文件读取：

例：程序file\_test/file\_test1.py

# 读取整个文件  
# 通过with访问文件不需要手动关闭  
with open('pi.txt') as file\_object:  
 contents = file\_object.read()  
 print(contents)  
  
print()  
  
# 读取二进制文件  
with open('画板.png', 'rb') as file\_binary:  
 print(file\_binary.read())  
  
  
# 以指定编码方式读取文件  
with open('..\gbk.txt', 'r', encoding='gbk') as file\_gbk:  
 print(file\_gbk.read())  
print()  
  
# 逐行读取  
# 打印结果每行之间有空白行  
# 这是打印换行符造成的  
# 可以使用rstrip()清除换行符  
with open('pi.txt') as file\_line:  
 for line in file\_line:  
 print(line.rstrip())  
print()  
  
# 按行读取这个文件  
with open('pi.txt') as file\_lines:  
 lines = file\_lines.readlines()  
  
for content in lines:  
 print(contents.rstrip())

文件写入：

例：程序file\_test/file\_test2.py

# 类似C语言中的文件写入  
# 如果不存在会创建文件  
# 会覆盖原来的内容  
with open('测试.txt', 'w') as file\_object:  
 file\_object.write("Python文件测试\n")  
 file\_object.write("Python学习\n")  
print()  
  
# 附加模式写入  
with open('测试.txt', 'a') as file\_add:  
 file\_add.write("Python开发工具\n")

文件解析：

例：程序file\_test/file\_test3.py

import pickle  
  
  
with open('datafile.txt') as file\_obj:  
 line = file\_obj.readline()  
 print(line)  
 parts = line.split('$')  
 # eval()将字符串转换为python对象  
 objects = [eval(part) for part in parts]  
 print(objects)  
print()  
  
# 存储python的原生对象  
data = {'a': 1, 'b': 2}  
with open('pythondata.pkl', 'wb') as file\_save:  
 pickle.dump(data, file\_save)  
  
  
# 读取  
with open('pythondata.pkl', 'rb') as file\_read:  
 e = pickle.load(file\_read)  
 print(e)