t = ThreadDemo(i, loops[i], "loop")
    threads.append(t)
```

```
for i in nloops:
    threads[i].start()

for i in nloops:
    threads[i].join()

print "all end   at: ", ctime()
```

`start` `threading.Thread` `start`

`start`

```
class Thread(_Verbose):
    def __init__(self, group=None, target=None, name=None,
                 args=(), kwargs=None, verbose=None):
        """This constructor should always be called with keyword arguments. Arguments are
:

        *group* should be None; reserved for future extension when a ThreadGroup
        class is implemented.

        *target* is the callable object to be invoked by the run()
        method. Defaults to None, meaning nothing is called.

        *name* is the thread name. By default, a unique name is constructed of
        the form "Thread-N" where N is a small decimal number.

        *args* is the argument tuple for the target invocation. Defaults to ().

        *kwargs* is a dictionary of keyword arguments for the target
        invocation. Defaults to {}.

        If a subclass overrides the constructor, it must make sure to invoke
        the base class constructor (Thread.__init__()) before doing anything
        else to the thread.

"""
        assert group is None, "group argument must be None for now"
        _Verbose.__init__(self, verbose)
        if kwargs is None:
            kwargs = {}
        self.__target = target
        self.__name = str(name or _newname())
        self.__args = args
        self.__kwargs = kwargs
        self.__daemonic = self._set_daemon()
        self.__ident = None
        self.__started = Event()
        self.__stopped = False
        self.__block = Condition(Lock())
```

```
        self.__initialized = True
        # sys.stderr is not stored in the class like
        # sys.exc_info since it can be changed between instances
        self.__stderr = _sys.stderr

    def run(self):
        """Method representing the thread's activity.

        You may override this method in a subclass. The standard run() method
        invokes the callable object passed to the object's constructor as the
        target argument, if any, with sequential and keyword arguments taken
        from the args and kwargs arguments, respectively.

        """
        try:
            if self.__target:
                self.__target(*self.__args, **self.__kwargs)
        finally:
            # Avoid a refcycle if the thread is running a function with
            # an argument that has a member that points to the thread.
            del self.__target, self.__args, self.__kwargs
```

> `append`, `start`, `join`
>
> `append` `start` `start` `start`
>
> `join` `join` `join`
>
> `join` `start`

```
#coding=utf-8

import threading
from time import ctime,sleep


class ThreadDemo(threading.Thread):
    def __init__(self, nloop, nsec, name=""):
        threading.Thread.__init__(self)
        self.name = name
        self.nloop = nloop
        self.nsec = nsec

    def run(self):
        print "loop", self.nloop, "start  at: ", ctime()
        sleep(self.nsec)
        print "loop", self.nloop, "end    at: ", ctime()


print "all start at: ",ctime()

loops = [4,2]
threads = []
nloops = range(len(loops))
```

```
for i in nloops:
    t = ThreadDemo(i, loops[i], "loop")
    threads.append(t)
    t.start()

for i in nloops:
    threads[i].join()

print "all end   at: ",ctime()
```

```
# coding=utf-8

import threading
from time import ctime,sleep

def loop(nloop,nsec):
    print "loop",nloop," start at: ",ctime()
    sleep(nsec)
    print "loop",nloop,"end    at: ",ctime()

print "all start at: ",ctime()
loops = [4,2]
threads = []
nloops = range(len(loops))

#
for i in nloops:
    t = threading.Thread(target=loop,args=(i,loops[i]))
    threads.append(t)

#
for i in nloops:
    threads[i].start()

#
#joinstart
#
#
#
#
for i in nloops:
    threads[i].join()

print "all end   at: ",ctime()
```

threading_demo.py

```
C:\Users\dell\.ssh\Python_Lib\code>python threading_demo.py
all start at:  Wed Dec 09 21:01:23 2015
loop 0loop  start at:  1 Wed Dec 09 21:01:23 2015
  start at:  Wed Dec 09 21:01:23 2015
loop 1 end    at:  Wed Dec 09 21:01:25 2015
loop 0 end    at:  Wed Dec 09 21:01:27 2015
all end   at:  Wed Dec 09 21:01:27 2015
```

Python `thread` , `threading` `thread`

`threading` `Daemon` Python

1.
2. ,

`threading.Thread.setDaemon(True)`

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

import threading
from time import ctime, sleep


def loop(nloop, nsec):
    print("loop", nloop, " start at: ", ctime())
    sleep(nsec)
    print("loop", nloop, "end    at: ", ctime())


if __name__ == '__main__':

    print("all start at: ", ctime())
    loops = [4, 2]
    threads = []
    nloops = range(len(loops))

    #
    for i in nloops:
        t = threading.Thread(target=loop, args=(i, loops[i]))
        t.setDaemon(True)
        threads.append(t)

    #
    for i in nloops:
        threads[i].start()

    print("all end   at: ", ctime())
```

