

zipfile

--zipfile.is_zipfile(filename)

判断是不是zip

--class f = zipfile.ZipFile(fileName[, mode[, compression[, allowZip64]]])

mode: 'w' 覆盖式写入, 'r' 读取内容模式, 'a' 追加内容

compression: ZIP_STORE; ZIP_DEFLATED. ZIP_STORE是默认的, 表示不压缩; ZIP_DEFLATED表示压缩

allowZip64: 为True时, 表示支持64位的压缩, 一般而言, 在所压缩的文件大于2G时, 会用到这个选项; 默认情况下, 该值为False, 因为Unix系统不支持。

--f.close()

关闭

--f.write(filename)

要是这里加入路径, 那么就会在zip中自动创建对应文件夹

所以可以先os.chdir()到当前路径, 再添加文件

--f.extract(member, path=None, pwd=None)

--f.extractall(path=None, members=None, pwd=None)

都是'r'模式打开

pwd是bytes-like对象

--f.setpassword(password)

bytes-like对象

--f.namelist()

--f.printdir()

打印信息

FE:

```
>>> f = zipfile.ZipFile(r'C:\Users\lenovo\Desktop\cracker\ZIP.zip', 'r')
>>> f.printdir()
File Name                               Modified                               Size
Users/lenovo/Desktop/cracker/for12.txt  2017-02-08 10:52:04                   74
for12.txt                               2017-02-08 10:52:04                   74
>>> f.namelist()
['Users/lenovo/Desktop/cracker/for12.txt', 'for12.txt']
>>> f.setpassword('123456')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "D:\编程\lib\zipfile.py", line 1206, in setpassword
    raise TypeError("pwd: expected bytes, got %s" % type(pwd))
TypeError: pwd: expected bytes, got <class 'str'>
>>> f.setpassword('123456'.encode('utf8'))
>>> f.close()
>>>
```

--f.open(filename[, mode][, pwd])

--F = f.getinfo(filename)

返回ZipInfo对象

-- ZipInfo.filename

-- ZipInfo.date_time

返回值的格式为(year, month, date, hour, minute, second)

-- ZipInfo.compress_type

--ZipInfo.comment

--ZipInfo.extra

--ZipInfo.create_system

--ZipInfo.extract_version

--ZipInfo.reserved

总是0

--ZipInfo.flag_bits

--ZipInfo.volume

--ZipInfo.internal_attr

--ZipInfo.external_attr

--ZipInfo.header_offset

```
--ZipInfo.CRC  
--ZipInfo.file_size  
--ZipInfo.compress_size  
--ZipFile.testzip()
```

检查每个文件和它对应的CRC，如果有错误返回对应的文件列表