

# 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**



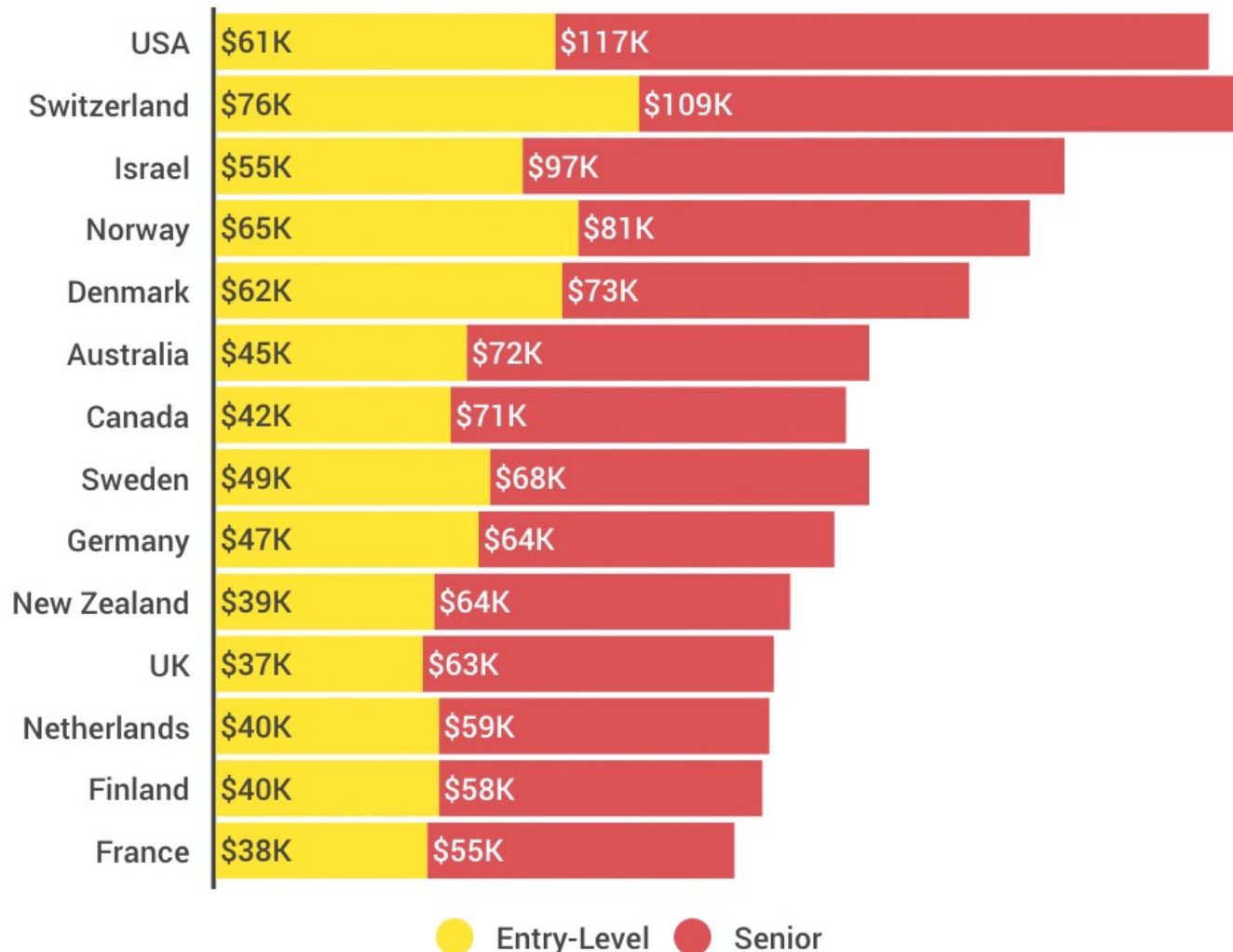
python™

# Desvendando programação

**Porque aprender programação**

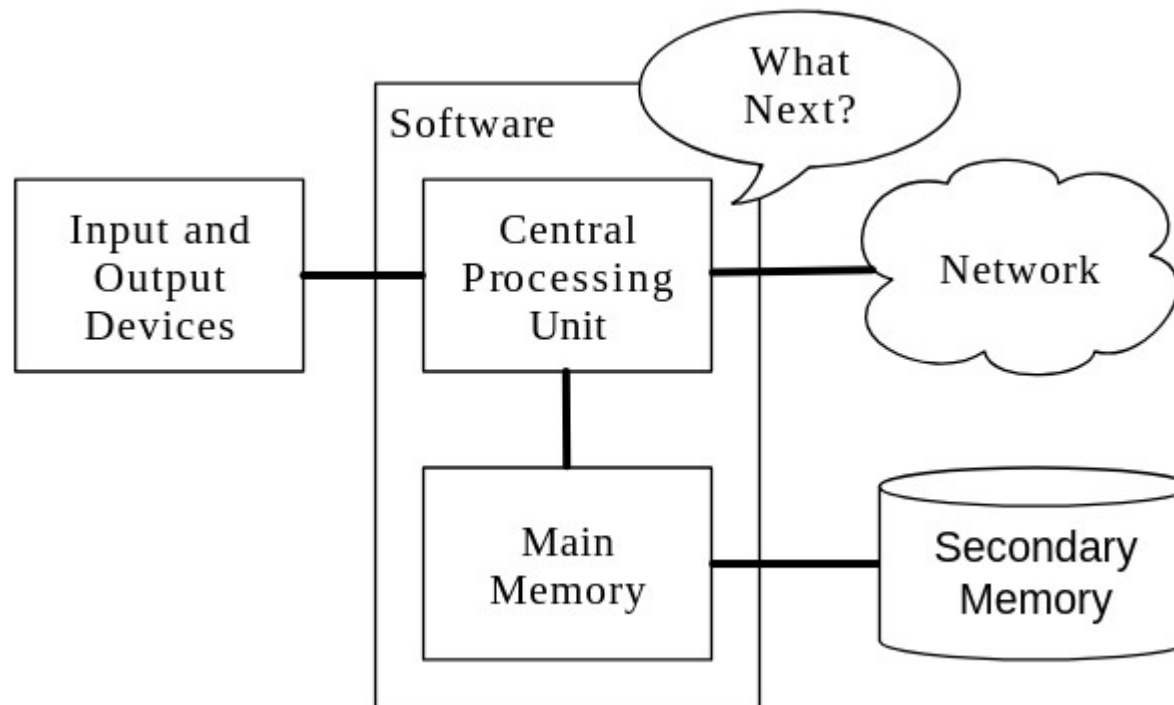
# Motivação

## Average Entry-Level and Senior Software Developer Salaries in the World

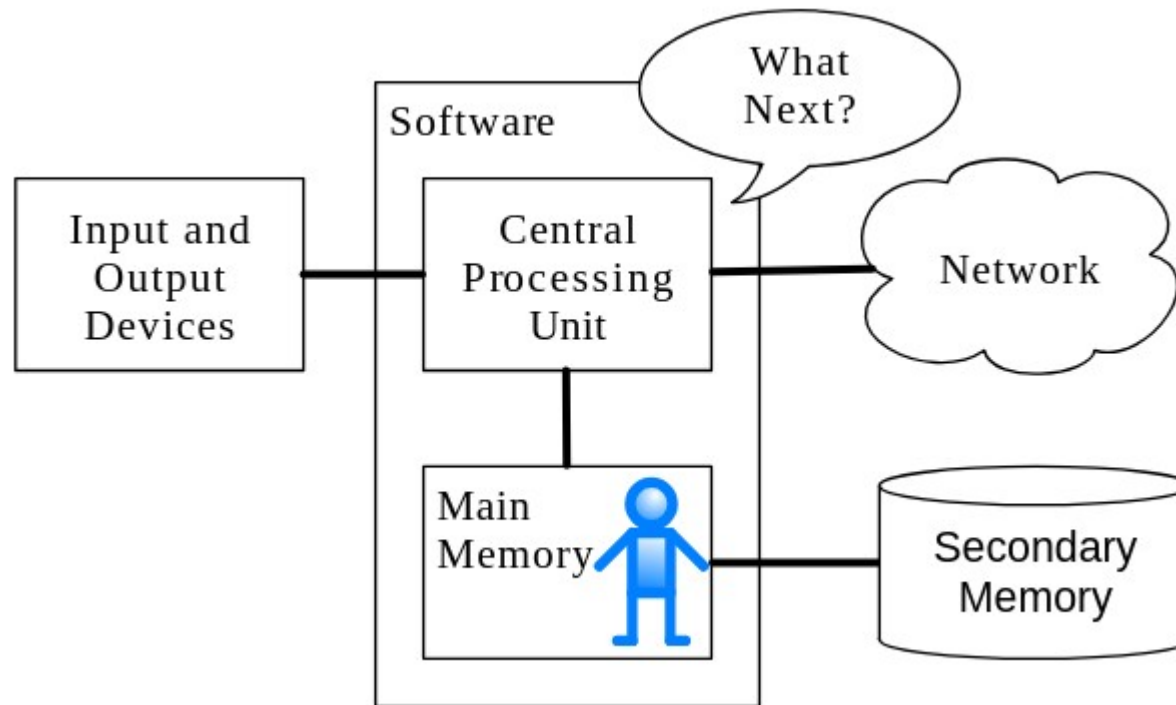


Source: [daxx.com](https://daxx.com)

# Arquitetura de computadores



# Arquitetura de computadores



# Desvendando programação

## Habilidades para programação

Identificar um problema de análise de dados ou informação

e

desenvolver um programa computacional para resolvê-lo

# Desvendando programação

## Habilidades para programação

Conhecer uma linguagem de programação  
vocabulário e gramática;

# Desvendando programação

## Habilidades para programação

Conhecer uma linguagem de programação  
vocabulário e gramática;

Conhecer as palavras reservadas e como construir  
sentenças gramaticalmente corretas;



# Desvendando programação

## Habilidades para programação

Conhecer uma linguagem de programação  
vocabulário e gramática;

Conhecer as palavras reservadas e como construir  
sentenças gramaticalmente corretas;

Contar uma história

# Desvendando programação

## Habilidades para programação

Contar uma estória:

escrevendo uma estória, se combinam palavras e sentenças para transmitir uma ideia ao leitor. Existe habilidade e arte na construção de estórias.

em programação, o programa é a estória e o problema a ser resolvido é a ideia.