.

```
$ python threading_start_daemon.py
all start at:  Thu Oct  8 10:28:57 2020
```

```
loop 0loop 1  start at:  Thu Oct  8 10:28:57 2020
 all end   at:  Thu Oct  8 10:28:57 2020
  start at:  Thu Oct  8 10:28:57 2020
```

> `start` `join`
> (join)
> `join` `join`
> `join`

`join`

> (start)

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

from threading import Thread
import os
import time


def sleeper(name, seconds):
    print 'starting child process with id: ', os.getpid()
    print 'parent process:', os.getppid()
    print 'sleeping for %s ' % seconds
    time.sleep(seconds)
    print "%s done sleeping" % name


if __name__ == '__main__':
    print "in parent process (id %s)" % os.getpid()
    p = Thread(target=sleeper, args=('bob', 5))
    print 'daemon?', p.isDaemon()
    p.setDaemon(not p.isDaemon())
    print 'daemon?', p.isDaemon()
    p.start()
    print "in parent process after child process start"
    print "parent process about to join child process"
    p.join()
    print "in parent process after child process join"
    print "parent process exiting with id ", os.getpid()
    print "The parent's parent process:", os.getppid()
```

```
#coding=utf-8

import threading
from time import ctime,sleep
```

```
class DaemonThreadDemo(threading.Thread):
    def __init__(self, nloop, nsec, name=""):
        threading.Thread.__init__(self)
        self.name = name
        self.nloop = nloop
        self.nsec = nsec

    def run(self):
        print "loop", self.nloop, "start  at: ", ctime()
        sleep(self.nsec)
        print "loop", self.nloop, "end    at: ", ctime()


print "all start at: ",ctime()

loops = [4,2]
threads = []
nloops = range(len(loops))

for i in nloops:
    t = DaemonThreadDemo(i, loops[i], "loop")
    t.setDaemon(True)
    t.start()
    threads.append(t)

for i in nloops:
    threads[i].join()

print "all end   at: ",ctime()
```

-

-

Queue
1001

```
# coding=utf-8

import threading
from Queue import Queue
from random import random
from time import ctime,sleep

def writeQ(queue):
    for i in range(100):
        print "Producting project for Q..."
        sleep(random())
        # sleep(random()/2.0)
        queue.put('xxx',1)
        print "Size now",queue.qsize()
```

```
def readQ(queue):
    for i in range(100):
        print "Consuming project from Q..."
        sleep(random())
        queue.get(1)
        print "Size now",queue.qsize()

print "all start at: ",ctime()

funcs = [writeQ,readQ]
nfunc = range(len(funcs))

q = Queue(48)
threads = []

for i in nfunc:
    t = threading.Thread(target=funcs[i],args=(q,))
    threads.append(t)

for i in nfunc:
    threads[i].start()

for i in nfunc:
    threads[i].join()

print "all end   at: ",ctime()
```

threading_queue.py

50

```
# coding=utf-8

import threading
from Queue import Queue
from random import random
from time import ctime,sleep

def writeQ(queue):
    for i in range(25):
        print "Producting project for Q..."
        sleep(random())
        queue.put('xxx',1)
        print "Size now",queue.qsize()

def readQ(queue):
    for i in range(25):
```

```
        print "Consuming project from Q..."
        sleep(random())
        queue.get(1)
        print "Size now",queue.qsize()

print "all start at: ",ctime()

funcs = [writeQ,readQ]
nfunc = range(len(funcs))

q = Queue(48)
threads = []

for i in nfunc:
    for j in range(4):
        t = threading.Thread(target=funcs[i],args=(q,))
        threads.append(t)

for i in range(8):
    threads[i].start()

for i in range(8):
    threads[i].join()

print "all end   at: ",ctime()
```

threading_queue_last.py

```
# coding=utf-8

import threading
from time import ctime,sleep

counter = 0

class MyThread1(threading.Thread):
    def __init__(self):
        threading.Thread.__init__(self)

    def run(self):
        global counter
        counter += 1
        print "  "+str(counter)+"  "

class MyThread2(threading.Thread):
    def __init__(self):
```

```
        threading.Thread.__init__(self)

    def run(self):
        global counter
        counter -= 1
        print "  "+str(counter)+"  "

if __name__ == '__main__':
    threads = []
    for i in range(20):
        if i%2:
            t = MyThread1()
        else:
            t = MyThread2()
        threads.append(t)

    for t in threads:
        t.start()
```

Threading_nolock.py

```
C:\Users\dell\.ssh\Python_Lib\code (master)
λ python threading_nolock.py
 -1   0

