Desvendando programação com Python

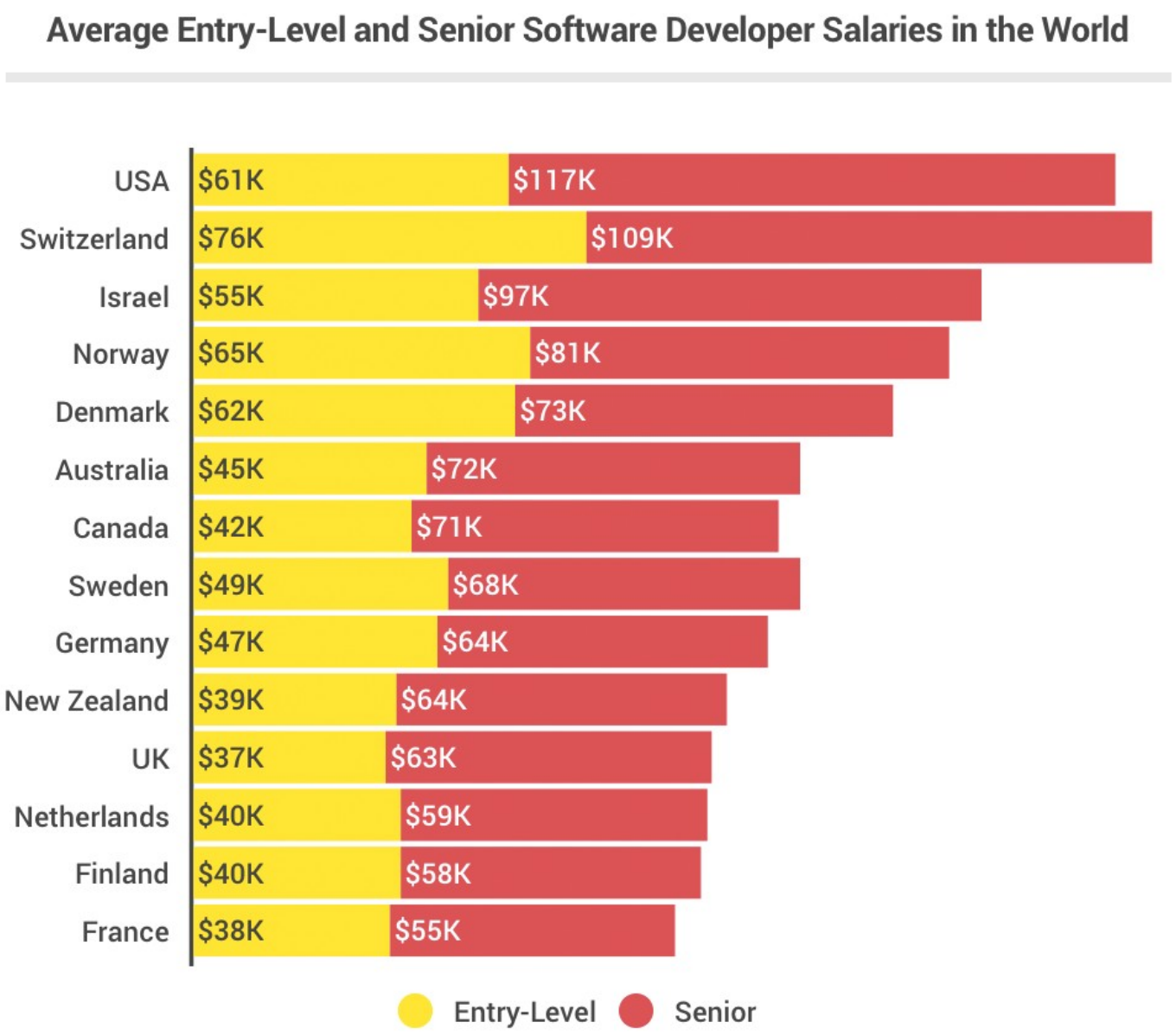
 Carlos ROLAND

python™

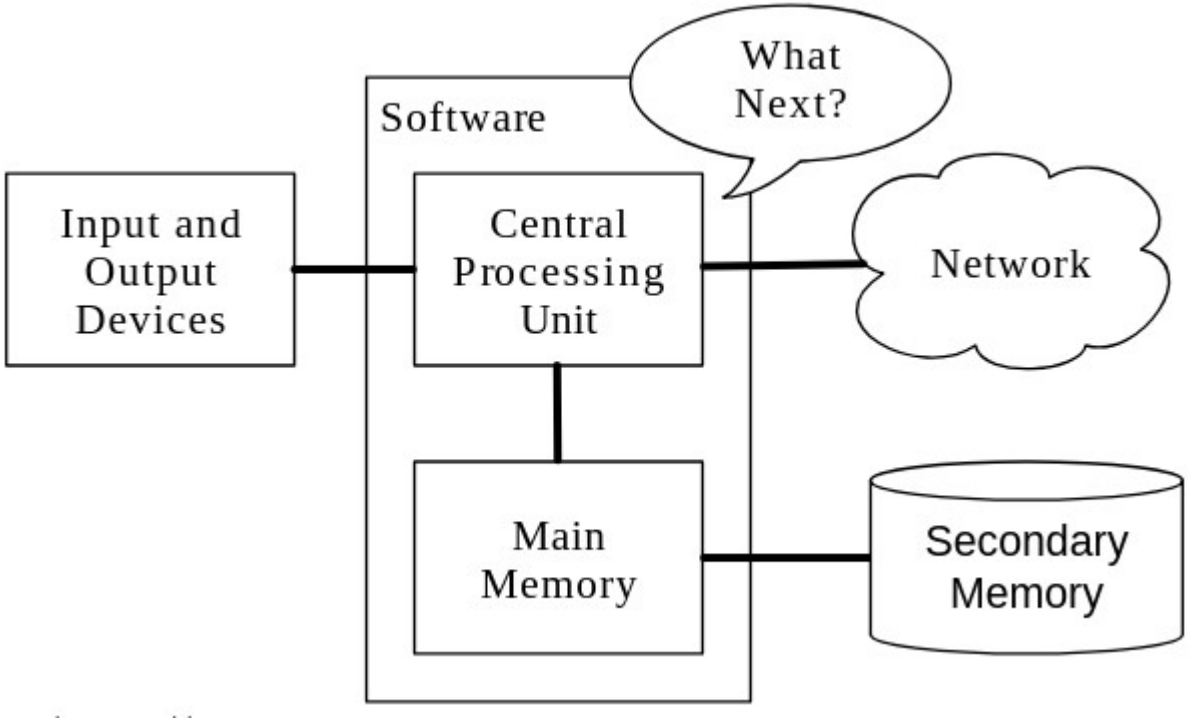
Desvendando programação

Porque aprender programação

Motivação

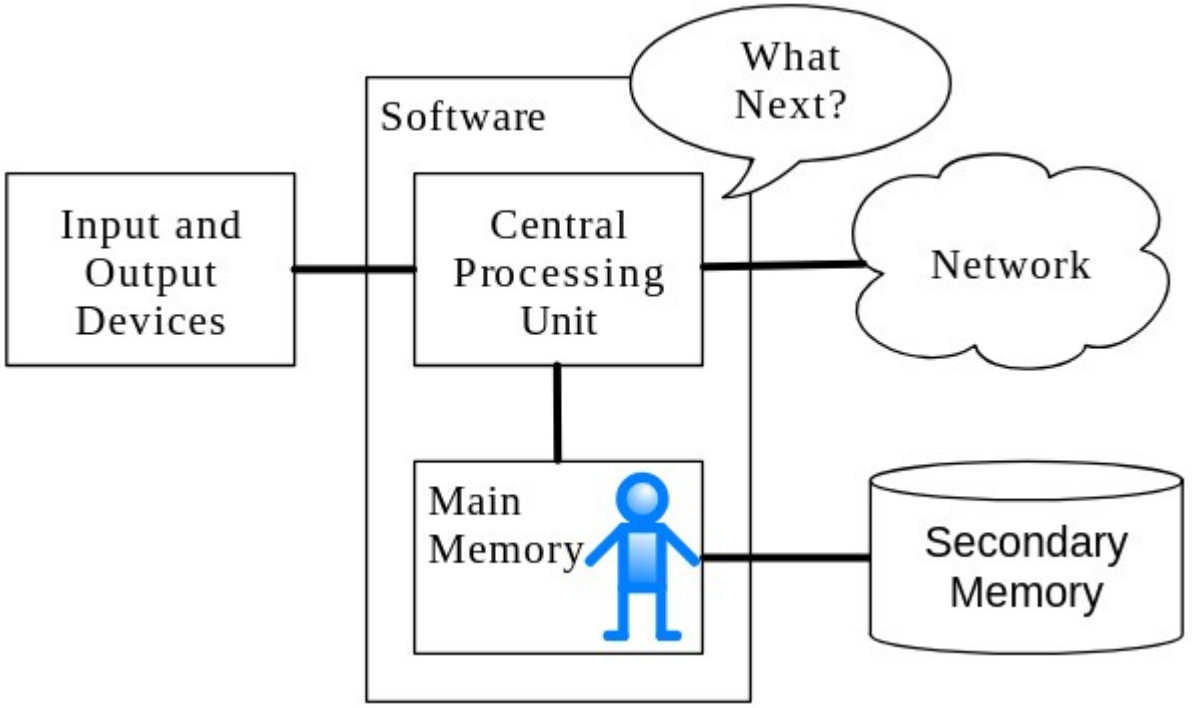

Average Entry-Level and Senior Software Developer Salaries in the World
USA
$61K
$117K
Switzerland
$76K
$109K
Israel
$55K
$97K
Norway
$65K
$81K
Denmark
$62K
$73K
Australia
$45K
$72K
Canada
$42K
$71K
Sweden
$49K
$68K
Germany
$47K
$64K
New Zealand
$39K
$64K
UK
$37K
$63K
Netherlands
$40K
$59K
Finland
$40K
$58K
France
$38K
$55K
Entry-Level
Senior
Source: daxx.com

Arquitetura de computadores



Fonte: https://www.py4e.com/html3/

Arquitetura de computadores



Fonte: https://www.py4e.com/html3/

Desvendando programação

Habilidades para programação

Identificar um problema de análise de dados ou informação

e

desenvolver um programa computacional para resolvê-lo

Habilidades para programação

Conhecer uma linguagem de programação
vocabulário e gramática;

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 estória

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
execept
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