# Desvendando programação

## Vocabulário – palavras reservadas

**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**

# Desvendando programação

## Sentenças

```
>>>print("Hello world!")  
Hello world!
```

```
>>>print('Hello world!')  
Hello world!
```

# Desvendando programação com PYTHON

- Porque Python?

# **We Love Python**

- **Permite focar no problema, sem perder tempo na sintaxe**
- **Interativa**
- **Alta produtividade**
- **Baterias inclusas**
- **Comunidade livre, forte, diversificada, alegre e acolhedora**
- **Linguagem de verdade: orientada a objetos, funcional, estruturada**
- **Linguagem de uso geral vs nichos**
- **Simples de iniciar, sem esgotar**

# Desvendando programação com PYTHON

**Porque aprender Python**

# Desvendando programação com PYTHON

Porque Python é fácil e simples

```
Python 3.7.4 (default, Jul 9 2019, 00:06:43)  
[GCC 6.3.0 20170516] on linux  
➤ import this  
The Zen of Python, by Tim Peters
```



# Desvendando programação

## com PYTHON

```
1 >> import this
2
3 Beautiful is better than ugly.
4 Explicit is better than implicit.
5 Simple is better than complex.
6 Complex is better than complicated.
7 Flat is better than nested.
8 Sparse is better than dense.
9 Readability counts.
10 Special cases aren't special enough to break the rules.
11 Although practicality beats purity.
12 Errors should never pass silently.
13 Unless explicitly silenced.
14 In the face of ambiguity, refuse the temptation to guess.
15 There should be one-- and preferably only one --obvious way to do it.
16 Although that way may not be obvious at first unless you're Dutch.
17 Now is better than never.
18 Although never is often better than *right* now.
19 If the implementation is hard to explain, it's a bad idea.
20 If the implementation is easy to explain, it may be a good idea.
21 Namespaces are one honking great idea -- let's do more of those!
```

# Motivação

## 1. SQL, Java and Javascript are the most advertised software skills

Overall the findings at Gooroo and Indeed were very similar: ranging from around 3,500 outstanding job postings for Perl at the bottom to 31,000 outstanding postings for SQL at the top. It has to be said that we noticed that in many cases SQL was demanded *in combination with* a host of other skills.

In total there are about 250,000 outstanding postings for these 17 skills at the moment. Half of which are for the top-5:

1. SQL
2. Java
3. Javascript
4. C
5. Python

# Motivação

## 2. Hiring a Ruby, Python and C++ developer is most expensive

Whereas Ruby might not be among the most advertised skills, it is definitely among the relatively most scarce skills. When looking at the salaries offered for developers, Ruby tops the list:

1. Ruby
2. Python
3. C++
4. NodeJS
5. Android
6. iOS
7. Java
8. Javascript

Interestingly, Python is the only skill that is both high in absolute demand and high in relative demand (i.e. expensive). Of the other 4 highly advertised languages three (Java, Javascript and C) are somewhere in the middle salary range. SQL is the only one that is found at the bottom of the salary range.