 0  -1

 -1
 0   -1    0


 -1   -1   0


 0   -1

 -1   0

 -1   0

 -1   0
 0
```

```
# coding=utf-8

import threading
from time import ctime,sleep

counter = 0
lock = threading.Lock()

class MyThread1(threading.Thread):
    def __init__(self):
        threading.Thread.__init__(self)
```

```
    def run(self):
        if lock.acquire():
            global counter
            counter += 1
            print "  "+str(counter)+"  "
            lock.release()

class MyThread2(threading.Thread):
    def __init__(self):
        threading.Thread.__init__(self)

    def run(self):
        if lock.acquire():
            global counter
            counter -= 1
            print "  "+str(counter)+"  "
            lock.release()

if __name__ == '__main__':
    threads = []
    for i in range(20):
        if i%2:
            t = MyThread1()
        else:
            t = MyThread2()
        threads.append(t)

    for t in threads:
        t.start()
```

threading_lock.py

```
C:\Users\dell\.ssh\Python_Lib\code (master)
λ spython threading_lock.py
  -1
  0
  -1
  0
  -1
  0
  -1
  0
  -1
  0
  -1
  0
  -1
  0
  -1
  0
  -1
  0
  -1
  0
  -1
  0
```

，

```
def run(self):
    with lock:
        global counter
        counter -= 1
        print "  "+str(counter)+"  "
```

Lock RLock Lock RLock

threadLocal

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

import threading


local = threading.local()


def process_name():
    print "hello %s, in %s" % (local.name, threading.current_thread().name)


def process_local(name):
    local.name = name
    process_name()


if __name__ == '__main__':
    local.name = 'Cli'
    process_name()
    t1 = threading.Thread(target=process_local, args=('Bob', ), name='Target-A')  # noqa
    t2 = threading.Thread(target=process_local, args=('Alice', ), name='Target-B')  # noqa


    t1.start()
    t2.start()

    t1.join()
    t2.join()
```

flask flask

-

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

import threading
local = threading.local()


def show_local():
    print "I'm child"
    print "local is %r" % local
    print "local value %s" % hasattr(local, "name")


if __name__ == '__main__':
    print "I'm child"
    print "local is %r" % local
    print "local value %s" % hasattr(local, "name")
    local.name = "father"
    print "local value %s" % local.name

    threading.Thread(target=show_local).start()

    print "local value %s" % local.name
```

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

import os
import time
import random
import threading

def long_time_task(name):
    print 'Running task %s (%s)' % (name, os.getpid())
    start = time.time()
    time.sleep(random.random() * 5)
    end = time.time()
    print 'Task %s run %0.2f econds.' % (name, end - start)


if __name__ == '__main__':
    for i in xrange(10):
        threading.Thread(target=long_time_task, args=(str(i))).start()

    for i in xrange(10):
        print threading.enumerate(), len(threading.enumerate())
        time.sleep(1)
```

event

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

import time
import logging
from threading import Thread, Event
from logging.config import dictConfig


logging_config = {
    "version": 1,
    "disable_existing_loggers": False,
    "formatters": {"default": {"format": "%(asctime)s %(levelname)-8s %(filename)s:%(line
no)d %(message)s"}},
    "handlers": {
        "console": {"level": "INFO", "class": "logging.StreamHandler", "formatter": "defa
ult"},
        "file_logger": {
            "level": "INFO",
            "class": "logging.FileHandler",
            "formatter": "default",
            "filename": "threading.log",
        },
    },
    "root": {"handlers": ["file_logger", "console"], "level": "INFO"},
}
dictConfig(logging_config)
stop_event = Event()


def continue_thread_run():
    logger = logging.getLogger(__name__)
    while True:
        if stop_event.is_set():
            break
        logger.info("wait for stop_event")
        time.sleep(1)


def wait_for_time():
    logger = logging.getLogger(__name__)
    while True:
        if int(time.time()) > 1615711196:
            stop_event.set()
            break
        logger.info("stop_event not ready:%d" % int(time.time()))
        time.sleep(1)
```

