

Файловый ввод-вывод в Python

Для открытия файла в Python, как на чтение, так и на запись, используется встроенная функция `open`, которая возвращает открытый файл. Например:

```
file_to_read = open('task.in', 'r') # Открытие файла на чтение
file_to_write = open('task.out', 'w') # Открытие файла на запись
```

Первый аргумент функции `open` — имя файла, который нужно открыть. Вторым аргументом — режим открытия файла ('r' — для чтения, 'w' — для записи).

Теперь переменные `file_to_read` и `file_to_write` можно использовать для чтения из файла и для записи в файл соответственно:

```
line = file_to_read.readline() # Читает и возвращает очередную строку из
                                # файла. Аналогична функции input(), но в
                                # конце каждой строки ставит символ
                                # перевода строки '\n' (это один символ,
                                # который имеет составное обозначение).
```

```
file_content = file_to_read.read() # Читает весь файл и возвращает
                                    # его содержимое как строку.
                                    # Например, при следующем файле:
                                    # 1
                                    # abc
                                    # 5
                                    # результат будет: '1\nabc\n5\n'.
```

```
lines = file_to_read.readlines() # Читает весь файл и возвращает
                                   # его содержимое, как список строк.
```

```
file_to_write.write('any string') # В качестве единственного аргумента
                                    # принимает строку и печатает её в файл
                                    # без последующего перевода строки.
```

```
print(1, '345', file=file_to_write) # Функция print имеет именованный
                                       # аргумент file, позволяющий выводить
                                       # данные в файл вместо экрана.
```

```
file_to_read.close() # Функция close используется для закрытия файла
file_to_write.close()
```

В некоторых случаях удобно использовать следующий способ чтения файла:

```
for line in file_to_read.readlines(): # Цикл по всем строкам файла
    print(line, file=file_to_write)
```

Этот пример выведет в файл `file_to_write` содержимое файла `file_to_read`. Вместо второй строки (с функцией `print`) вы можете написать любой свой код.

Вот как можно считать массив чисел из первой строки файла:

```
fin = open('input.txt', 'r')
numbers = [int(i) for i in fin.readline().split()]
```

А вот как считать два числа n и m из первой строки файла:

```
fin = open('input.txt', 'r')
n, m = [int(i) for i in fin.readline().split()]
```

Чтобы вывести массив чисел в строчку, можно использовать `join`. Но `join` принимает на вход последовательность строк, поэтому числа нужно сначала преобразовать в строки с помощью генератора:

```
fout = open('output.txt', 'w')
numbers = [42, 666, 100500]
print(' '.join([str(i) for i in numbers]), file = fout)
fout.close()
```

Если в последнем примере не закрыть файл `fout`, то данные могут не записаться в файл (а могут и записаться).