# Origem do nome Python



## Python: quem usa



**1h video/s 4 billion views.**

# Python: quem usa



## Python: quem usa

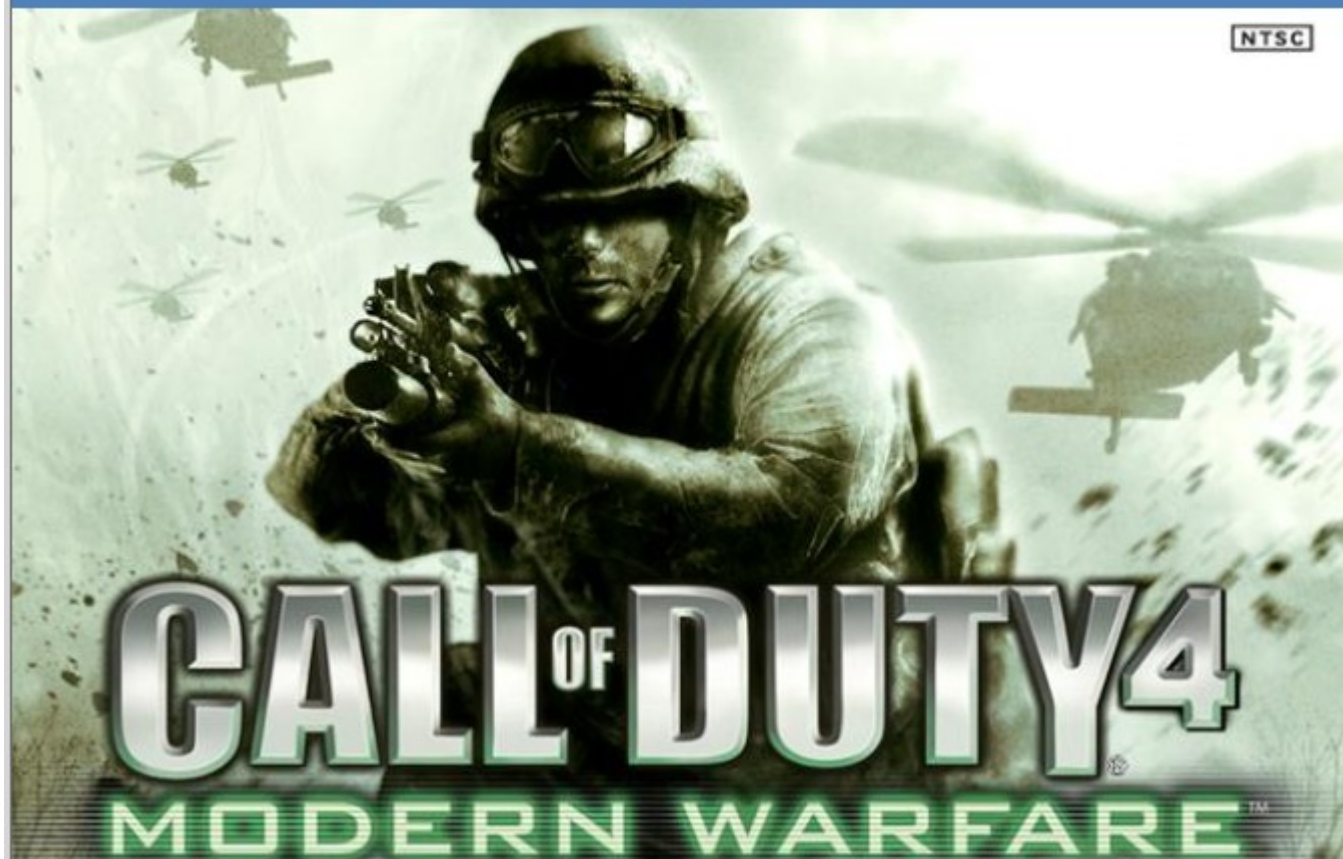
DISQUS

NATIONAL  
GEOGRAPHIC





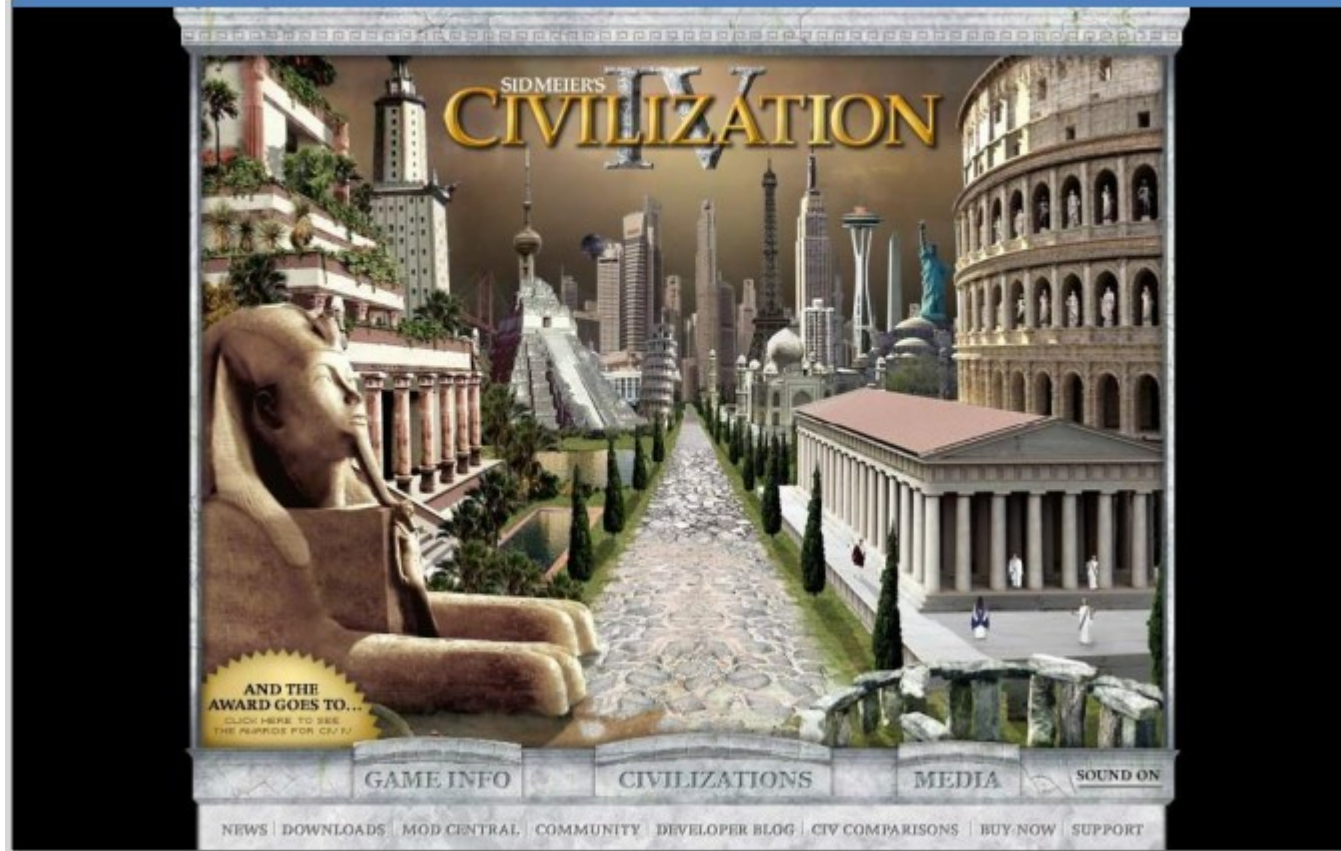
**Python: quem usa**



Fonte: <https://pt.slideshare.net/fmasanori/twp00-apresentao-e-motivao>



# Python: quem usa



Fonte: <https://pt.slideshare.net/fmasanori/twp00-apresentao-e-motivao>

# Python: quem usa

**THE FOUNDRY**

Home > Products > NUKE

(empty) 

NUKE | Advancing the art of digital compositing

[Home](#)  
[News](#)  
[Products](#)  
[Support](#)  
[Training](#)  
[About](#)  
[Resellers](#)  
[Users](#)



Rio © 2011 Fox. All rights reserved. Images courtesy of Blue Sky Studios.

**NUKE**

"NUKE allows us to leverage our 3D pipeline while remaining in the compositing environment which puts a great deal of power into the hands of the compositors. We work on very complex shots and the remarkable speed at which NUKE operates means our artists can focus on the art of visual effects and not have their creativity impeded"

**NUKE**

6.3v5

[Free 15 day trial](#)  
[Buy](#)  
[Rent](#)

简体中文  
繁體

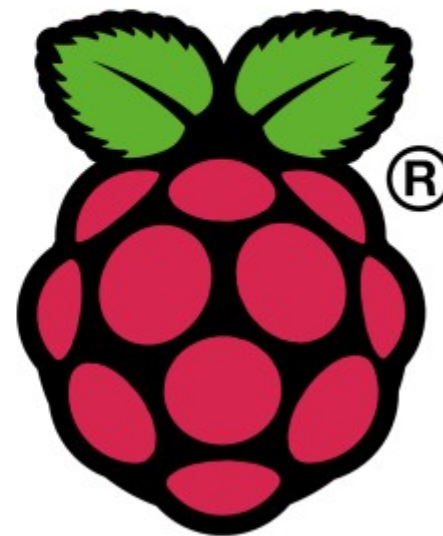


Fonte: <https://pt.slideshare.net/fmasanori/twp00-apresentao-e-motivao>

# Python: quem usa



# Python: quem usa



Você sabia que o “Pi” do nome se origina de Python?



# Python: quem usa



**Python: quem usa**

**luizalabs**

# Conversando com Python – repl.it



Sign up

## BUILD AND DEPLOY IN SECONDS

Instant programming environment for your favorite language

```
1  const http = require('http');
2  const server = http.createServer();
3
4  server.on('request', (req, res) => {
5    res.end('hello world!');
6  });
7
8  server.listen(3000);
```

Node.js v9.8.0 on linux

Server started on port 3000

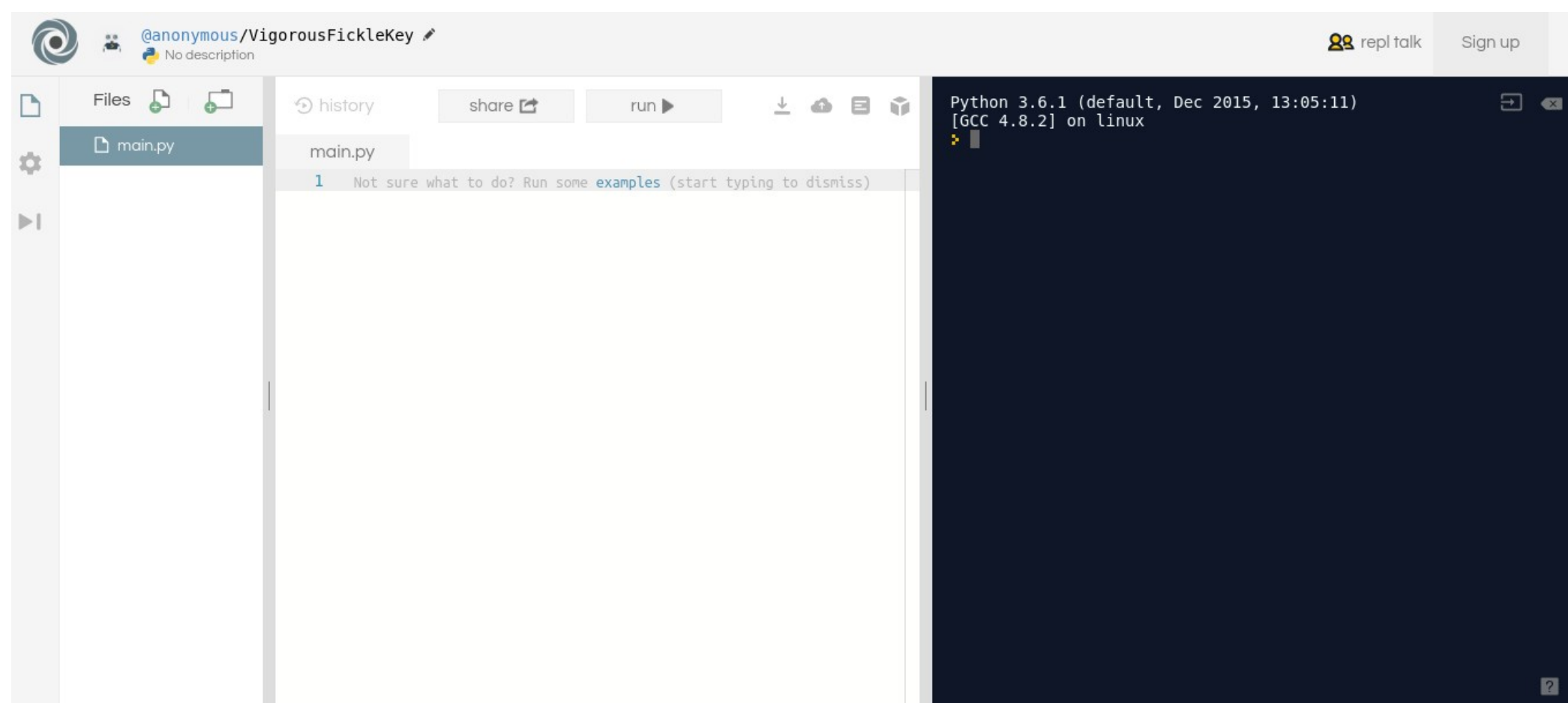
<https://you.repl.co>

hello world!

