

Desvendando programação com Python

O conteúdo deste material é licenciado sob a
Licença Atribuição Creative Commons 3.0 Brasil
(CC BY 3.0 BR)

<https://creativecommons.org/licenses/by/3.0/br/>



Copyright 2021 **Carlos ROLAND**



python™

O que é um programa?

Qual a palavra mais usada no texto?

We are surrounded in our daily lives with computers ranging from laptops to cell phones. We can think of these computers as our “personal assistants” who can take care of many things on our behalf. The hardware in our current-day computers is essentially built to continuously ask us the question, “What would you like me to do next?”

Programmers add an operating system and a set of applications to the hardware and we end up with a Personal Digital Assistant that is quite helpful and capable of helping us do many different things. Our computers are fast and have vast amounts of memory and could be very helpful to us if we only knew the language to speak to explain to the computer what we would like it to “do next”. If we knew this language, we could tell the computer to do tasks on our behalf that were repetitive. Interestingly, the kinds of things computers can do best are often the kinds of things that we humans find boring and mind-numbing.

For example, look at the first three paragraphs of this chapter and tell me the most commonly used word and how many times the word is used. While you were able to read and understand the words in a few seconds, counting them is almost painful because it is not the kind of problem that human minds are designed to solve. For a computer the opposite is true, reading and understanding text from

a piece of paper is hard for a computer to do but counting the words and telling you how many times the most used word was used is very easy for the computer: python word

O que é um programa?

```
# script file: wordsCount.py
```

```
name = input('Enter file:')  
handle = open(name, 'r')  
counts = dict()
```

```
for line in handle:  
    words = line.split()  
    for word in words:  
        counts[word] = counts.get(word, 0) + 1
```

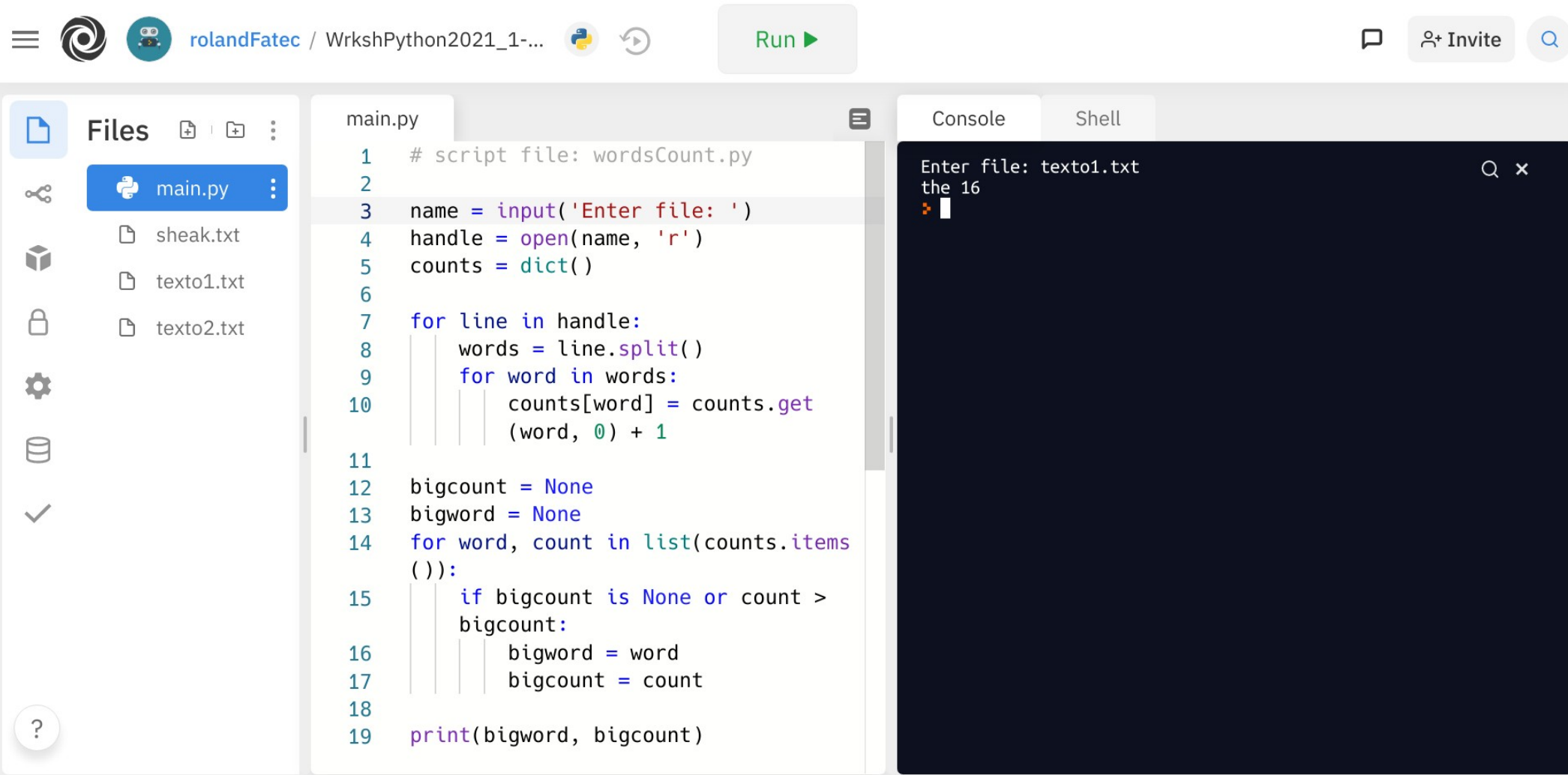
```
bigcount = None  
bigword = None  
for word, count in list(counts.items()):  
    if bigcount is None or count > bigcount:  
        bigword = word  
        bigcount = count
```

```
print(bigword, bigcount)
```