```
if __name__ == '__main__':
    thread_list = [Thread(target=continue_thread_run), Thread(target=wait_for_time)]
    for thread in thread_list:
        # thread.setDaemon(True)
        thread.start()

    for thread in thread_list:
        thread.join()
```

()

1.
2.
3.

-
-

-
-
-

Python `concurrent`

, `Ctrl + C` kill signal main Mac `Ctrl + \` quit

,

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

import sys, time, threading, abc
from optparse import OptionParser

def parse_options():
    parser = OptionParser()
    parser.add_option("-t", action="store", type="int", dest="threadNum", default=1,
                      help="thread count [1]")
    (options, args) = parser.parse_args()
    return options
```

```
class thread_sample(threading.Thread):
    def __init__(self, name):
        threading.Thread.__init__(self)
        self.name = name
        self.kill_received = False

    def run(self):

        while not self.kill_received:
            # your code
            print self.name, "is active"
            time.sleep(1)

def has_live_threads(threads):
    return True in [t.isAlive() for t in threads]

def main():
    options = parse_options()
    threads = []

    for i in range(options.threadNum):
            thread = thread_sample("thread#" + str(i))
            thread.start()
            threads.append(thread)

    while has_live_threads(threads):
        try:
            # synchronization timeout of threads kill
            [t.join(1) for t in threads
             if t is not None and t.isAlive()]
        except KeyboardInterrupt:
            # Ctrl-C handling and send kill to threads
            print "Sending kill to threads..."
            for t in threads:
                t.kill_received = True

    print "Exited"

if __name__ == '__main__':
   main()
```

```
# -*- coding: utf-8 -*-
import time
import threading


def on_timer():
    print time.time()
    set_timer()


def set_timer():
    _timer = threading.Timer(10, on_timer)
    _timer.start()


set_timer()
while 1:
    time.sleep(5)
    print 'sleep', time.time()
```

concurrent

python `threading` `multiprocessing`

`ThreadPoolExecutor` `ProcessPoolExecutor`

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

import time
import requests
import functools
from concurrent import futures


def time_count(func):
    @functools.wraps(func)
    def wrapper(*args, **kwargs):
        start = time.time()
        result = func(*args, **kwargs)
        end = time.time()
        print "time", end - start
        return result
    return wrapper


urls = ['https://ele.me',
        'https://baidu.com',
        'https://jd.com',
        'https://v2ex.com',
        'https://windard.com',
        'https://taobao.com',
        'https://zhihu.com',
        'https://vip.com',
        'https://t.tt']


@time_count
def main():
    executor = futures.ThreadPoolExecutor()
    roads = []
    results = []
    for url in urls:
        future = executor.submit(requests.get, url)
        roads.append(future)

    for future in futures.as_completed(roads):
        result = future.result()
        results.append(result.status_code)

    executor.shutdown()
    return results
```

```
@time_count
def sync_main():
    with futures.ThreadPoolExecutor() as executor:
        roads = executor.map(requests.get, urls)
        results = [result.status_code for result in roads]
    return results


@time_count
def async_main():
    results = []
    for url in urls:
        results.append(requests.get(url).status_code)
    return results


if __name__ == '__main__':
    print main()
    print async_main()
    print sync_main()
```

Python-PoolExecutor
Python concurrent.futures
pythonconcurrent.futures
python 1 futures

pdb

pycharm

```
import pdb
pdb.set_trace()
```

```
import pdb;pdb.set_trace()
```

> `python -m pdb file.py` pdb

pdb

1. `help [command]`
2. `n | next`
3. `j | jump [lineno]`
4. `c | continue` , pdb
5. `q | exit | quit`
6. `l | list`
7. `w | where | bt`
8. `b | break [lineno]`
9. `cl | clear`
10. `a | args`
11. `r | return`
12. `whatis [arg]`
13. `s | step`

pdb

> `a,b,c` `a,b,c`

 2021-03-14 10:34:30

pip python

pip Redis MySQL

requests

requestsHTTP client for human

 2020-09-23 04:45:19

redis

openpyxl

`xlwt` Excel 65535

> Mac Numbers 65535
> 😁 numbers xls xlsx 104 xls 65535

```
# coding=utf-8

import openpyxl


filename = 'sku.xlsx'


def readExel():
    inwb = openpyxl.load_workbook(filename)  #
    sheetnames = inwb.get_sheet_names()  # sheet
    ws = inwb.get_sheet_by_name(sheetnames[0])  # sheet