Python3

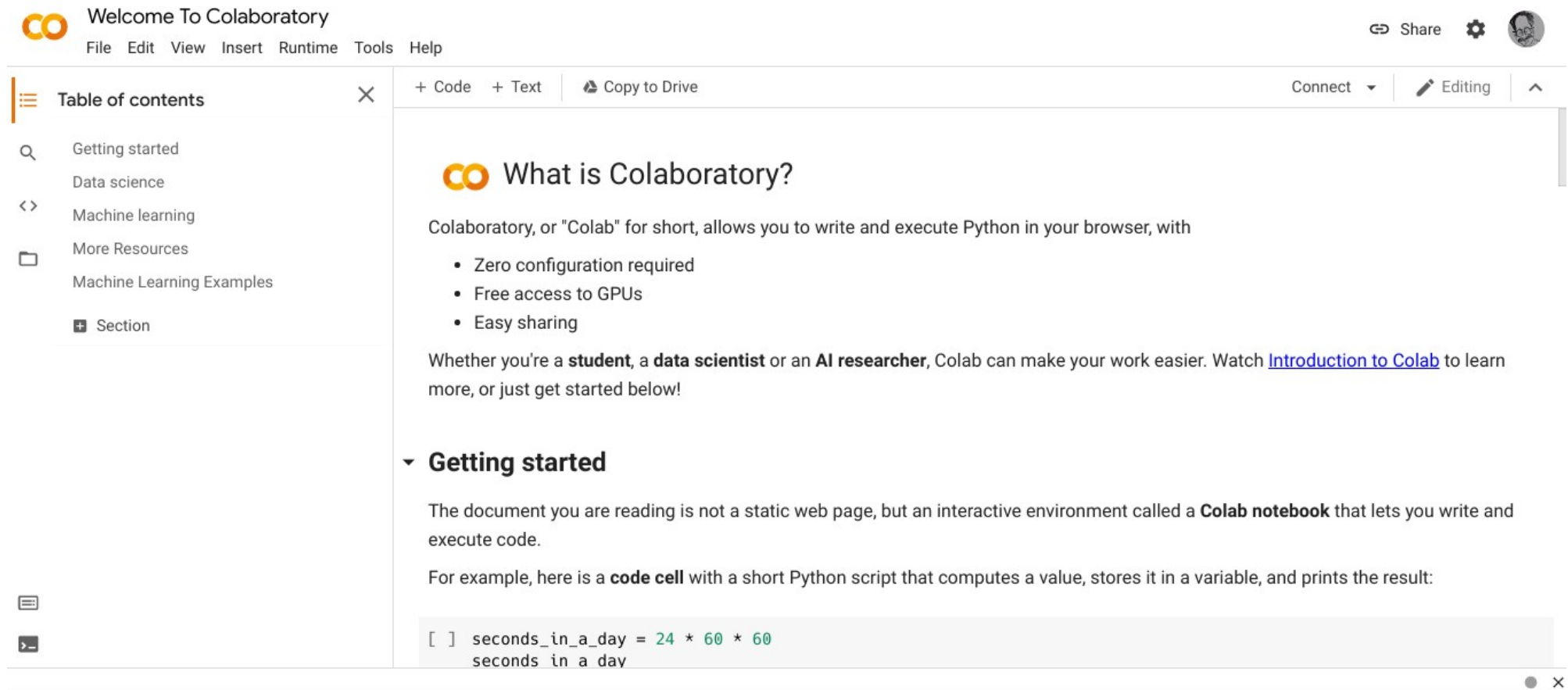
# Conversando com Python – repl.it



The image shows the repl.it web interface for a Python REPL. The top header bar includes the Repl.it logo, the username `@anonymous/VigorousFickleKey`, a "No description" label, a "repl talk" button with a user icon, and a "Sign up" button. Below the header, the interface is divided into three main sections. On the left is a sidebar with a "Files" tab, a "main.py" file icon, and a "history" tab. The middle section is a code editor showing the file `main.py` with a single line of text: `1 Not sure what to do? Run some examples (start typing to dismiss)`. Above the code editor are buttons for "share", "run", and a set of utility icons (download, upload, menu, and a cube). On the right is a dark-themed terminal window. The terminal output shows the Python environment: `Python 3.6.1 (default, Dec 2015, 13:05:11)` and `[GCC 4.8.2] on linux`, followed by a prompt `>` and a cursor. A small help icon is visible in the bottom right corner of the terminal.



# Conversando com Python – COLAB



The screenshot displays the Google Colaboratory (COLAB) web interface. At the top, the 'Welcome To Colaboratory' header is visible, along with a menu bar containing 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. On the right side of the header, there are links for 'Share', a settings gear icon, and a user profile picture. A left-hand sidebar titled 'Table of contents' lists various topics: 'Getting started', 'Data science', 'Machine learning', 'More Resources', 'Machine Learning Examples', and a 'Section' button. The main content area is titled 'What is Colaboratory?' and features the COLAB logo. It explains that Colaboratory, or 'Colab', allows users to write and execute Python code directly in their browser. Key features listed include zero configuration required, free access to GPUs, and easy sharing. A paragraph follows, stating that whether you're a student, data scientist, or AI researcher, Colab can simplify your work, with a link to 'Introduction to Colab'. Below this, a section titled 'Getting started' describes the interactive 'Colab notebook' environment. It provides an example of a 'code cell' containing a short Python script that calculates the number of seconds in a day. The script is shown in a light gray code editor box with a terminal icon on the left.

Welcome To Colaboratory

File Edit View Insert Runtime Tools Help

Share

Table of contents

- Getting started
- Data science
- Machine learning
- More Resources
- Machine Learning Examples
- Section

+ Code + Text Copy to Drive

Connect Editing

## 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
```

# Conversando com Python - Sololearn

SOLOLEARN

COURSES

CODE PLAYGROUND

DISCUSS

TOP LEARNERS

BLOG

MY CODES(0)

Python 3

Dark

Light

Python 3

Output

SHARE



