**Прочитать все строки с клавиатуры**

import sys

for line in sys.stdin:

print(line)

**Прочитать все строки из файла**

Пусть файл называется my.txt. Надо прочитать (и напечатать) из него все строки.

with open('my.txt', 'r') as f:

for line in f:

print(line)

**Открыть файл**

**open** (file, mode='r', buffering=None, encoding=None, errors=None, newline=None, closefd=True)

|  |  |
| --- | --- |
| [**Как открыть**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonFile?sortcol=0&table=1&up=0#sorted_table) |  |
| 'r' | открытие на чтение (является значением по умолчанию). |
| 'w' | открытие на запись, содержимое файла удаляется, если файла не существует, создается новый. |
| 'x' | открытие на запись, если файла не существует, иначе исключение. |
| 'a' | открытие на запись, информация добавляется в конец файла. |
| 'b' | открытие в двоичном режиме. |
| 't' | открытие в текстовом режиме (является значением по умолчанию). |
| '+' | открытие на чтение и запись |

**Закрыть файл**

*переменная*. **close** ()

**Читать из файла**

**Прочитать все**

* **read()** - прочитать весь файл
* **read(n)** - прочитать n байт из файла.

f = open('tr.log', 'r') # открыть файл tr.log на чтение

str = f.read() # прочитать весь файл в переменную str

print str # напечатать переменную str

f.close() # закрыть файл

**Прочитать сначала первую строку, а потом остальные**

* **readline()** - прочитать строку из файла

# -\*- coding: utf-8 -\*-

f = open('tr.log', 'r')

line = f.readline() # отдельно читаем первую строку

while line:

print line

line = f.readline() # читаем все остальные строки

f.close()

**Прочитать по строкам**

# -\*- coding: utf-8 -\*-

f = open('tr.log', 'r')

for line in f:

print line

f.close()

**Писать в файл**

**python 2.7**

# -\*- coding: utf-8 -\*-

f = open('output.dat', 'w')

print >> f, 'Hello, world!'

f.close()

**python 3.x**

# -\*- coding: utf-8 -\*-

with open('output.dat', 'w') as f:

print ('Hello, world!', file=f)

# f.close() сделает автоматически

**Задачи**

**Задача 1. Файлы входных и выходных данных**

Для проверяющей системы Ejudge тесты должны быть записаны в виде файлов входных данных 001.dat, 002.dat, 003.dat и так далее. И в виде эталона выходных данных по этим входным данным. Эталонные выходные данные пишутся в файлы 001.ans, 002.ans и т.п.

Запишем входные и выходные данные всех тестов в один файл, по которому будем генерировать эти файлы. В файл пишем данные в формате

входные данные теста 1

---

выходные данные теста 1

===

входные данные теста 2

---

выходные данные теста 2

===

входные данные теста 3

---

выходные данные теста 3

===

Написать программу, которая в аргументах командной строки получает общий тестовый файл и в той же директории, где лежит этот общий тестовый файл делает директорию tests с файлами 001.dat, 001.ans, 002.dat, 002.ans и так далее.

python gentest.py ../task1/alltest.txt

Ожидаемая структура директории task1:

task1/

+--- alltest.txt

+--- tests

+---- 001.dat

+---- 001.ans

+---- 002.dat

+---- 002.ans

+---- 003.dat

+---- 004.ans

**Задача 2. Генератор паролей**

Генерируем пароль, состоящий из случайных:

* буковок,
* буковок + циферок,
* буковок + циферок + буковок в верхнем регистре,
* буковок + циферок + буковок в верхнем регистре + цифирьки в верхнем регистре
* заданной строки

В консоли задаётся аргументами:

* Длина пароля (по умолчанию 8)
* Сложность (от 1 до 4) или алфавит пароля

Доп.задание: Использовать только английские буковки, или же английские + русские

Пароль выводим в консоль.

**Задача 3. Обзорщик жёсткого диска:**

Программа показывает следующую информацию:

* Сколько файлов какого типа (по расширению) есть в указанной директории? Сколько папок?
* Сколько места занимают файлы каждого типа?
* (Дополнительно++) Какие файлы и папки самые большие по размеру (папки — вместе с вложенными файлами)?

Аргументами командной строки задаётся:

* путь к директории, с которой начнём поиски.
* какие запросы делать (количество файлов разного типа, размер файлов разного типа)

Максимальный размер таблицы (по умолчанию — только 10 самых часто встречающихся типов файлов).

NB: Таблички с результатами выводить с помощью модуля PTable или Tabulate

-- [TatyanaDerbysheva](http://acm.mipt.ru/twiki/bin/view/Main/TatyanaDerbysheva) - 15 Dec 2015