    # sheet
    rows = ws.max_row
    cols = ws.max_column
    for r in range(1, rows):
        for c in range(1, cols):
            print(str(ws.cell(row=r, column=c).value))
        if r == 10:
            break


def writeExcel():
    outwb = openpyxl.Workbook()  #
    outws = outwb.create_sheet(index=0)  # sheet
    for row in range(1, 70000):
        for col in range(1, 4):

            # outws.cell(row, col).value = row*2  #
            outws.cell(row=row, column=col, value=row * 2)  #
        print(row)
    outwb.save(filename)  #


if __name__ == '__main__':
    # writeExcel()
    readExel()
```

loguru

-
-
-
-
-

`logging` `loguru`

> `loguru` C

`logging` warning

> , `loguru` python3

```
# coding=utf-8

from loguru import logger


logger.debug("Hi")
logger.info("hello world")
logger.warning("I'm comming")
logger.error("Be careful")
```

```
2020-10-18 11:37:54.804 | DEBUG    | __main__:<module>:6 - Hi
2020-10-18 11:37:54.805 | INFO     | __main__:<module>:7 - hello world
2020-10-18 11:37:54.805 | WARNING  | __main__:<module>:8 - I'm comming
2020-10-18 11:37:54.805 | ERROR    | __main__:<module>:9 - Be careful
```

> `logging`

```
# coding=utf-8

import sys
from loguru import logger


logger.add("file_{time}.log")
```

```
logger.info("hello, world!")

logger.add(sys.stderr, colorize=True, format="<green>{time}</green> <level>{message}</lev
el>", level="INFO")
logger.debug("not important message")
logger.info("aha, I'm here")
```

```
logger.add(LOG_FILE, rotation = "200KB", compression="zip")
```

```
# coding=utf-8

import sys
from loguru import logger


logger.add("file_{time}.log", rotation = "200KB", compression="zip", retention=1)
logger.info("hello, world!")

for i in range(10000):
    logger.debug("not important message")
    logger.info("aha, I'm here")
```

`logger.remove()` `handle` , `add` `handle_id` `handle`

, `logger.exception` `logger.error` `loguru`

```
# coding=utf-8

from loguru import logger


def division(a, b):
    try:
        c = a / b
        return c
    except Exception as e:
        logger.exception(e)



if __name__ == '__main__':
    division(3, 0)
```

```
python loguru_exception.py
2020-10-18 15:29:55.401 | ERROR    | __main__:division:11 - division by zero
Traceback (most recent call last):

  File "loguru_exception.py", line 16, in <module>
    division(3, 0)
    └ <function division at 0x7ff37d568ef0>

> File "loguru_exception.py", line 8, in division
    c = a / b
        |   └ 0
        └ 3

ZeroDivisionError: division by zero
```

`loguru.add` `logging.Handler` `loguru` `logging` `handle` ,

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

import logging
from loguru import logger
from logging.config import dictConfig

logging_config = {
    "version": 1,
    "disable_existing_loggers": False,
    "formatters": {"default": {"format": "%(asctime)s %(levelname)-8s %(filename)s:%(line
no)d %(message)s"}},
    "handlers": {
        "console": {"level": "INFO", "class": "logging.StreamHandler", "formatter": "defa
ult"},
        "file_logger": {"level": "INFO", "class": "logging.FileHandler", "formatter": "de
fault", "filename": "file.log"}
    },
    "root": {"handlers": ["file_logger", "console"], "level": "INFO"},
}
dictConfig(logging_config)
logger.remove()

for handle in logging.root.handlers:
    logger.add(handle)

# logger = logging.getLogger(__name__)

# class InterceptHandler(logging.Handler):
#     def emit(self, record):
#         try:
```

```
#             level = logger.level(record.levelname).name
#         except ValueError:
#             level = record.levelno
#
#         frame, depth = logging.currentframe(), 2
#         while frame.f_code.co_filename == logging.__file__:
#             frame = frame.f_back
#             depth += 1
#
#         logger.opt(depth=depth, exception=record.exc_info).log(level, record.getMessage
())
#
#
# logging.basicConfig(handlers=[InterceptHandler()], level=0)


if __name__ == '__main__':

    logger.info("Test in Fusion")
```

loguru
logginloguru

1. pythonweb, `localhost:8080`

```
python -m SimpleHTTPServer 8080
```

> `python -m http.server` for python 3

CGI

```
python -m CGIHTTPServer 8080
```

FTP

```
python -m pyftpdlib
```

```
python -m pydoc -p 8000
```

PHP 5.4web `localhost:8000`

```
php -S localhost:8000
```

nodejs

```
npm install http-server -g
```

`http-server` 8080

nodejs

```
var http = require('http');
http.createServer(function (req, res) {
    res.send('Hello');
    res.end();
}).listen(3000);
```

server.js,cmd `node server.js` `localhost:3000`

python

1. Python shell Python , `python -c "import request"`
2. python2 python3 Python2 python , pyhton3 Python34 , pip python3 pip `python34 -m pip install requests` , pip3.py

