

Работа с файлами в Python

Функция `open` принимает два параметра: имя файла (или путь к файлу, если файл находится не в том каталоге, где записана программа) и режим открытия файла:

'r' – открыть на чтение,

'w' – открыть на запись,

'a' – открыть на добавление.

Метод `close`, определённый для файлового указателя, закрывает файл:

```
F = open ('input.txt')
```

```
O = open ( 'output.txt', 'w' )
```

```
# здесь работаем с файлами
```

```
F.close()
```

```
O.close()
```

При первом вызове функции `open` режим работы с файлом не указан, но по умолчанию предполагается режим 'r' (чтение).

Чтение одной строки из текстового файла выполняет метод `readline`, связанный с файловой переменной:

```
s = F.readline()
```

Если нужно прочитать несколько данных в одной строке, разделённых пробелами, используют метод `split`. Этот метод разбивает строку по пробелам и строит список из соответствующих 'слов':

```
s = F.readline().split()
```

Если в прочитанной строке файла были записаны числа 1 и 2, список `s` будет выглядеть так: ['1', '2']

Элементы этого списка – символьные строки. Поэтому для того, чтобы выполнять с этими данными вычисления, их нужно перевести в числовой вид с помощью функции `int`, применив её для каждого элемента списка:

```
a, b = int(s[0]), int(s[1])
```

То же самое можно записать с помощью генератора

```
a, b = [int(x) for x in s]
```

или с помощью функции `map`, которая применяет функцию `int` ко всем элементам списка:

```
a, b = map( int, s )
```

Запись `map(int,s)` означает 'применить функцию `int` ко всем элементам списка `s`'.

Проще все сразу одной строкой

```
a, b = map( int, F.readline().split())
```

Неизвестное количество данных

Предположим, что в текстовом файле записано в столбик неизвестное количество чисел и требуется найти их сумму. В этой задаче не нужно одновременно хранить все числа в памяти, достаточно читать по одному числу и сразу его обрабатывать. Для того, чтобы определить, когда данные закончились, будем использовать особенность метода `readline`: когда файловый курсор указывает на конец файла, метод `readline` возвращает пустую строку, которая воспринимается как ложное логическое значение:

```
while True:
```

```
    s = F.readline()
```

```
    if not s: break
```

В этом примере при получении пустой строки цикл чтения заканчивается с помощью оператора **break**. Возможны и другие варианты.

Например, метод `readlines` позволяет прочитать все строки сразу в список:

```
F = open ( 'input.txt' )
list = F.readlines()
for s in list:
    print ( s, end = " )
F.close()
```

Строки файла читаются в список `lst` и выводятся на экран. Обратите внимание, что переход на новую строку в функции `print` отключен (`end=""`), потому что при чтении символы перевода строки `'\n'` в конце строк файла сохраняются.

В Python есть ещё один способ работы с файлами, при котором закрывать файл не нужно, он закроется автоматически. Это конструкция **with-as**:

```
with open ( 'input.txt' ) as F:
    for s in F:
        print ( s, end = " )
```

В первой строке файл `input.txt` открывается в режиме чтения и связывается с файловым указателем `F`. Затем в цикле перебираются все строки в этом файле, каждая из них по очереди попадает в переменную `s` и выводится на экран. Закрывать файл с помощью `close` не нужно, он закроется автоматически после окончания цикла.

Наконец, приведём ещё один способ в стиле Python:

```
for s in open ( 'input.txt' ):
    print ( s, end = " )
```

Этот вариант обычно рекомендуют к использованию, при нём также не нужно закрывать файл.

Примеры задач

1. Считать два числа из одной строки и вывести их сумму в файл	2. Считать три числа из одной строки и записать в файл большее из них
<pre>f=open('1_in.txt') o=open('1_out.txt', 'w') a,b = map(int, f.readline().split()) o.write(str(a+b)) f.close() o.close()</pre>	<pre>f=open('2_in.txt') o=open('2_out.txt', 'w') a,b,c = map(int, f.readline().split()) o.write(str(max(a,b,c))) f.close() o.close()</pre>
3. В первом файле 3in.txt есть строки, содержащие по два числа в каждой. Создать файл 3out.txt где будут просуммированы все числа построчно. Пример вывода: 5+8=13	
<pre>f=open('3in.txt') o=open('3out.txt','w') while True: l=f.readline() if not l: break a,b = l.split() s=a+'+'+b+'=' +str(int(a)+int(b)) o.write(s+'\n') f.close() o.close()</pre>	<p style="text-align: center;"><i>2 вариант</i></p> <pre>o=open('3out.txt','w') for s in open ('3in.txt'): a,b=s.split() o.write(a+'+'+b+'=' +str(int(a)+int(b))+'\n') o.close()</pre>

