1 模块级别函数

1.1 logging.getLogger(name=None)

返回具有指定 name 的日志记录器，或者当 name 为 None 时返回层级结构中作为根日志记录器的日志记录器。附带给定 name 的所有对此函数的调用都将返回相同的日志记录器实例。默认生成的 root logger 的 level 是 logging.WARNING, 低于该级别的就不输出了。

创建一个 logging

```python
import logging

l_g = logging.getLogger()

# 要显示debug和info加上
# l_g.setLevel(logging.DEBUG)
# s_h = logging.StreamHandler(sys.stderr)
# l_g.addHandler(s_h)

l_g.debug('debug msg')
l_g.info('info msg')
l_g.error('error msg')
l_g.warning('warn msg')

#OUTPUT:
#       error msg
#       warn msg
```

日志输入文件

```python
import logging  # 引入logging模块
import os.path
import time
# 第一步，创建一个logger
logger = logging.getLogger()
logger.setLevel(logging.INFO)  # Log等级总开关
# 第二步，创建一个handler，用于写入日志文件
rq = time.strftime('%Y%m%d%H%M', time.localtime(time.time()))
log_path = os.path.dirname(os.getcwd()) + '/rpmnet1/'
log_name = log_path + rq + '.log'
logfile = log_name
fh = logging.FileHandler(logfile, mode='w')
fh.setLevel(logging.DEBUG)  # 输出到file的log等级的开关
# 第三步，定义handler的输出格式
formatter = logging.Formatter("%(asctime)s - %(levelname)s: %(message)s"
    )
```

```
fh.setFormatter(formatter)
formatter1 = logging.Formatter("%(asctime)s - %(filename)s[line:%(lineno)d] - %(levelname)s: %(message)s")
fh.setFormatter(formatter1)
# 第四步，将logger添加到handler里面
logger.addHandler(fh)
# 日志
logger.debug('this is a logger debug message')
logger.info('this is a logger info message')
logger.warning('this is a logger warning message')
logger.error('this is a logger error message')
logger.critical('this is a logger critical message')
```

1.2 logging.FileHandler 和 logging.Formatter

logging.basicConfig

```
import coloredlogs, logging
import os

# Create a logger object.
logger = logging.getLogger()
log_path ='D:/python_code/RPMNet-master/RPMNet-master/rpmnet1'

os.makedirs(log_path, exist_ok=True)
coloredlogs.install(level='INFO', logger=logger)
file_handler = logging.FileHandler('{}/temp.txt'.format(log_path))
log_formatter = logging.Formatter('%(asctime)s [%(levelname)s] %(pathname)s %(filename)s %(funcName)s %(lineno)d %(process)d %(processName)s %(thread)d %(threadName)s - %(message)s')
#%(levelno)s  打印日志级别的数值
#%(levelname)s  打印日志级别名称
#%(pathname)s  打印当前执行程序的路径
#%(filename)s  打印当前执行程序名称
#%(funcName)s  打印日志的当前函数
#%(lineno)s  打印日志的当前行号
#%(asctime)s  打印日志的时间
#%(thread)d  打印线程id
#%(threadName)s  打印线程名称
#%(process)d  打印进程ID
#%(processName)s  打印进程名
#%(message)s  打印日志信息
file_handler.setFormatter(log_formatter)
logger.addHandler(file_handler)
logger.info('Output and logs will be saved to {}'.format(log_path))
```