> python3 script
> `python -m pip install requests` python

```
# coding=utf-8
import sys
import subprocess
print(" ".join(sys.argv))
cmd = "python34 -m pip " + " ".join(sys.argv[1:] + " -i http://pypi.douban.com/simple --trusted-host pypi.douban.com")
obj = subprocess.Popen(cmd)
obj.wait()
```

urlencode urldecode

- `urllib.urlencode` <-> `urlparse.parse_qsl` | `query_string.query_string`
- `urllib.quote` <-> `urllib.unquote`
- `urlparse.urlparse` <-> `urlparse.urlunparse`

- `re.escape`
- `cgi.escape` XSS

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

import urlparse
import urllib
from query_string import query_string

data = {
    'name': 'windard',
    'year': 23,
    'price': 100000000.111,
    'company': 'https://ele.me'
}

if __name__ == '__main__':
    raw_data = urllib.urlencode(data)
    print raw_data
    parse_data = urlparse.parse_qsl(raw_data)
    print parse_data
    query_data = query_string(raw_data)
    print query_data
```

python json

```
cat test.json | python -m json.tool
echo '{"name": "lucy", "age": "18"}' | python -mjson.tool
```

bash `jq`

```
$ echo '{"name": "lucy", "age": "18"}' |jq
{
  "name": "lucy",
  "age": "18"
}
$ echo '{"name": "lucy", "age": "18"}' |jq '.name'
"lucy"
```

Gitbook

Gitbook GitHub Pages

Gitbook + GitHub Page Gitbook

Gitbook

Plugins

Gitbook

> `npm install gitbook-plugin-tbfed-pagefooter` `gitbook install`

highlight

lunr

search

sharing

font-settings

"A"

livereload

theme-default

```
{
    "plugins": [
        "theme-default"
    ],
    "pluginsConfig": {
```

```
        "theme-default": {
            "showLevel": true
        }
    }
}
```

search-pro

```
{
    "plugins": [
        "-lunr",
        "-search",
        "search-pro"
    ]
}
```

search-plus

search

```
{
    "plugins": ["-lunr", "-search", "search-plus"]
}
```

ga

Google

> `GitBook doesn't satisfy the requirements of this plugin: >=4.0.0-alpha.0.` `ga@1.0.1`

```
{
    "plugins": [
        "ga"
     ],
    "pluginsConfig": {
        "ga": {
            "token": "UA-XXXX-Y"
        }
    }
}
```

baidu

```
{
```

```
    "plugin": ["baidu"],
    "pluginsConfig": {
        "baidu": {
            "token": "YOUR TOKEN"
        }
    }
}
```

sharing-plus

介绍

```
{
    "plugins": [
        "-sharing",
        "sharing-plus"
    ],
    "pluginsConfig": {
        "sharing": {
            "douban": false,
            "facebook": false,
            "google": true,
            "pocket": false,
            "qq": false,
            "qzone": true,
            "twitter": false,
            "weibo": true,
            "all": [
                "douban",
                "facebook",
                "google",
                "instapaper",
                "linkedin",
                "twitter",
                "weibo",
                "messenger",
                "qq",
                "qzone",
                "viber",
                "whatsapp"
            ]
        }
    }
}
```

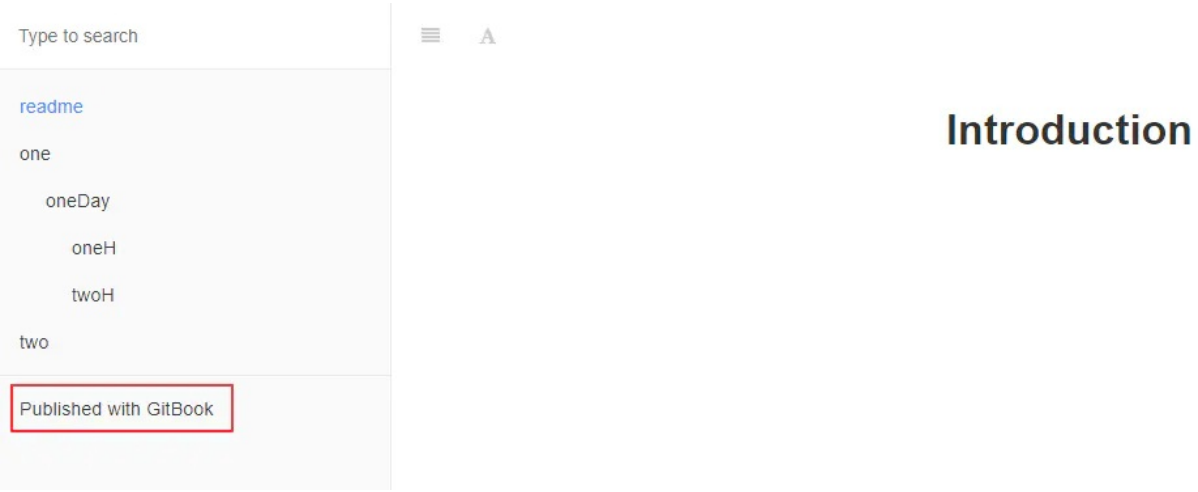
sitemap

sitemap, `/sitemap.xml`

