



# Introduction to Using Files in Python

# About me



## Trayan Iliev

- CEO of IPT – Intellectual Products & Technologies  
<http://www.iproduct.org>
- Oracle® certified programmer 15+ Y
- end-to-end reactive fullstack apps with [Java](#), [ES6+](#), [TypeScript](#), [Angular](#), [React](#) and [Vue.js](#)
- 12+ years IT trainer: [Spring](#), [Java EE](#), [Node.js](#), [Express](#), [GraphQL](#), [SOA](#), [REST](#), [DDD](#) & [Reactive Microservices](#)
- Voxxed Days, jPrime, Java2Days, jProfessionals, BGOUG, BGJUG, DEV.BG speaker
- Organizer RoboLearn hackathons and IoT enthusiast

# Where to Find The Code and Materials?

<https://github.com/iproduct/intro-python>

# Files in Python

- Използват се за постоянно съхраняване на данни
- В най-простия си вид въвеждането и извеждането на информация във файл наподобява това със стандартния вход (клавиатура) и изход (екран)
- Входно-изходните операции с файлове са много повече на брой, с много по-големи възможности и позволяват всякаква обработка на информацията съхранявана във файловете
- Стандартната команда за свързване на обект от тип данни файл с конкретен физически файл:

```
afile = open(<filename>,< mode>)
```

# Files in Python

```
<file_obj_name> = open(<filename>,< mode>)
```

- `open()` е функция, която отваря физическия файл за вход/изход, създава входно-изходен буфер в паметта и връща като резултат адреса на този буфер
- `<file_obj_name>` е променлива, която задава името на обект от тип файл, който се свързва с току-що създадения входно-изходен буфер, т.е. тази променлива съдържа като стойност адреса на буфера на файла

# The Full Picture

```
afile = open(file, mode='r', buffering=- 1, encoding=None, errors=None,
newline=None, closefd=True, opener=None)
```

- *mode* – optional string that specifies the mode in which the file is opened. It defaults to `'r'` which means open for reading in text mode. Other common values are `'w'` for writing (truncating the file if it already exists), `'x'` for exclusive creation, and `'a'` for appending.
- *buffering* – optional integer used to set the buffering policy. Pass 0 to switch buffering off (only allowed in binary mode), 1 to select line buffering (only usable in text mode), and an integer  $> 1$  to indicate the size in bytes of a fixed-size chunk buffer
- *encoding* is the name of the encoding used to decode or encode the file. This should only be used in text mode. The default encoding is platform dependent (whatever [`locale.getpreferredencoding\(\)`](#) returns)
- *errors* is an optional string that specifies how encoding and decoding errors are to be handled—this cannot be used in binary mode. A variety of standard error handlers are available (listed under [Error Handlers](#)), though any error handling name that has been registered with [`codecs.register\_error\(\)`](#) is also valid.

# Files Open Modes

Character	Meaning
'r'	open for reading (default)
'w'	open for writing, truncating the file first
'x'	open for exclusive creation, failing if the file already exists
'a'	open for writing, appending to the end of file if it exists
'b'	binary mode
't'	text mode (default)
'+'	open for updating (reading and writing)

# Binary vs. Text

Когато файла е отворен като двоичен, четене и записване става чрез обект данни `bytes`

Когато файла е отворен като текстов, четене и записване става чрез обект данни `str`, като първо се прави декодиране на съдържанието (четене) или кодиране (записване)

'b' - работа с двоичен (не текстов) файл

'+' - използване на файла за писане и четене (обновяване)

'x' - създаване и извеждане във файл (писане, `write`) – ако съществува връща грешка



# Other Arguments of open()

**buffering** – размер на буфера за четене и писане

**encoding** – метод на кодиране (само за текстов файл)

**errors** – метод за обработка на грешките

**newline** – как да се интерпретира край на ред (само за текстов файл)  
– има различия в различни ОС

**closefd** – указва как и кога да се затваря физически файла

**opener** – за използване на алтернативна програма за отваряне на файла

# File Methods

`aString = input.read()` # чете цял файл в низ

`aString = input.read(N)` # чете следващите N символа  
# (или байта) в низ

`aString = input.readline()` # чете един ред в низ

`aList = input.readlines()` # чете цял файл в списък низове  
# по редове

`output.write(aString)` # записва низ от символи (или  
# байтове) във файл

`output.writelines(aList)` # записва списък от низове  
# (редове) в цял файл

# File Methods - II

<code>aString = input.read()</code>	<code># чете цял файл в низ</code>
<code>output.close()</code>	<code># Затваряне на файл (след # всички записи)</code>
<code>output.flush()</code>	<code># Запис от буфера във файла без # затваряне</code>
<code>anyFile.seek(N)</code>	<code># Позиционира файл до позиция # N за следващо действие</code>
<code>for line in open('data'):</code>	<code># използва line за итеративно # четене по редове от файл</code>