Executando o script

```
$python3 wordsCount.py  
Enter file:words.txt  
the 16  
$
```

Executando o script



The image shows a web-based Python IDE interface. At the top, there's a header with a menu icon, a logo, the username 'rolandFatec', a project path 'WrkshPython2021_1-...', a Python logo, a refresh icon, a green 'Run' button with a play icon, a chat icon, and an 'Invite' button. Below the header, the interface is divided into three main sections:

- Files:** A sidebar on the left showing a file explorer. It contains a 'main.py' file (highlighted in blue) and three text files: 'sheak.txt', 'texto1.txt', and 'texto2.txt'.
- Code Editor:** The central area displays the content of 'main.py'. The code is a Python script that reads a file, counts the frequency of words, and prints the most frequent word and its count. The code is as follows:

```
1 # script file: wordsCount.py
2
3 name = input('Enter file: ')
4 handle = open(name, 'r')
5 counts = dict()
6
7 for line in handle:
8     words = line.split()
9     for word in words:
10         counts[word] = counts.get(word, 0) + 1
11
12 bigcount = None
13 bigword = None
14 for word, count in list(counts.items()):
15     if bigcount is None or count > bigcount:
16         bigword = word
17         bigcount = count
18
19 print(bigword, bigcount)
```
- Console:** A dark-themed panel on the right showing the output of the script. It displays the prompt 'Enter file: texto1.txt' followed by the output 'the 16'.

Hello, World!

Para aprender Python:



Welcome To Colaboratory

File Edit View Insert Runtime Tools Help



☰	Table of contents	×
🔍	Getting started	
	Data science	
<>	Machine learning	
📁	More Resources	
	Machine Learning Examples	
📄	Section	

+ Code + Text | 📄 Copy to Drive

What is Colaboratory?

Colaboratory, or "Colab" for short, allows you to write and execute Python in your browser, with

- Zero configuration required
- Free access to GPUs
- Easy sharing

Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch [Introduction to Colab](#) to learn more, or just get started below!

▼ Getting started

The document you are reading is not a static web page, but an interactive environment called a **Colab notebook** that lets you write and execute code.

For example, here is a **code cell** with a short Python script that computes a value, stores it in a variable, and prints the result:

```
seconds_in_a_day = 24 * 60 * 60
seconds_in_a_day
```

86400

To execute the code in the above cell, select it with a click and then either press the play button to the left of the code, or use the keyboard shortcut "Command/Ctrl+Enter". To edit the code, just click the cell and start editing.