8. Считать из файла строку и поместить в массив посимвольно без пробелов

```
f=open('8in.txt','r', encoding='utf-8')
line = f.readline()
a=[]
for x in line:
    if x!=' ':
        a.append(x)
f.close
print(a)
```

9. Считать из файла строку и поместить в массив по словам без пробелов

<pre>for x in open('9in.txt','r', encoding='utf-8'): a=x.split() print(a) Можно удалить знаки препинания →</pre>	<pre>for i in range(len(a)): if not a[i].isalnum(): a[i]=a[i][:len(a[i])-1] print(a)</pre>
--	--

10. Считать из произвольного файла текст и поместить в массив по словам без пробелов

```
s=""
for x in open('16in.txt','r', encoding='utf-8'):
    s+=x.strip()+' '
print(s)
a=s.split()
for i in range(len(a)):
    if not a[i].isalnum():
        a[i]=a[i][:len(a[i])-1]
print(a)
```

11. Напишите программу, записывающую в текстовый файл random.txt 25 случайных чисел в диапазоне от 111 до 777 (включительно), каждое с новой строки.

```
import random
with open('11out.txt', 'w') as f:
    for i in range(25):
        x = random.randint(111, 777)
        f.write(str(x) + '\n')
```

12. Вам доступен текстовый файл 12in.txt, состоящий из нескольких строк. Напишите программу для записи содержимого этого файла в файл 12out.txt в виде нумерованного списка, где перед каждой строкой стоит ее номер, символ) и пробел. Нумерация строк должна начинаться с 1.

```
o=open('12out.txt','w')
k=1
for s in open('12in.txt'):
    o.write(str(k)+' '+s)
    k+=1
o.close()
```

13. Вам доступны два текстовых файла 13names.txt и 13fam.txt, один с именами, другой с фамилиями. Напишите программу, которая с помощью модуля random создает 3 случайные пары имя + фамилия, а затем выводит их, каждую на отдельной строке.

```
import random
```

```
with open('13name.txt', 'r', encoding='utf-8') as n, open('13fam.txt', 'r', encoding='utf-8') as f:
    z, x = n.readlines(), f.readlines()
    for i in range(5):
        print(random.choice(z).strip(), random.choice(x).strip(), sep=' ')
```

14. На вход программе подается натуральное число n и n строк с названиями файлов. Напишите программу, которая создает файл `output.txt` и выводит в него содержимое всех файлов с указанными именами, не меняя их порядка.

```
put=open('file_out.txt', 'w')
f=open('file_in.txt')
n=int(f.readline())
for i in range(n):
    x=open(f.readline().rstrip() + '.txt')
    put.write(x.read()+'\n')
f.close()
put.close()
```

15. Вам доступен текстовый файл `population.txt` с названиями стран и численностью их населения, разделенными символом табуляции `'\t'`. Напишите программу, выводящую все страны, название которых начинается с буквы `'К'` или `'С'`, численность населения которых больше чем 500 000 человек, не меняя их порядок.

```
for s in open('country.txt', 'r', encoding='utf-8'):
    n, p = s.split('\t')
    if n.startswith('К') or n.startswith('С') and int(p) > 500000:
        print(n,p)
```

16. Дан файл **16in.txt** с текстом. Определить сколько в нем каждой из гласных букв

```
s=""
a=[0]*20
ga='АаЁёЕеИиОоУуЫыЭэЮюЯя'
for x in open('16in.txt','r', encoding='utf-8'):
    s+=x
for x in ga:
    a[ga.find(x)]= s.count(x)
    print(x,'-',s.count(x))
print(a)
```

```
ga='АаЁёЕеИиОоУуЫыЭэЮюЯя'
s=""
for x in open('16in.txt','r', encoding='utf-8'):
    s+=x
for x in ga:
    print(x,'-',s.count(x))
```

17. Дан файл с текстом. В тексте есть слова, числа, знаки препинания и перенос по слогам. Найти самое распространенное слово.

```
s=""
for x in open('17in.txt','r', encoding='utf-8'):
    s+=x.strip()+' '
s=s.replace(' ','')
print(s)
a=s.split()
for i in range(len(a)):
    if not a[i].isalnum():
        a[i]=a[i][:len(a[i])-1]
print(a)
```

```
max=0
for x in a:
    if a.count(x)>max:
        max=a.count(x)
        r=x
print(r,max)
```

18. Дан файл с текстом. В тексте есть слова, числа, знаки препинания и перенос по слогам. Вывести все слова по рейтингу распространенности

```
s=""
for x in open('17in.txt','r', encoding='utf-8'):
    s+=x.strip()+ ' '
s=s.replace('- ','')
print(s)
a=s.split()
for i in range(len(a)):
    if not a[i].isalnum():
        a[i]=a[i][:len(a[i])-1]
print(a)
while a!=[]:
    max=0
    for x in a:
        if a.count(x)>max:
            max=a.count(x)
            r=x
    print(r,max)
    while r in a:
        a.remove(r)
```