```
1 print('Hello world!')
```

Hello world!

SAVE

SAVE AS

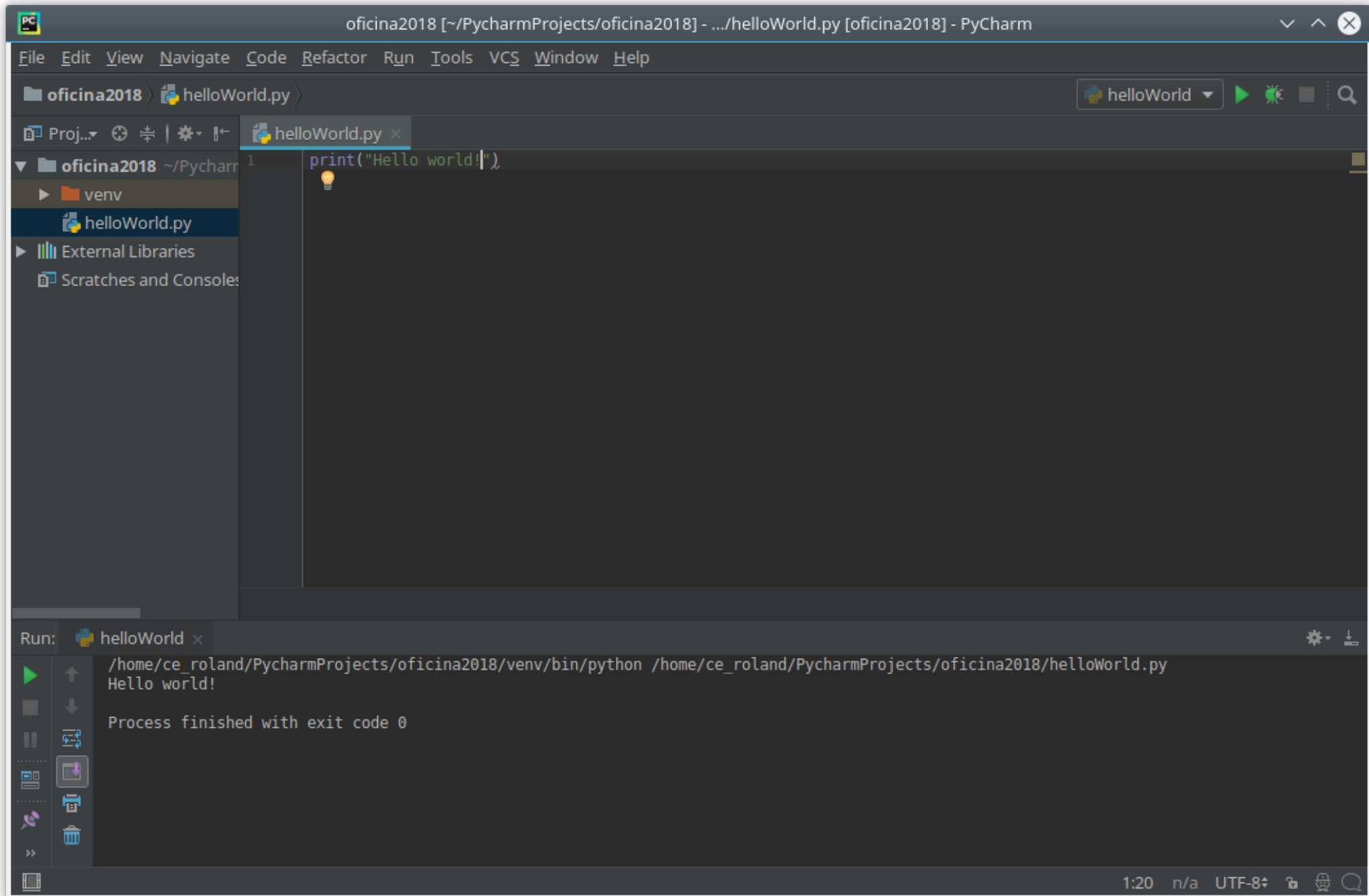
RESET

▶ RUN

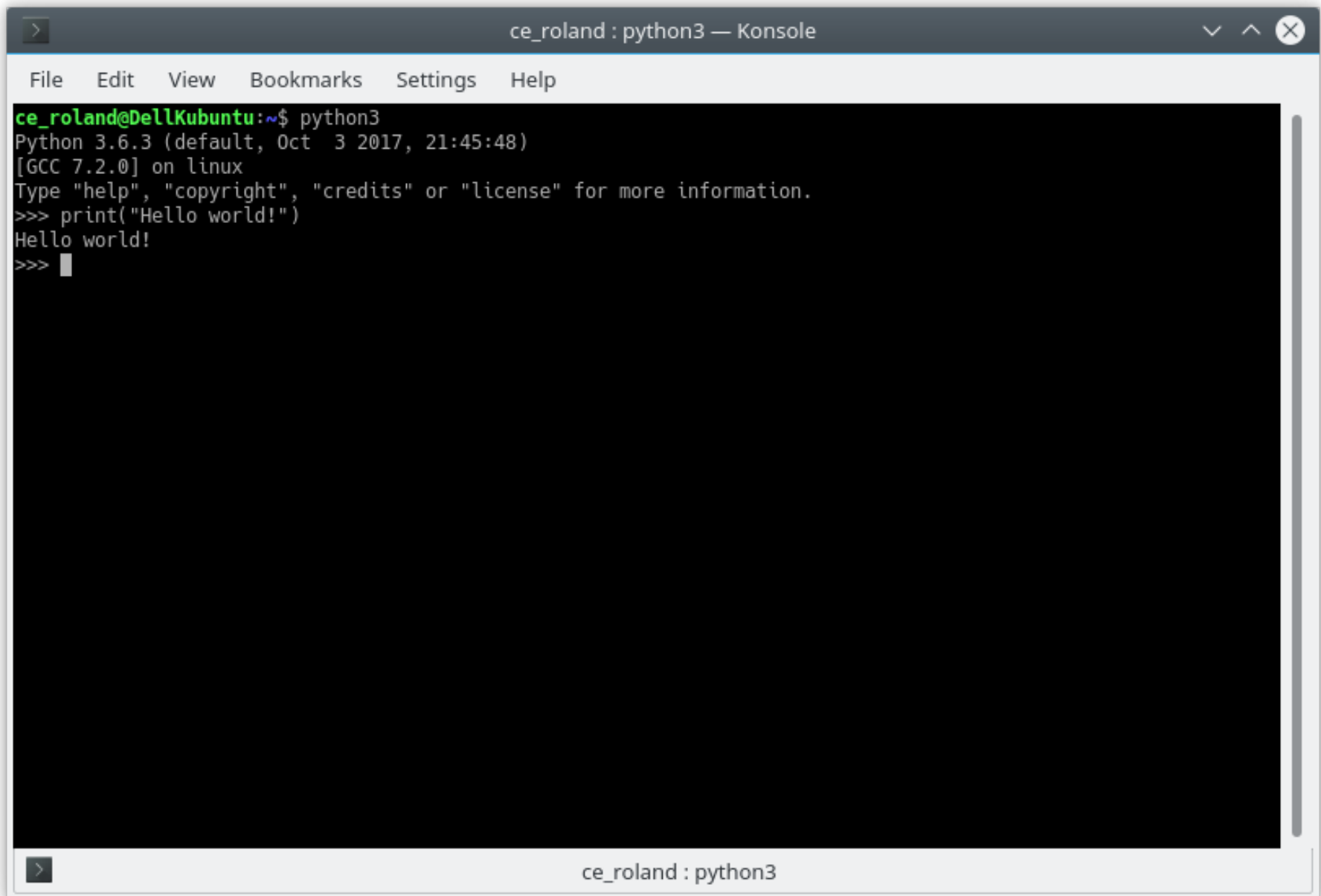
GET IT ON  
Google play

Download on the  
App Store

# Conversando com Python - pyCharm

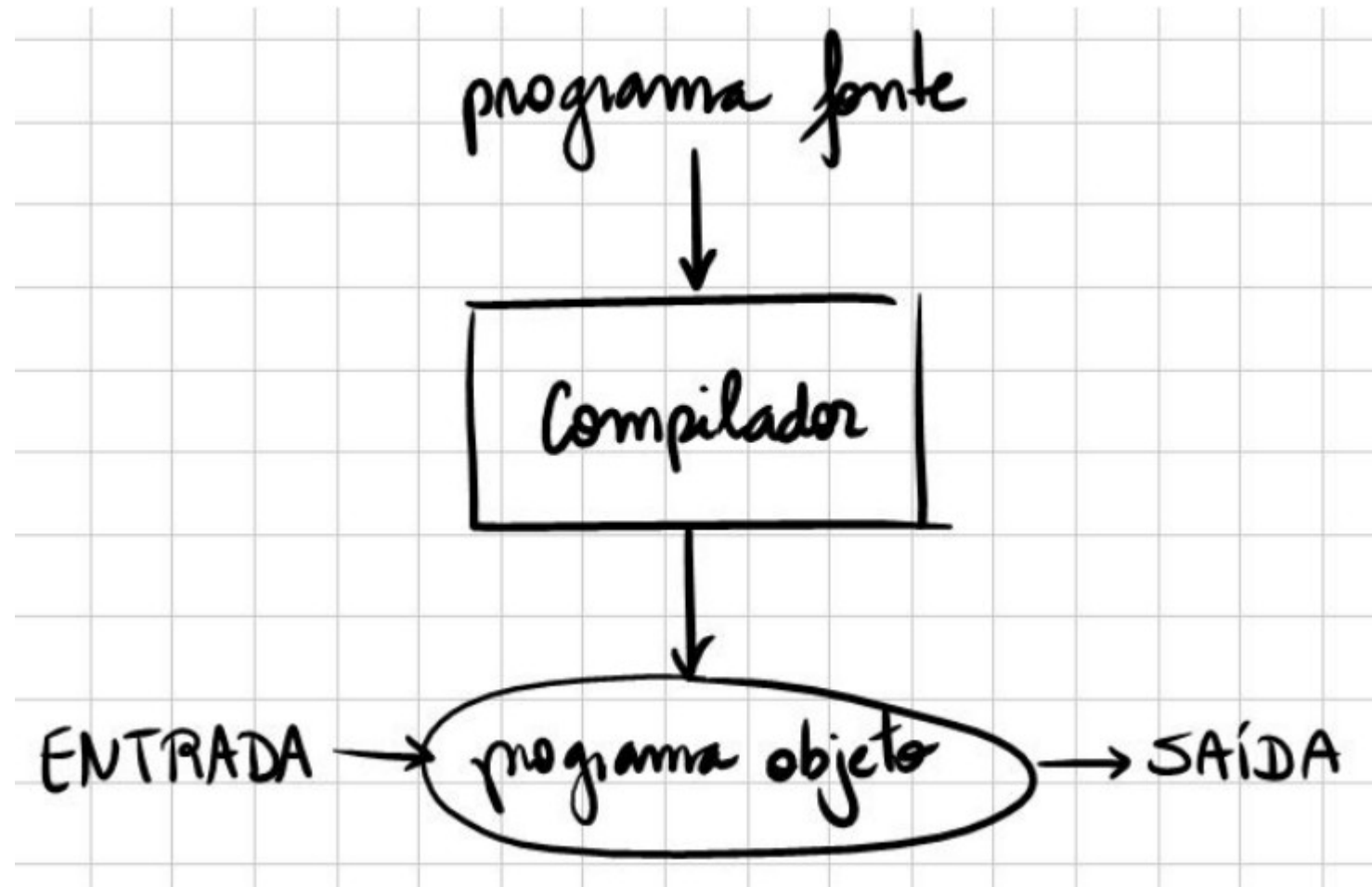


# Conversando com Python - terminal

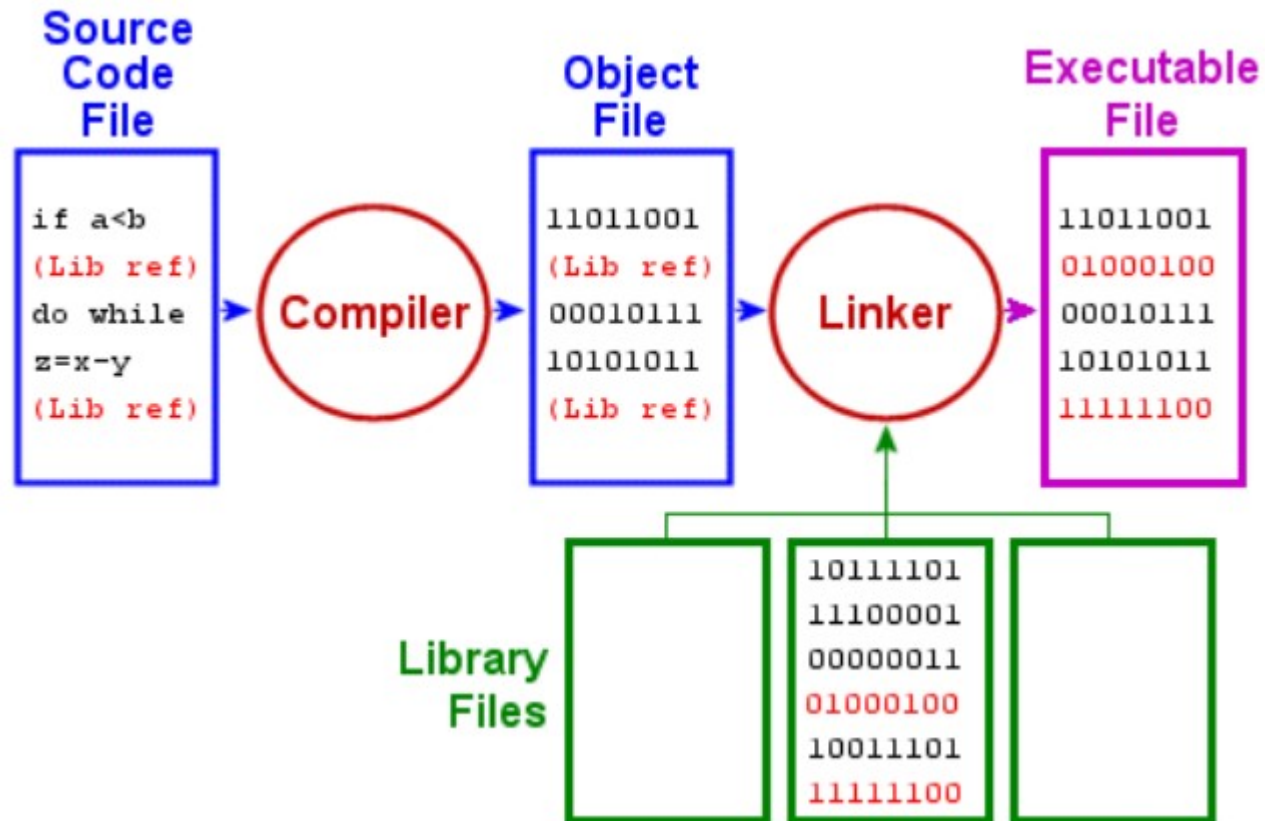