Variables that you define in one cell can later be used in other cells:

```
[ ] seconds_in_a_week = 7 * seconds_in_a_day
seconds_in_a_week
```


604800

Colab notebooks allow you to combine **executable code** and **rich text** in a single document, along with **images**, **HTML**, **LaTeX** and more. When you create your own Colab notebooks, they are stored in your Google Drive account. You can easily share your Colab notebooks with co-workers

Hello, World!

Para aprender Python:



[Tutorials](#)▼[References](#)▼[Exercises](#)▼

HTMLCSSJAVASCRIPTSQLPYTHONPHPBOOTSTRAPHOW TOW3.CSSJAJQUERYC++C#R

Python Tutorial

Python HOME

Python Intro

Python Get Started

Python Syntax

Python Comments

Python Variables

Python Data Types

Python Numbers

Python Casting

Python Strings

Python Booleans

Python Operators

Python Lists

Python Tuples

Python Sets

Python Dictionaries

Python If...Else

Python While Loops

Python For Loops

Python Functions

Python Lambda

Python Arrays

Python Classes/Objects

Python Inheritance

Python Iterators


Python Scope

Python Modules

Python Tutorial

< Home

Next >



Python is a programming language.

Python can be used on a server to create web applications.

[Start learning Python now »](#)

Learning by Examples

With our "Try it Yourself" editor, you can edit Python code and view the result.

Example

```
print("Hello, World!")
```

[Try it Yourself »](#)

Variáveis

Nome para um valor armazenado em memória

Variáveis

Nome para um valor armazenado em memória

Criadas por comandos de atribuição (=)

```
>>> message = 'Dado inválido! Tecle <ENTER>.'
```

```
>>> n = 17
```

```
>>> pi = 3.1415926535897931
```

Variáveis

Executar os comandos. Qual o resultado? Porque?

```
>>> message = 'Dado inválido! Tecle <ENTER>.'
```

```
>>> n = 17
```

```
>>> pi = 3.1415926535897931
```

Variáveis

Executar os comandos. Qual o resultado? Porque?

```
>>> message = 'Dado inválido! Tecle <ENTER>.'  
>>> n = 17  
>>> pi = 3.1415926535897931
```

Como visualizar os valores?

```
>>> print(message)  
>>> print(n)  
>>> print(pi)
```

Variáveis

Executar os comandos. Qual o resultado? Porque?

```
>>> message = 'Dado inválido! Tecle <ENTER>.'  
>>> n = 17  
>>> pi = 3.1415926535897931
```

Como visualizar os valores?

```
>>> print(message)  
>>> print(n)  
>>> print(pi)
```

Como visualizar os tipos?

```
>>> type(message)  
>>> type(n)  
>>> type(pi)
```

Time for
a Break

Nomes de variáveis e palavras-chave

Significativas e descritivas

Nomes de variáveis e palavras-chave

Significativas e descritivas

Sem limitação de tamanho

Nomes de variáveis e palavras-chave

Significativas e descritivas

Sem limitação de tamanho

Podem conter letras e números

Nomes de variáveis e palavras-chave

Significativas e descritivas

Sem limitação de tamanho

Podem conter letras e números

Não podem começar com números

Nomes de variáveis e palavras-chave

Significativas e descritivas

Sem limitação de tamanho

Podem conter letras e números

Não podem começar com números

Sensíveis a caixa

Boas práticas (BP): iniciar com minúsculas

Nomes de variáveis e palavras-chave

Significativas e descritivas

Sem limitação de tamanho

Podem conter letras e números

Não podem começar com números

Sensíveis a caixa

Boas práticas (BP): iniciar com minúsculas

Podem usar sublinhado (_)

BP: usar _ no início para variáveis de bibliotecas

Nomes de variáveis e palavras-chave

Readability Counts

https://youtu.be/knMg6G9_XCg

Nomes de variáveis e palavras-chave

Testar:

```
>>>76trombones = 'big parade'
```

```
>>>more@ = 1000000
```

```
>>>class = 'Advanced Theoretical Zymurgy'
```

```
>>>precoUnitario = 3456.80
```

```
>>>pr_unit = 3456.80
```

```
>>>lado1 = 10
```

```
>>>taxaJuroAnual = 546.68
```