```
{
    "plugins": [
        "sitemap"
    ],
    "sitemap": {
        "hostname": "https://python-book.windard.com/"
    }
}
```

hide-element

> gitbook `Published with GitBook`




```
{
    "plugins": [
        "hide-element"
    ],
    "pluginsConfig": {
        "hide-element": {
            "elements": [".gitbook-link"]
        }
    }
}
```

back-to-top-button


```
{
    "plugins": [
        "back-to-top-button"
    ]
}
```

donate

该文件最后修改时间： 2020-09-23 20:10:44


```
{
    "plugins": [
        "back-to-top-button"
    ],
    "pluginsConfig": {
        "donate": {
            "wechat": "https://xxx.jpg",
            "alipay": "https://xxx.jpg",
            "title": "~ "     ,
            "button": ""  ,
            "alipayText": ""     ,
            "wechatText": ""
        }
    }
}
```

github

GitHub

```
GitBook doesn't satisfy the requirements of this plugin: >=4.0.0-alpha.0.
github@2.0.0
```


```
{
    "plugins": [ "github" ],
    "pluginsConfig": {
        "github": {
            "url": "https://github.com/your/repo"
        }
    }
}
```

github-buttons

GitHub， GitHub GitHub:buttons


```
{
  "plugins": [
    "github-buttons"
  ],
  "pluginsConfig": {
    "github-buttons": {
      "buttons": [{
        "user": "windard",
        "repo": "python-book",
        "type": "star",
        "size": "small"
      }, {
        "user": "windard",
        "type": "follow",
        "width": "160",
        "count": true,
        "size": "small"
      }]
    }
  }
}
```

disqus

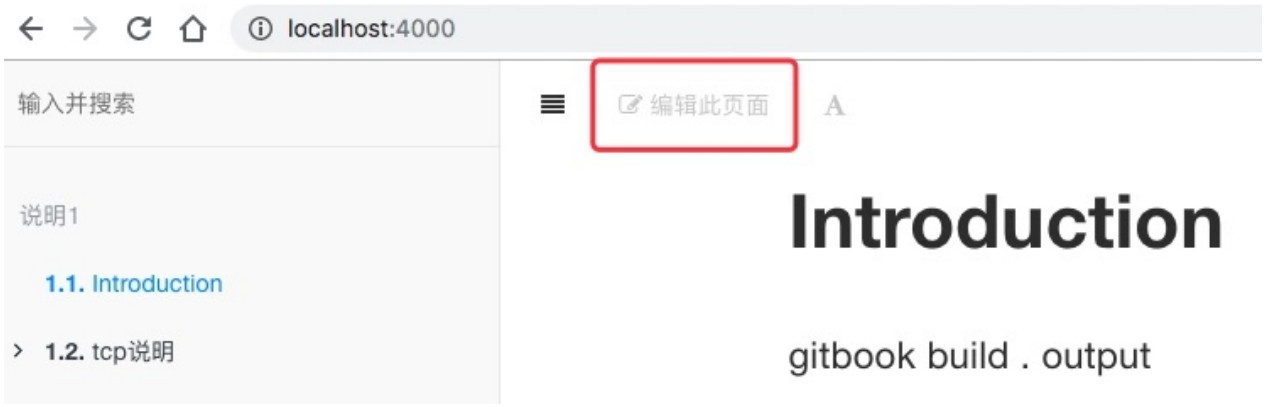
disqus , disqus

> `GitBook doesn't satisfy the requirements of this plugin: >=4.0.0-alpha.0. disqus@0.1.0`