# Examples

Преброяване на броя редове в един файл:

```
file1 = open("text.txt", "r")
```

```
line_count = 0
```

```
for line_count in file1:
```

```
    line_count += 1
```

```
print('Брой на редове във файла: ', line_count)
```

Прочитане на цял файл:

```
File2 = open("text2.txt")
```

```
Batch = File2.read()
```

```
print(len(Batch))
```

# Examples - II

```
>>> lst= [ x for x in open("text.txt", "r").readlines() ]
```

```
>>> lst
```

```
['Chen Lin\n', 'clin@brandeis.edu\n', 'Volen 110\n', 'Office Hour: Thurs. 3-5\n', '\n', 'Yaqin Yang\n', 'yaqin@brandeis.edu\n', 'Volen 110\n', 'Office Hour: Tues. 3-5\n']
```

Игнориране на заглавен ред

```
for (i,line) in enumerate(open('text.txt','r').readlines()):  
    if i == 0: continue  
    print line
```

# Word Counting

```
>>> for line in open('names.txt'):
...     name = line.strip()
...     name_count[name] = name_count.get(name,0)+ 1
...
>>> for (name, count) in name_count.items():
...     print name, count
...
Chen 3
Ben 3
Yaqin 3
```

# Text File Copy Line by Line

```
input_file = open("in.txt")  
output_file = open("out.txt", "w")  
for line in input_file:  
    output_file.write(line)
```

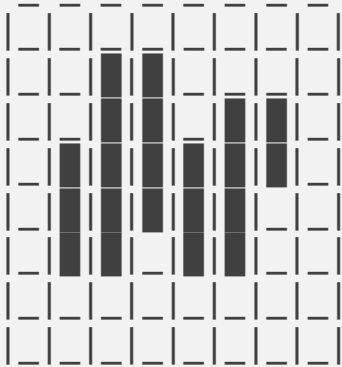
# Problem: Word Counting

Реализирайте конзолно приложение което по подадено като аргумент от команден ред име на файл извлича top 20 ключови думи за файла на база на честотата на тяхното срещане.

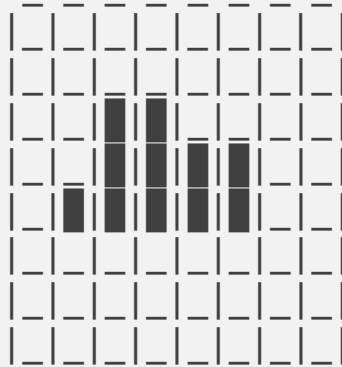


# Problem: Melting Iceberg

Айсберг има форма, която може да се изобрази в таблица с  $N$  реда и  $N$  стълба,  $7 < N < 200$ , например айсбергът от фиг. 1 след един час в резултат на топенето се превръща в айсберга от фиг. 2:



фиг. 1



фиг. 2

Клетките от първия и последния ред и стълб са винаги празни. Външните клетки, които са изложени на съприкосновение с топлия въздух и вода се топят, а вътрешните не. Айсбергът се топи по следното правило: всяка клетка която има поне 2 от съседните 4 клетки (с обща страна) празни се стопява изцяло за 1 час, а останалите клетки не се топят изобщо. Напишете програма, която прочита от текстов файл размера и съдържанието на таблицата:

```
8
00000000
00**0000
00**0**0
0*****0
0*****00
0**0**00
00000000
00000000
```

В резултат програмата следва да извежда на екрана броя часове, за които айсбергът ще се разтопи изцяло. В горния пример изходът на програмата следва да бъде: 4.

# Problem: Melting Iceberg II

Реализирайте конзолно приложение за интерактивно въвеждане и редактиране на таблицата от задача 3. Приложението следва да поддържа текстово меню с възможности за:

- 1) редактиране на таблицата;
- 2) създаване на нова празна таблица с възможност за редактиране;
- 3) прочитане на таблицата от текстов файл;
- 4) запис на таблицата в текстов файл;
- 5) изход от програмата.

Редактирането на таблицата трябва да стане в текстов вид, интерактивно от клавиатурата с поддържане на активен курсор (символ '#') и с натискане на '+' за запълване на клетката където е курсора и '-' за изчистване на клетката, където е курсора. Преместването на курсора става със стрелките от клавиатурата.

След натискане на всеки клавиш се извежда цялата таблица и един празен ред за разделител. Редактирането приключва с натискане на клавиша <Enter>, след което се връщаме в главното меню на програмата, като редактираната таблица се запомня.

# Thank's for Your Attention!



Trayan Iliev

IPT – Intellectual Products & Technologies

<http://iproduct.org/>

<https://github.com/iproduct>

<https://twitter.com/trayaniliev>

<https://www.facebook.com/IPT.EACAD>