Palavras reservadas - revisão

**and
as
assert
break
class
continue
def
del
elif
else
except**

**False
finally
for
from
global
if
import
in
is
lambda
None
nonLocal
not**

**or
pass
raise
return
True
try
while
with
yield

>import keyword
>keyword.kwlist**

Comandos

Unidade de código executável pelo interpretador

Comandos

Unidade de código executável pelo interpretador

Exemplos:

`print()`

`=`

`type()`

`int()`

`str()`

`float()`

Comandos

Unidade de código executável pelo interpretador

Exemplos:

```
print()  
=  
type()  
int()  
str()  
float()
```

No modo interativo são executados quando teclado <ENTER>

Comandos

Unidade de código executável pelo interpretador

Exemplos:

```
print()  
=  
type()  
int()  
str()  
float()
```

No modo interativo são executados quando teclado <ENTER>

No modo script todos são executados na sequência de escrita

Nomes de variáveis e palavras-chave

Testar (modos interativo e programado):

```
>>>firstname = 'Rick'
```

```
>>>lastname = 'Riordan'
```

```
>>>series = 'Percy Jackson'
```

```
>>>book = 'Percy Jackson e os Olimpianos'
```

```
>>>precoUnitario = 34.90
```

```
>>>pr_unit = 34.90
```

Nomes de variáveis e palavras-chave

Testar (modos interativo e programado):

```
>>>print(firstname, lastname)
```

```
>>>print(book, precoUnitario)
```

```
>>>print(book, pr_unit*2)
```

```
>>>print(book,":",pr_unit*2)
```

Concatenando – juntando - strings:

```
firstname = 'Rick'
```


```
lastname = 'Riordan'
```

```
print(firstname + lastname)
```

```
print(firstname + " " + lastname)
```

VAMOS PROGRAMAR!


Para treinar: URI Online Judge - Problems & Contests

ENGLISH

[LOGIN](#) [REGISTER](#) [FORUM](#) [CONTESTS](#) [PROBLEMS](#) [RANKS](#) [FOR PROFESSORS](#) [FOR COMPANIES](#)

URI ONLINE JUDGE


THE URI ONLINE JUDGE



The URI Online Judge is a project that is being developed by the Computer Science Department of URI University. The main goal of the project is to provide programming practice and knowledge sharing.

CREDITS


PROBLEM REPOSITORY



The URI Online Judge contains more than 1000 problems divided in 8 big categories. This division help the users to focus on specific programming topics. All problems are available in Portuguese and English.

REPOSITORY

URI ONLINE JUDGE FORUM



The URI Online Judge Forum is the right place for you to get help and to help other users. Share your knowledge and experience in algorithms and

URI ONLINE JUDGE PROBLEMS & CONTESTS

SIGN IN

EMAIL

PASSWORD

☐ REMEMBER ME (7 DAYS)

SIGN IN

[FACEBOOK](#) [GOOGLE](#) [GITLAB](#)


[GITHUB](#) [BITBUCKET](#) [TWITCH](#)

§ By signing up with a social platform you **AGREE** with the URI Online Judge Terms & Conditions.


You need to allow your EMAIL to be provided by the social sign in. Therefore, allow the default permissions when prompted.

FIRST TIME HERE?
SIGN UP today to view materials, solve problems, ask questions and much more.

[RESET PASSWORD](#) [ACTIVATE](#)

 **PRÊMIO GUIA DO ESTUDANTE DESTAQUES DO ANO**


COMPETITION AND RANKING



Solve the available problems using 11 programming languages, and compete with other users. As a challengee, improve your ranking, solving as many problems as possible and tuning your source code to run faster.

CHECK THE RANK


URI ONLINE JUDGE ACADEMIC



The URI Online Judge Academic is an unique module for professors and team coaches. Here you can create disciplines and lists of exercises. You can also track the progress of your students giving feedback in real time.

ACCESS ACADEMIC

URI ONLINE JUDGE CONTESTS



The URI Online Judge website also has public contests on a regular basis. Get in touch with us to host your contest at URI Online Judge for free. Create new



That's all Folks!