```
{
    "plugins": ["disqus"],
    "pluginsConfig": {
        "disqus": {
            "shortName": "XXXXXXX"
        }
    }
}
```

edit-link

`Edit on GitHub`




```
{
    "plugins": [ "edit-link" ],
    "pluginsConfig": {
        "edit-link": {
            "base": "https://github.com/itswl/gitbook/edit/master",
            "label": "Edit This Page"
        }
    }
}
```

chapter-fold


```
{
    "plugins": ["chapter-fold"]
}
```

expandable-chapters-small

```
{
    "plugins": [
         "expandable-chapters-small"
    ]
}
```

expandable-chapters

code

```
{
    "plugins" : ["code" ]
}
```

```
{
    "plugins" : ["code" ]
}
```

copy-code-button

```
{
    "plugins": ["copy-code-button"]
}
```

```
{
    "plugins" : [ "copy-code-button" ]
}
```

splitter


```
{
    "plugins": [
        "splitter"
    ]
}
```

pageview-count


```
{
  "plugins": ["pageview-count"]
}
```

tbfed-pagefooter


```
{
    "plugins": [
        "tbfed-pagefooter"
    ],
    "pluginsConfig": {
        "tbfed-pagefooter": {
            "copyright": "Copyright &copy 2020",
            "modify_label": ""          ,
            "modify_format": "YYYY-MM-DD HH:mm:ss"
        }
    }
}
```

image-captions

`alt` `title`

```
{
  "plugins": ["image-captions"]
}
```

anchors

GitHub

```
{
  "plugins": ["anchors"]
}
```

popup

```
{
  "plugins": [ "popup" ]
}
```

lightbox

```
{
  "plugins": [ "popup" ]
}
```

custom-favicon

icon icon

> TypeError [ERR_INVALID_ARG_TYPE]: The "path" argument must be of type string. Received undefined icon

```
{
    "plugins" : ["custom-favicon"],
    "pluginsConfig" : {
        "favicon": "path/to/favicon.ico"
    }
}
```

,

- favicon-custom
- custom-favicon-new
- custom-favicon-fix
- custom-favicon-pro

favicon

icon Gitbook icon

```
{
    "plugins": ["favicon"],
    "pluginsConfig": {
        "favicon":{
            "shortcut": "assets/images/favicon.ico",
            "bookmark": "assets/images/favicon.ico",
            "appleTouch": "assets/images/apple-touch-icon.png",
            "appleTouchMore": {
                "120x120": "assets/images/apple-touch-icon-120x120.png",
                "180x180": "assets/images/apple-touch-icon-180x180.png",
            }
        }
```

```
    }
}
```

toc

```
<!-- toc-->
```

re

- 正则表达式基本语法
- 简单使用
- 后向引用
- 高级使用
- 贪婪与懒惰
- 分组与捕获
- 常用正则
- 参考链接

python 的正则表达式库，现在几乎所有的编程语言都支持正则表达式了，无可否认，正则表达式确实强大。

```
{
  "plugins": [ "toc" ]
}
```

atoc

```
{
  "plugins": [ "atoc" ]
}
```

page-toc

```
---
showToc: true
---
```

logging

标准的输出日志库，比每次用 print 输出不知道高到哪里去了。

Table of Contents

常用配置

```
{
  "plugins": [ "page-toc" ]
}
```

autocover

```
{
    "title": "Windard's Python Book",
    "description": "Python "        ,
    "author": "Windard",
    "plugins": ["autocover"],
    "pluginsConfig": {
        "autocover": {
            "font": {
                "size": null,
                "family": "Impact",
                "color": "#FFF"
            },
            "size": {
                "w": 1800,
                "h": 2360
            },
            "background": {
                "color": "#09F"
            }
        }
    }
}
```

svg

```
error: error while generating page "README.md":
Error: Error converting /tmp/tmp-281LPVxSm5Bv6U1/842cb21f.svg into /tmp/tmp-281LPVxSm5Bv6U1/302430b8.png
```

`svgexport` `puppeteer`

```
npm i --unsafe-perm -g svgexport@0.3.2
```

```
      if (cb) cb.apply(this, arguments)
                 ^
TypeError: cb.apply is not a function
```

node `v12.18.4` `v14.12.0` , `v12.18.1` `v13.14.0`