

python的日志模块

```
1 logging模块
2
3 1. 日志级别:
4 调试信息->正常信息->警告信息->错误信息->严重错误
5 logging默认打印警告信息级别以上
6
7 import logging
8
9 logging.debug("调试信息")
10 logging.info("正常信息")
11 #只有下面三个有打印
12 logging.warning("警告信息")
13 logging.error("错误信息")
14 logging.critical("严重错误信息")
15
16 2. 存储日志信息
17 import logging
18
19 logging.basicConfig(filename="1.txt")#追加方式存储
20 logging.debug("调试信息")
21 logging.info("正常信息")
22 #只有下面三个有打印
23 logging.warning("警告信息")
24 logging.error("错误信息")
25 logging.critical("严重错误信息")
26
27 #日志存储案例
28 logging.basicConfig(filename="2.txt")
29 def mytest(a,b):
30     if b==0:
31         logging.critical("除数不能0")
32     return a/b
33
34 mytest(10,0)
35
36
37 3. logging的配置
38 #表示存储到xx.txt的日志级别最低是debug级别
39 logging.basicConfig(filename="xx.txt", level=logging.DEBUG)
40
41 #表示存储到xx.txt的加上时间和用户输出的信息
42 logging.basicConfig(filename="xx.txt", format='%(asctime)s %(message)s')
43
44
45 4. logging高级使用
46 介绍4个东东,记录器,处理器,过滤器,格式化器
47 logger:记录器,应用程序代码能直接使用的接口。
48 handler:处理器,将(记录器产生的)日志记录发送到合适的目的地
```

```
49 filter:过滤器,提供了更好的粒度控制,可以决定输出哪些日志记录
50 formatter:格式化器,指明了最终输出中日志记录的布局。
51
52 #!/usr/bin/env python
53 # -*- coding:utf-8 -*-
54 #====#====#====#====
55 #Author:
56 #CreateDate:
57 #Version:
58 #====#====#====#====
59 #调试信息->正常信息->警告信息->错误信息->严重错误
60 import logging
61
62 #1.创建记录器,并给记录器取名字
63 logger=logging.getLogger("maker")
64 #设置日志级别
65 logger.setLevel(logging.INFO)#2
66
67 #2.创建2个处理器,一个把日志发送到终端,一个把日志发送到文件中
68 ch=logging.StreamHandler()#发送到终端
69 ch.setLevel(logging.DEBUG)#1
70
71 fh=logging.FileHandler(filename="666.txt")
72 fh.setLevel(logging.WARNING)#3
73
74 #创建过滤器,过滤含abc的字符串
75 class stringFilter(logging.Filter):
76     def filter(self,record):
77         if record.msg.find("abc")==-1:
78             return True
79         return False
80
81 #4.创建格式化器
82 form=logging.Formatter("%(asctime)s -- %(name)s -- %(message)s")
83
84 #关联
85 #格式化器放到处理器里
86 ch.setFormatter(form)
87 fh.setFormatter(form)
88
89 #过滤器放到ch处理器中
90 ch.addFilter(stringFilter())
91
92 #处理器放到记录器中
93 logger.addHandler(ch)
94 logger.addHandler(fh)
95
96
97 logger.debug("调试信息")
98 logger.info("正常信息")
99 logger.warning("警告信息")
100 logger.error("错误")
101 logger.critical("严重错误")
```

```
102 logger.warning("abcdef")
103
104 设置的日志级别以高为准
105
106
107 #!/usr/bin/env python
108 # -*- coding:utf-8 -*-
109 #====#====#====#====
110 #Author:
111 #CreatDate:
112 #Version:
113 #====#====#====#====
114
115 from selenium import webdriver
116 import unittest
117 import time
118 from mylog import logger
119 import traceback
120
121 class Maker(unittest.TestCase):
122
123     def setUp(self):
124         self.dr=webdriver.Firefox()
125
126     def test_01(self):
127         self.dr.get("https://www.baidu.com")
128         time.sleep(2)
129         try:
130             e=self.dr.find_element_by_link_text("新")
131         except Exception as e:
132             # mystr1=f'出错的文件是:
{e.__traceback__.tb_frame.f_globals["__file__"]}'
133             # mystr2=f'第:{e.__traceback__.tb_lineno} 行'
134             # mystr=mystr1+mystr2
135             # print(mystr)
136
137             logger.critical("严重错误,没有该元素")
138             logger.error(traceback.format_exc())
139         finally:
140             self.dr.quit()
141
142         e.click()
143
144
145     def tearDown(self):
146         time.sleep(3)
147         self.dr.quit()
148
149 if __name__=='__main__':
150     unittest.main()
```