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

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

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

```
>> import this

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to d
Although that way may not be obvious at first unless you're Dutc
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
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 outstandings 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

Fonte: https://blog.tunga.io/the-top-17-most-in-demand-software-skills-for-2018-f37df6c0915b

Motivação

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

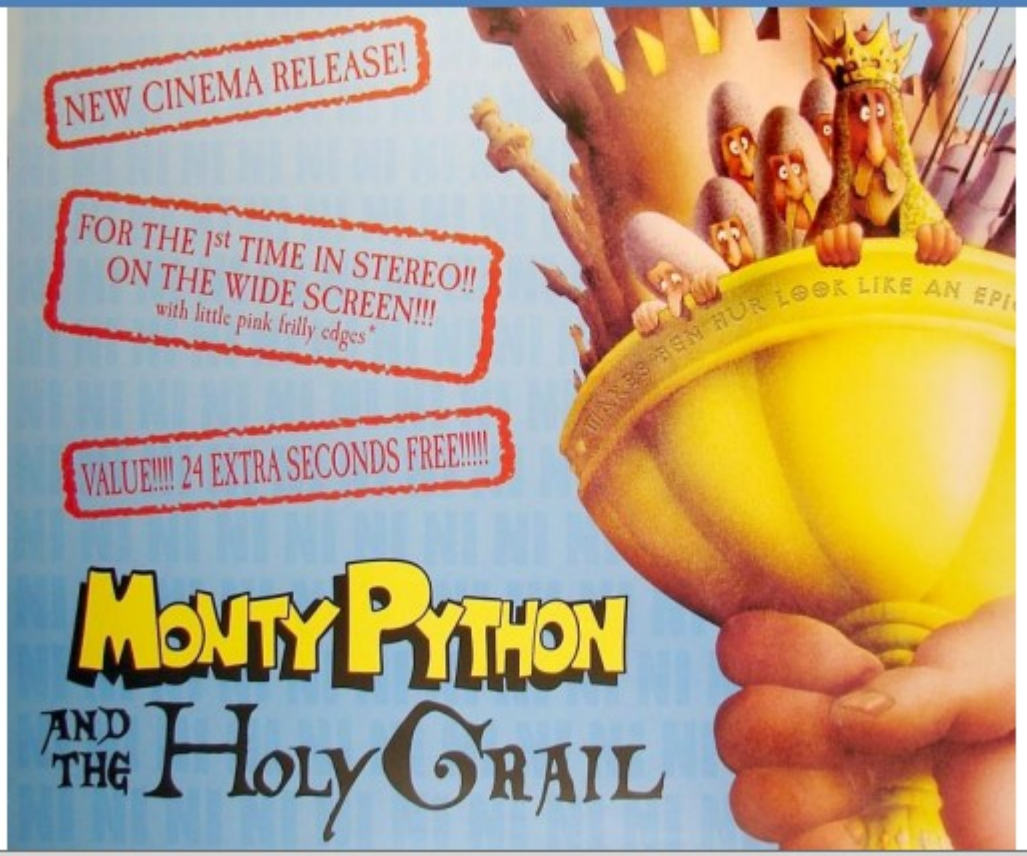
Whereas Ruby might not be among the most advertised skills, it is definitely among the relatively