```
ce_roland : python3 — Konsole
File Edit View Bookmarks Settings Help
ce_roland@DellKubuntu:~$ python3
Python 3.6.3 (default, Oct  3 2017, 21:45:48)
[GCC 7.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello world!")
Hello world!
>>> 
```

# Compilador X Interpretador X Híbrido



# Compilador X Interpretador X Híbrido



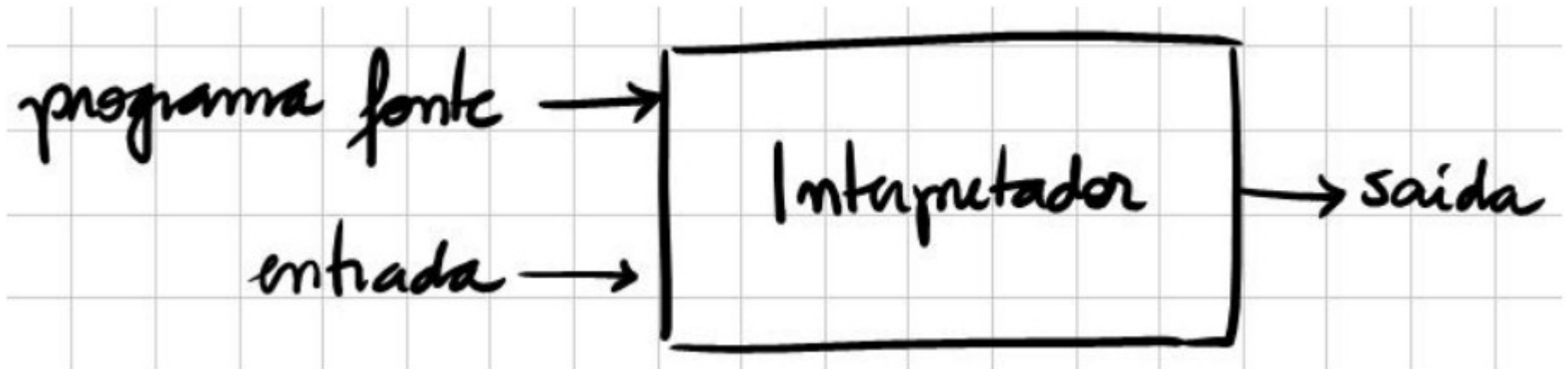
# Compilador X Interpretador X Híbrido

```
func greet() = {  
  Console.println("Hello, World!")  
}
```

↓  
Compilador  
↓

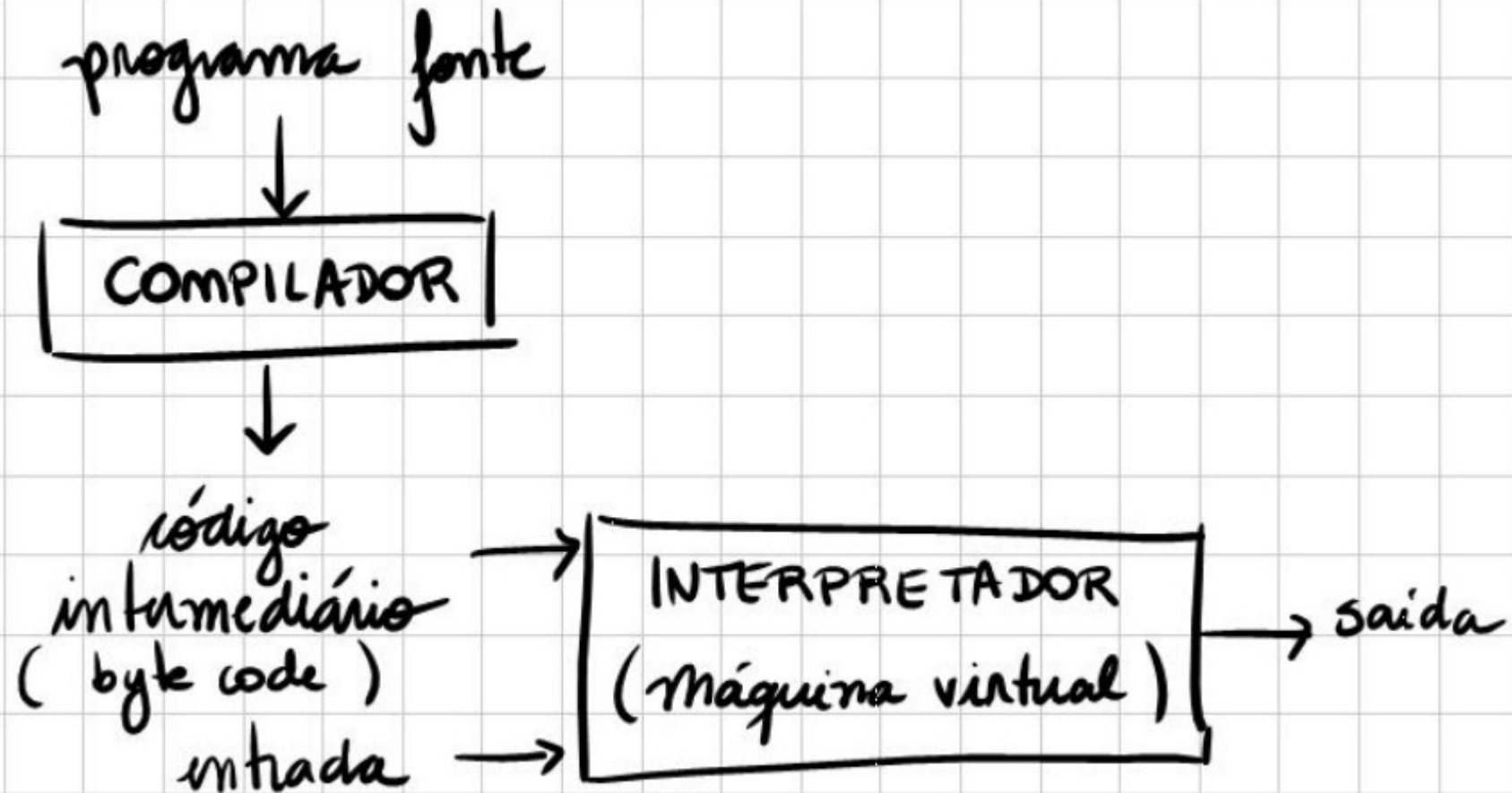
```
10100111100  
11110011001  
10010010010  
10110111001  
11101111011
```

# Compilador X Interpretador X Híbrido

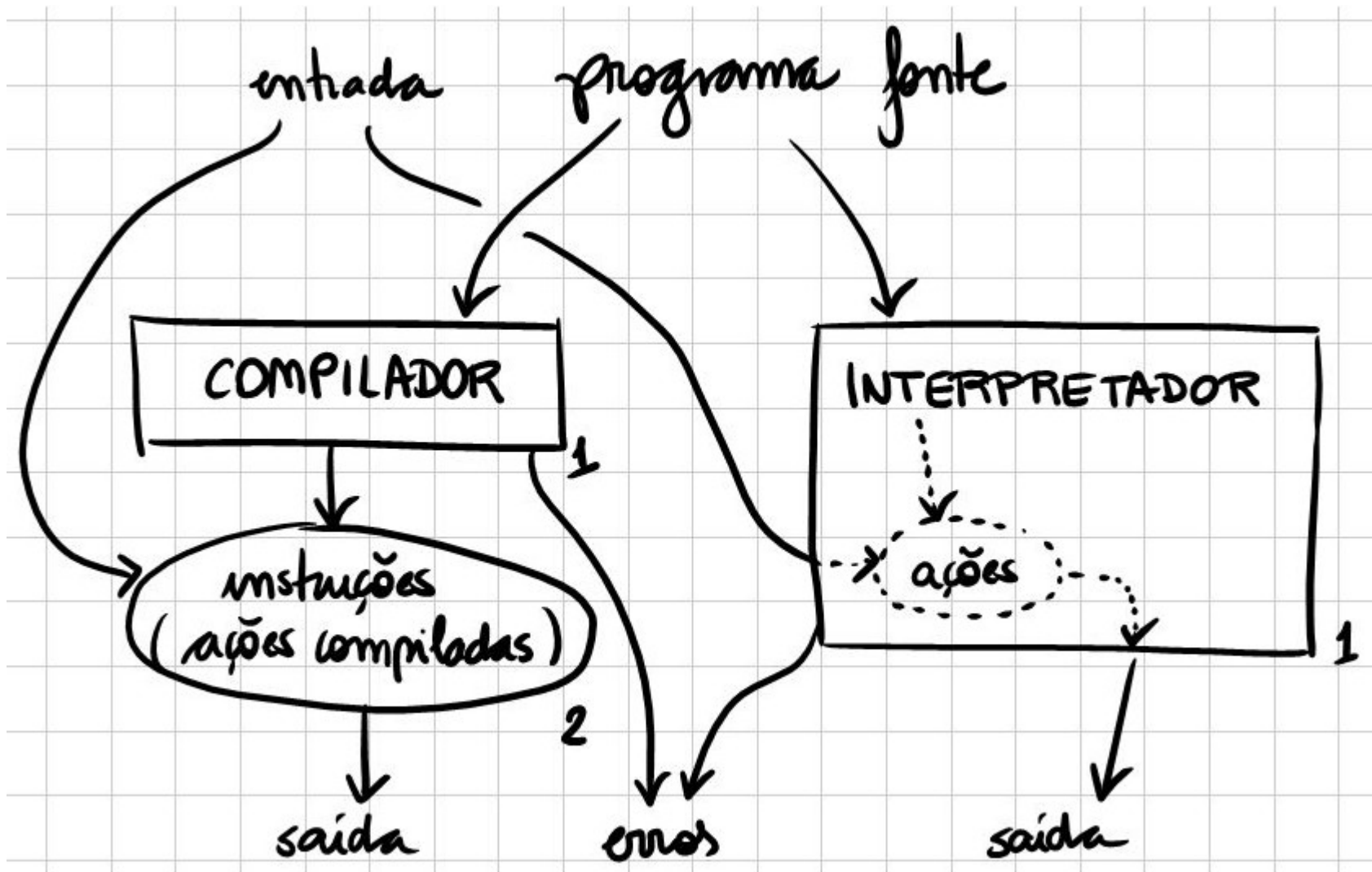




# Compilador X Interpretador X Híbrido



# Compilador X Interpretador



# Java X Python

Java