most scarce skills. When looking at the salaries offered for developers, Ruby tops the list:

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.

Fonte: https://blog.tunga.io/the-top-17-most-in-demand-software-skills-for-2018-f37df6c0915b

Origem do nome Python



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

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

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

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

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

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

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

Python: quem usa

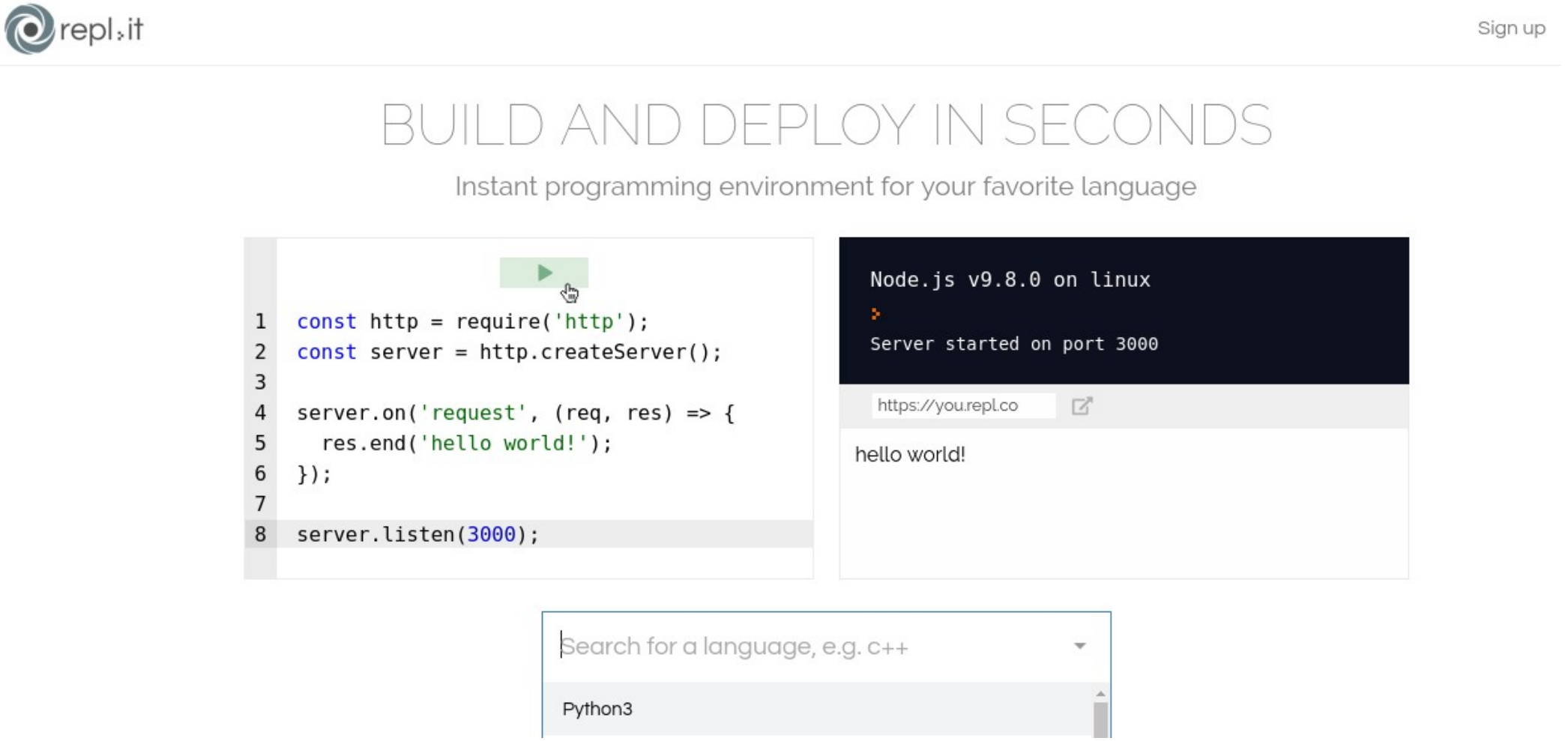
Você sabia que o “Pi” do nome se origina de Python?