```
public class HelloWorld
{
    public static void main (String[] args)
    {
        System.out.println("Hello world!");
    }
}
```

Python

```
print "Hello, world!"
```

```
print("Hello, world!") # Python version 3
```

# Java X Python

Java	Python
<pre>int    myCounter = 0; String myString = String.valueOf(myCounter); if (myString.equals("0")) ...</pre>	<pre>myCounter = 0 myString = str(myCounter) if myString == "0": ...</pre>
<pre>// print the integers from 1 to 9 for (int i = 1; i &lt; 10; i++) {     System.out.println(i); }</pre>	<pre>print the integers from 1 to 9 for i in range(1,10):     print i</pre>

# Java X Python

Java

```
import java.io.*;
...

BufferedReader myFile =
    new BufferedReader(
        new FileReader(argFilename));
```

Python

```
# open an input file
myFile = open(argFilename)
```

# Java X Python

## JAVA

```
public class Employee
{
    private String myEmployeeName;
    private int    myTaxDeductions = 1;
    private String myMaritalStatus = "single";

    //----- constructor #1 -----
    public Employee(String EmployeeName)
    {
        this(employeeName, 1);
    }

    //----- constructor #2 -----
    public Employee(String EmployeeName, int taxDeductions)
    {
        this(employeeName, taxDeductions, "single");
    }

    //----- constructor #3 -----
    public Employee(String EmployeeName,
                    int taxDeductions,
                    String maritalStatus)
    {
        this.employeeName = employeeName;
        this.taxDeductions = taxDeductions;
        this.maritalStatus = maritalStatus;
    }
    ...
}
```

## PYTHON

```
class Employee():

    def __init__(self,
                  employeeName
                  , taxDeductions=1
                  , maritalStatus="single"
                  ):

        self.employeeName = employeeName
        self.taxDeductions = taxDeductions
        self.maritalStatus = maritalStatus
    ...
```

In Python, a class has only one constructor.

The constructor method is simply another method of the class, but one that has a special name: `__init__`

# Java X Python

JAVA

```
if ( a > b )  
{  
    a = b;  
    b = c;  
}
```

PYTHON

```
if a > b :  
    a = b  
    b = c
```

# Desvendando programação com PYTHON

- Onde aprender?



# Onde aprender?

**PRIME CURSOS**  
UM NOVO CONCEITO EM ENSINO A DISTÂNCIA

523 alunos online (47) 3037-4150

Qual curso está buscando?

Home Cursos Sobre Como Funciona? Blog Certificado Fale Conosco **Cadastre-se!**

Área do Aluno

Email

Informe seu email

Senha

Informe sua senha

Esqueceu a senha?  
Cadastre-se!

Conectar

Home > Programação > Curso de Programação em Python Online Grátis

## Curso de Programação em Python Online Grátis

 Curtir 222  Tweet  

Este curso é **GRÁTIS**. Aproveite!  
**Totalmente livre de mensalidades.**

 **ASSINAR GRÁTIS\***

**Curso grátis de Programação em Python Básico** - com opção de certificado válido!



**CURSO GRÁTIS DE PYTHON**

**EM VIDEOAULAS!**

**Com certificado válido!**

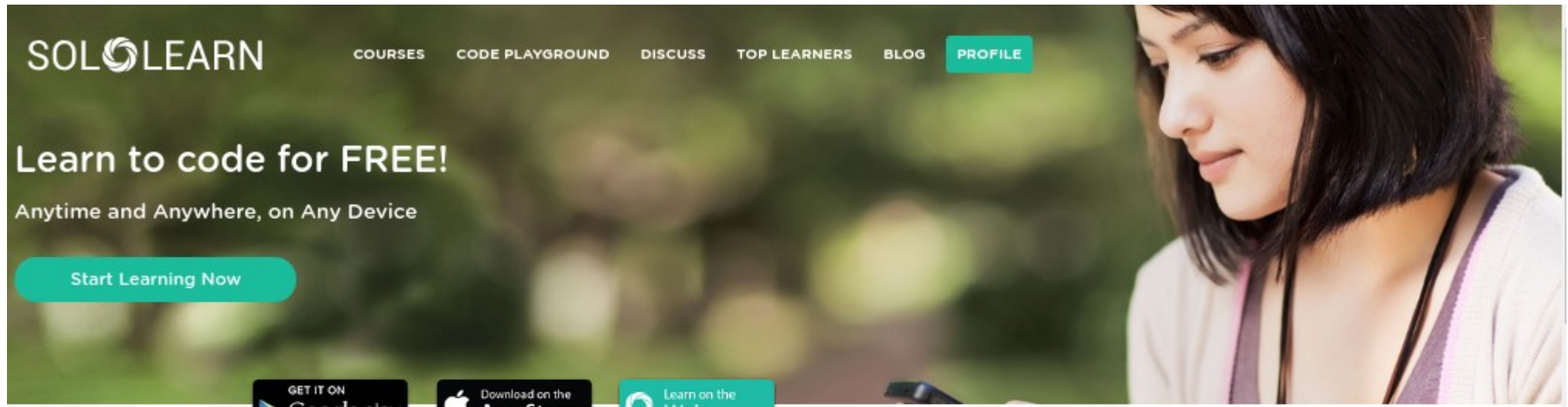
**WWW.PRIMECURSOS.COM.BR**

Áreas de Cursos

- ▶ Administração
- ▶ Comunicação e Vendas
- ▶ Concursos Públicos
- ▶ Contabilidade e Finanças
- ▶ Direito
- ▶ Educação
- ▶ Enem
- ▶ Gastronomia e Confeitaria
- ▶ Idiomas
- ▶ Informática
- ▶ Meio Ambiente

Uma linguagem de **programação** é um conjunto de regras e instruções que, ao ser escrita dentro de uma determinada sintaxe, poderá ser interpretada pelo computador e dessa forma poderemos dar comandos a ele. Desde os comandos mais simples como somar 10 e 20 até os mais complexos, como desenvolvimento de **aplicativos para celular**, por exemplo.

# Onde aprender?



**SOLOLEARN**


[COURSES](#) [CODE PLAYGROUND](#) [DISCUSS](#) [TOP LEARNERS](#) [BLOG](#) [PROFILE](#)

## Learn to code for FREE!

Anytime and Anywhere, on Any Device

[Start Learning Now](#)


[GET IT ON Google play](#) [Download on the App Store](#) [Learn on the Web](#)



### C++ Tutorial

Our C++ tutorial covers basic concepts, data types, arrays, pointers, conditional statements, loops, functions, classes, objects, inheritance, and polymorphism.


[Take This Course](#)



### Python 3 Tutorial

Learn Python, one of today's most in-demand programming languages on-the-go! Practice writing Python code, collect points, & show off your skills now!

[Take This Course](#)



### Java Tutorial

With our interactive Java course, you'll learn object-oriented Java programming and have the ability to write clear and valid code in almost no time at all.

[Take This Course](#)

# Onde aprender?





# Onde aprender?

codecademy

Community

Catalog

Login



Join the Millions  
Learning to Code  
with Codecademy



## Get Started For Free

Username can't be blank

Email

Password

Start coding now

This page is protected by reCAPTCHA, and subject to Google's [Privacy Policy](#) & [Terms of Service](#). By signing up you agree to Codecademy's [Terms of Service](#).

# Onde aprender?

## AQUI!

FACULDADE DE TECNOLOGIA

**Fatec**  
Franca  
Dr. Thomaz Novelino



[Início](#) [A Fatec Franca](#) [Cursos](#) [VESTIBULAR](#) [Extensão](#) [Estudante](#) [Docente](#) [Biblioteca](#) [Eventos](#) [Concurso Público](#) [Revista EduFatec](#) [Vagas Remanescentes](#)



Cursos Superiores de Tecnologia



Análise e  
Desenvolvimento  
de Sistemas




Gestão da  
Produção  
Industrial



novo!  
Gestão de  
Recursos  
Humanos

# Onde treinar?


## URI ONLINE JUDGE - Problems & Contests

 ENGLISH

LOGINREGISTERFORUMCONTESTSPROBLEMSRANKSFOR PROFESSORSFOR 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**



**SIGN IN**

EMAIL

PASSWORD

☐ REMEMBER ME (7 DAYS)

**SIGN IN**


FACEBOOKGOOGLEGITLABGITHUBBITBUCKETTWITCH

By signing up with a social platform you AGREE with the


### 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**




### PROBLEM REPOSITORY



The URI Online Judge contains more than 1000 problems divided in 8 big categories. This division help the users to focus on

### URI ONLINE JUDGE ACADEMIC

The URI Online Judge Academic is an unique module for professors and team coaches. Here you can create disciplines and



© 2011 - 2021 URI Online Judge

Cookies | Privacy | Terms & Conditions | FAQs | Status | Credits | Contact

Version 6.0.0

# Sobre

Engenheiro Eletrônico (1983)  
Cientista da Computação (1984)

Esp Des. Software para WEB (2005)

Mestre em Desenvolvimento Regional (2011)

Claretinas (2006) e ETEC (2008)

FATEC Franca (2010) e Uni-FACEF (2011)

Fotógrafo, coralista, marido, pai

Avô do Bento



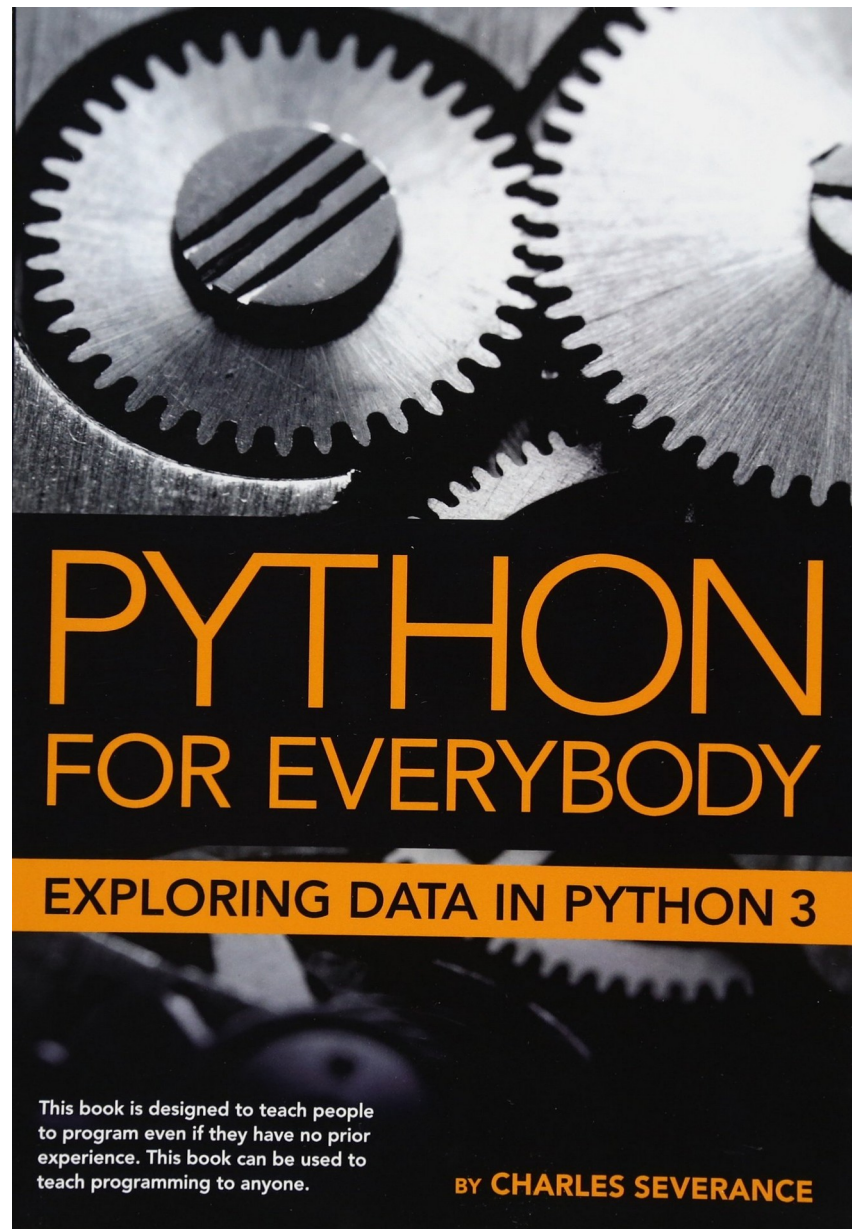
[roland@facef.br](mailto:roland@facef.br)

[carlos.roland@fatec.sp.gov.br](mailto:carlos.roland@fatec.sp.gov.br)

<https://www.facebook.com/carlos.e.roland.3>

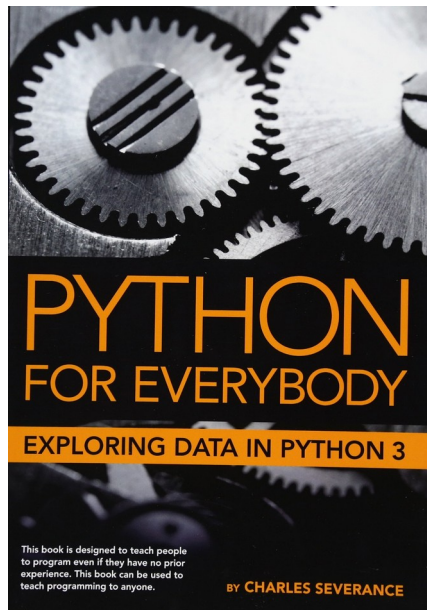


# Referência





# Referência



[http://do1.dr-chuck.com/pythonlearn/EN\\_us/pythonlearn.pdf](http://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf)

Time for  
a Break

# Escrevendo um programa

- Acessar repl.it e criar uma conta
- 
- Abrir Python3
- 
- Testar comandos

# Testando comandos

```
>print("Hello world!")
```

```
>print('Hello world!')
```

```
>print("Hello world!")
```

```
>print('Hello world!')
```

```
>print 'Hello world!'
```

```
>I hate you Python
```

```
>if you come out of there, I would teach you a lesson
```

```
>1024 + 512
```

```
>512 / 8
```

# 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 20  
$
```

# Blocos construtivos de um programa

<b>Bloco</b>	<b>Função</b>
<b>entrada</b>	Obter dados do mundo externo. Teclado, tela de toque, arquivos digitais, microfones, sensores, GPS, etc.
<b>saída</b>	Mostrar resultados em tela, armazenar em arquivos digitais, autofalantes, atuadores, etc.
<b>execução sequencial</b>	Executar comandos um após o outro na ordem que foram escritos no programa.
<b>execução condicional</b>	Testar certas condições e executar ou pular uma sequência de comandos.
<b>execução repetitiva</b>	Executar uma sequência de comandos repetidamente, usualmente com alguma variação.
<b>reuso</b>	Escrever um conjunto de instruções, dar ao conjunto um nome e então reusar o conjunto quando necessário no programa.



# O que pode dar errado?

<b>Tipos de erros</b>	<b>Causas</b>
<b>sintaxe</b>	Os primeiros a ocorrer e os mais fáceis de corrigir. Violação de regra gramatical. Python sinaliza onde ocorreu o erro, mas pode ser algumas linhas acima.
<b>lógica</b>	Scripts sintaticamente corretos, mas em sequência errada, provocando erros de execução. É necessário rever e corrigir o algoritmo.
<b>semântica</b>	Muito comuns e os mais difíceis de identificar e corrigir. Podem ficar ocultos até que uma condição diferente de dados ou condições ocorra (que não foram previstos inicialmente), ou por mudanças de regras do processo. Testes de validação nunca são feitos pelo desenvolvedor. De preferência pelo usuário que relatou os requisitos.

# Depurando

## Ações

## O que fazer

### leia

Examine o código, releia algumas vezes, e verifique se a ideia escrita é a desejada pela história.

### execute

Execute o script para diferentes condições de dados e contextos. Mostre resultados parciais antes e depois de cada bloco de instruções, especialmente desvios e repetições.

### rumine

Invista tempo em análise! Identifique o tipo de erro: sintaxe, lógica, semântica? Leia detalhadamente as mensagens de erros e os resultados apresentados. Que tipo de erro provoca o problema encontrado? O que foi mudado desde a última vez que o script rodou e agora?

### recue

Volte a versões anteriores do código desfazendo mudanças recentes até um ponto que execute corretamente e que se entenda o que está acontecendo. Então refaça as modificações necessárias. Use versionador de código (Git com GitLab, Bitbucket, ou GitHub).

# Variáveis, expressões e comandos

Valores e tipos

Variáveis

Nomes de variáveis e palavras-chave

Comandos

Operadores e operandos

Expressões

Ordem de operações

Operador módulo

Operações com cadeias de caracteres

Pedindo dados para o usuário

Comentários

Escolhendo nomes mnemônicos para variáveis

# Valores e tipos

## Valores

1, 58, "Hello world!", 'Digite seu nome: ', 3.14159265,  
True, False

## Tipos

inteiro (integer), real (float), cadeia (string), lógico (bool)

```
>>>type(1)  
<class 'int'>
```

```
>>>type("R$ 35.330,00")  
<class 'str'>
```

```
>>>type(35330.00)  
<class 'float'>
```

```
>>>type(True)  
<class 'bool'>
```

## Quiz

Execute o comando:

```
>>>print(1,000,000.00)
```

Qual o resultado? Porque? Qual o tipo de erro?



*That's all Folks!*