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

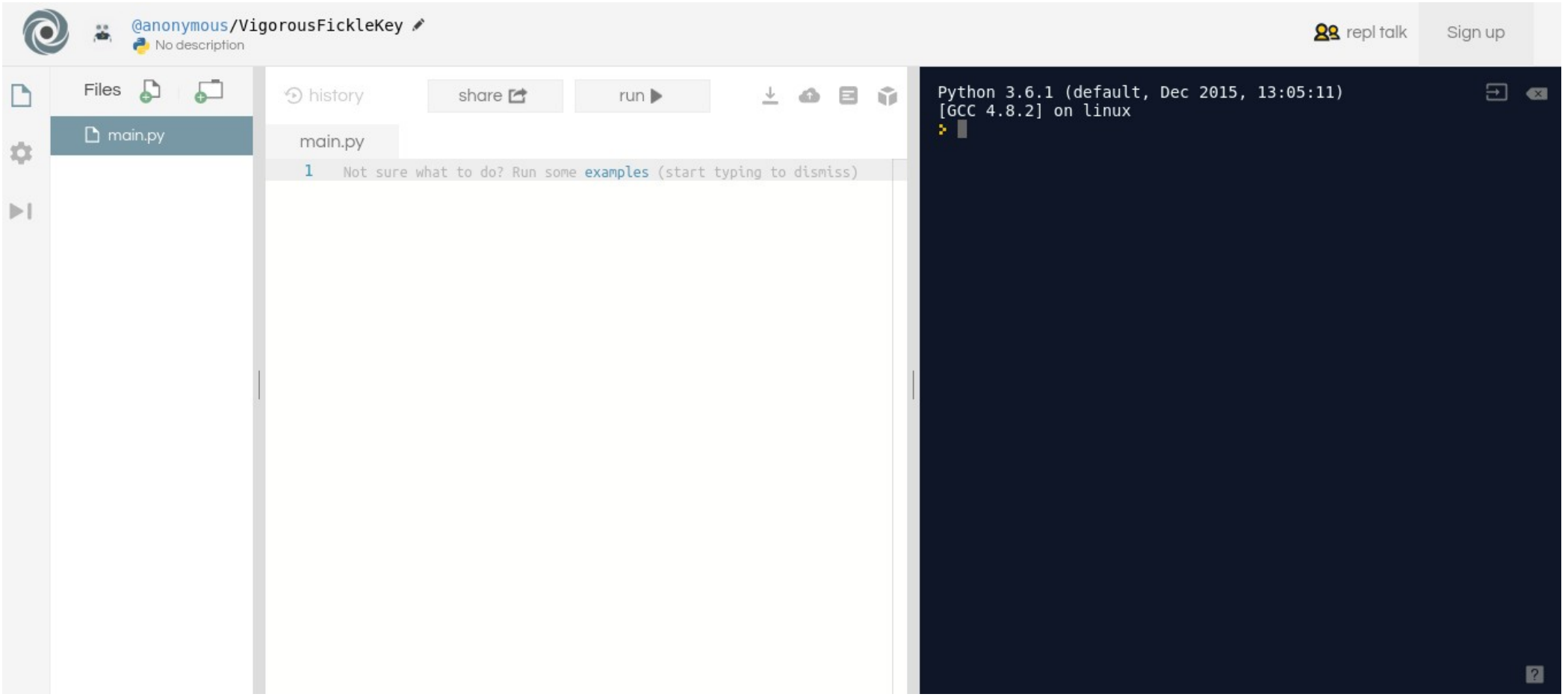
Fonte: https://pt.slideshare.net/fmasanori/twp00-apresentao-e-motivao

Python: quem usa

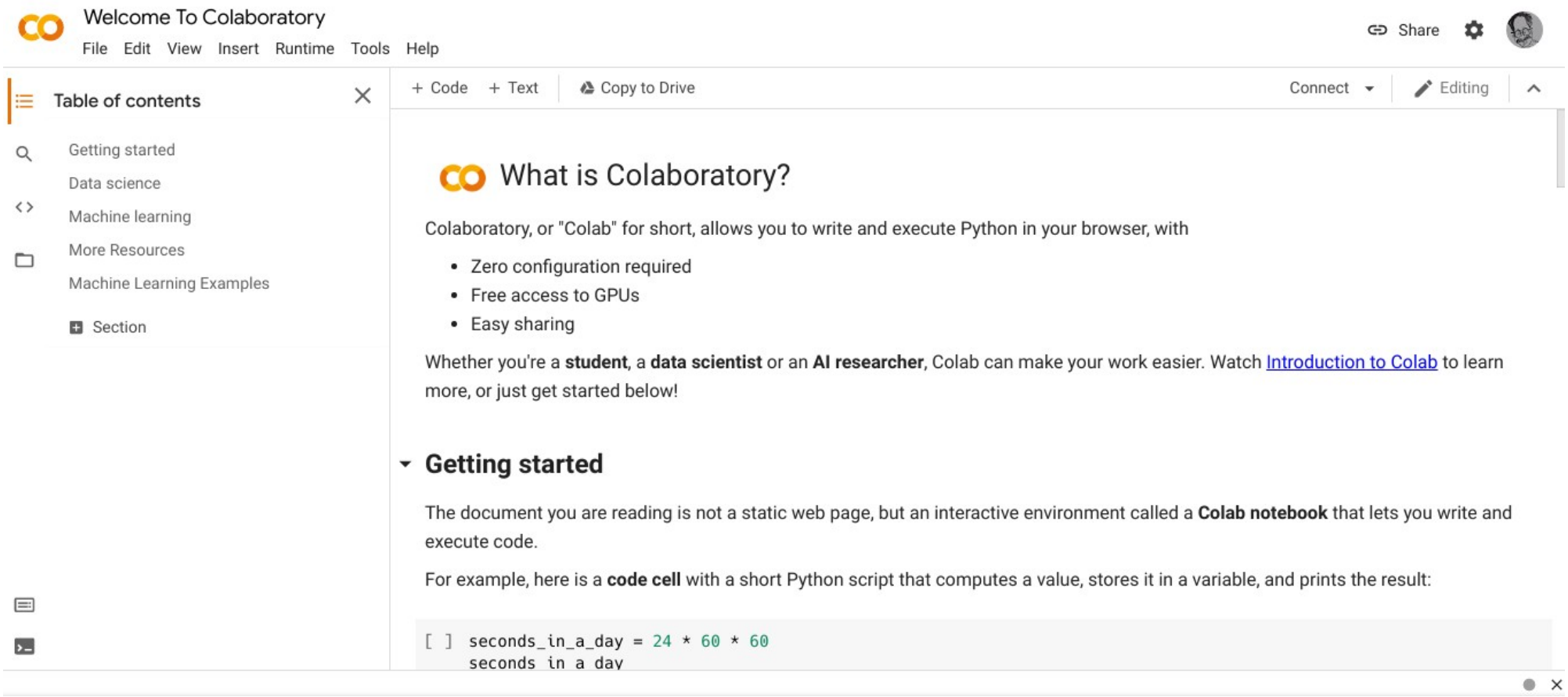
Conversando com Python – repl.it



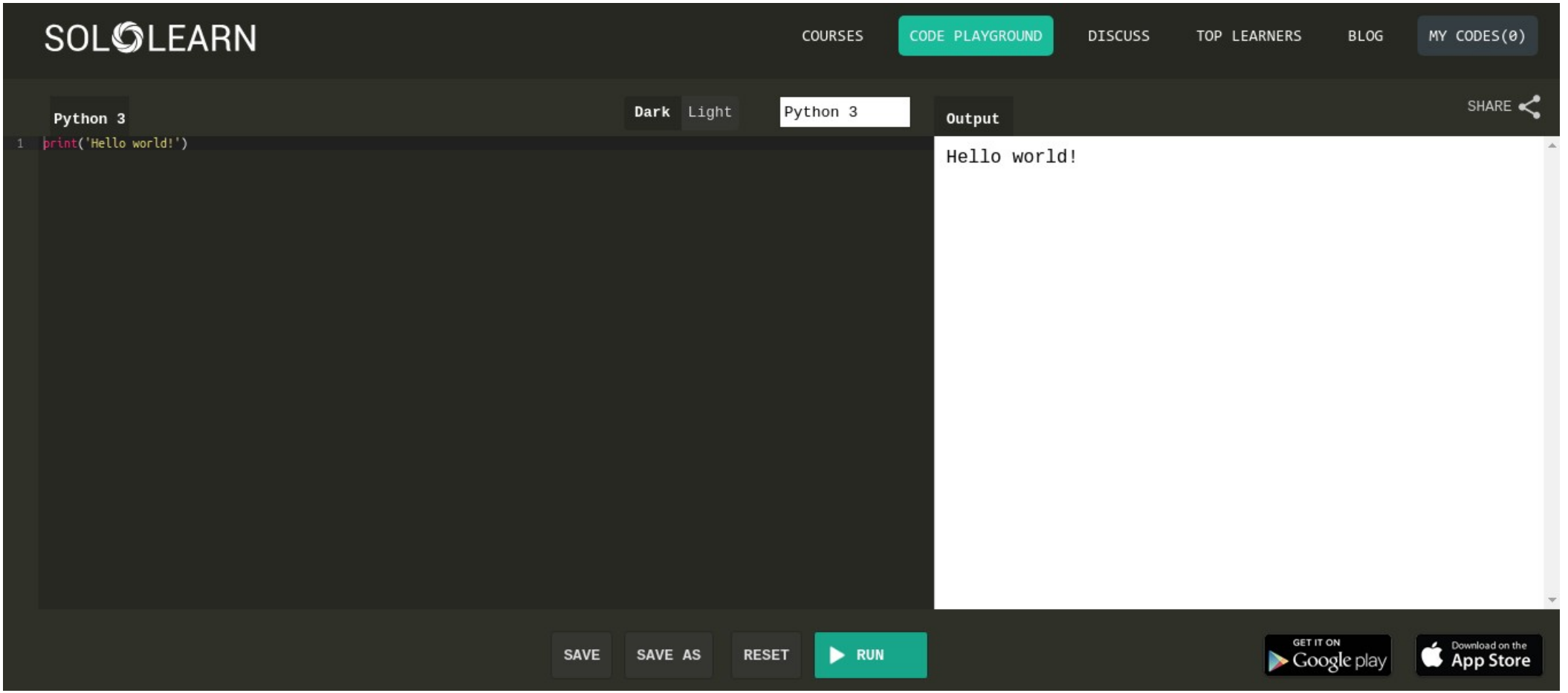
Conversando com Python – repl.it



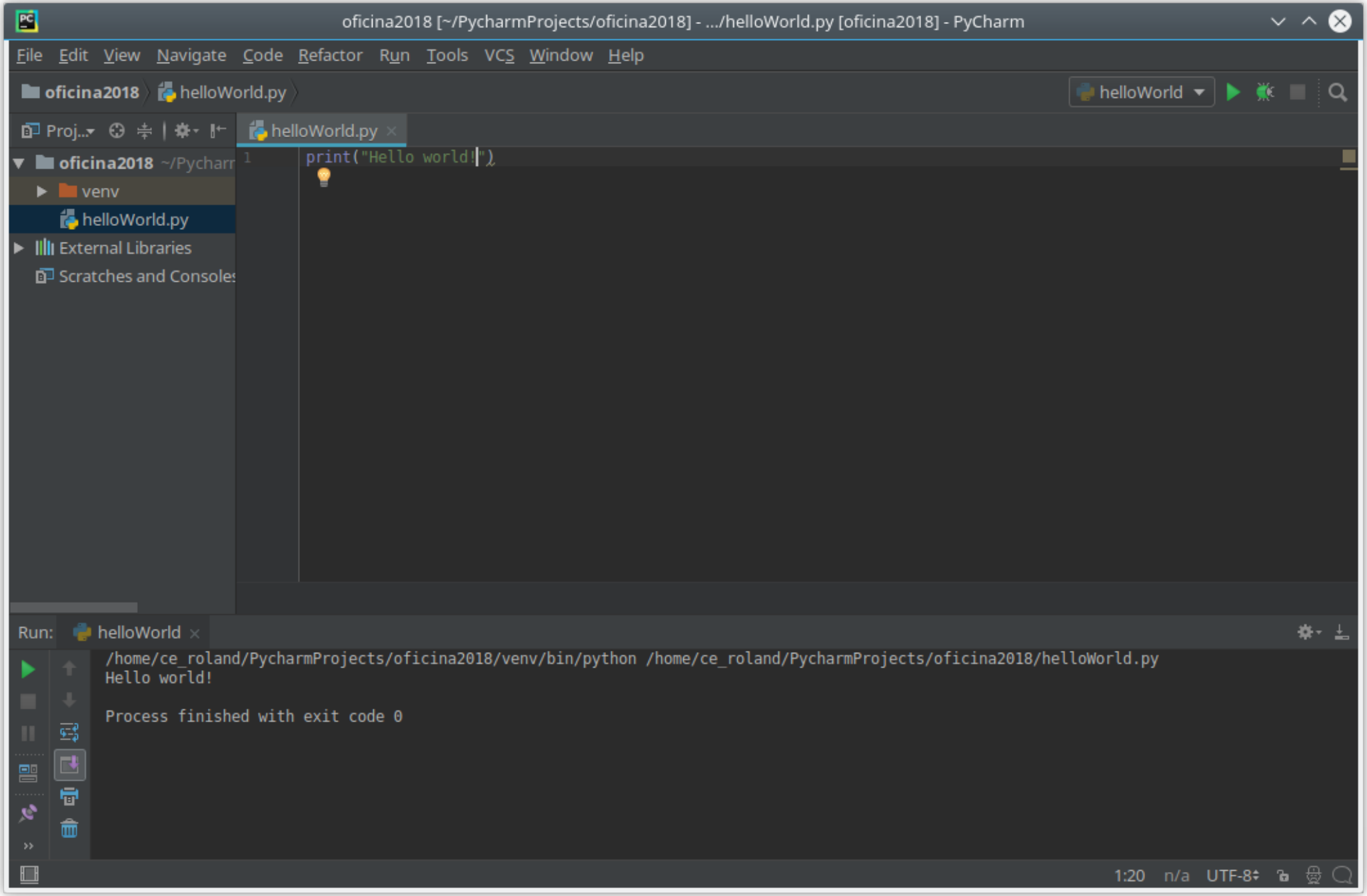
Conversando com Python – COLAB



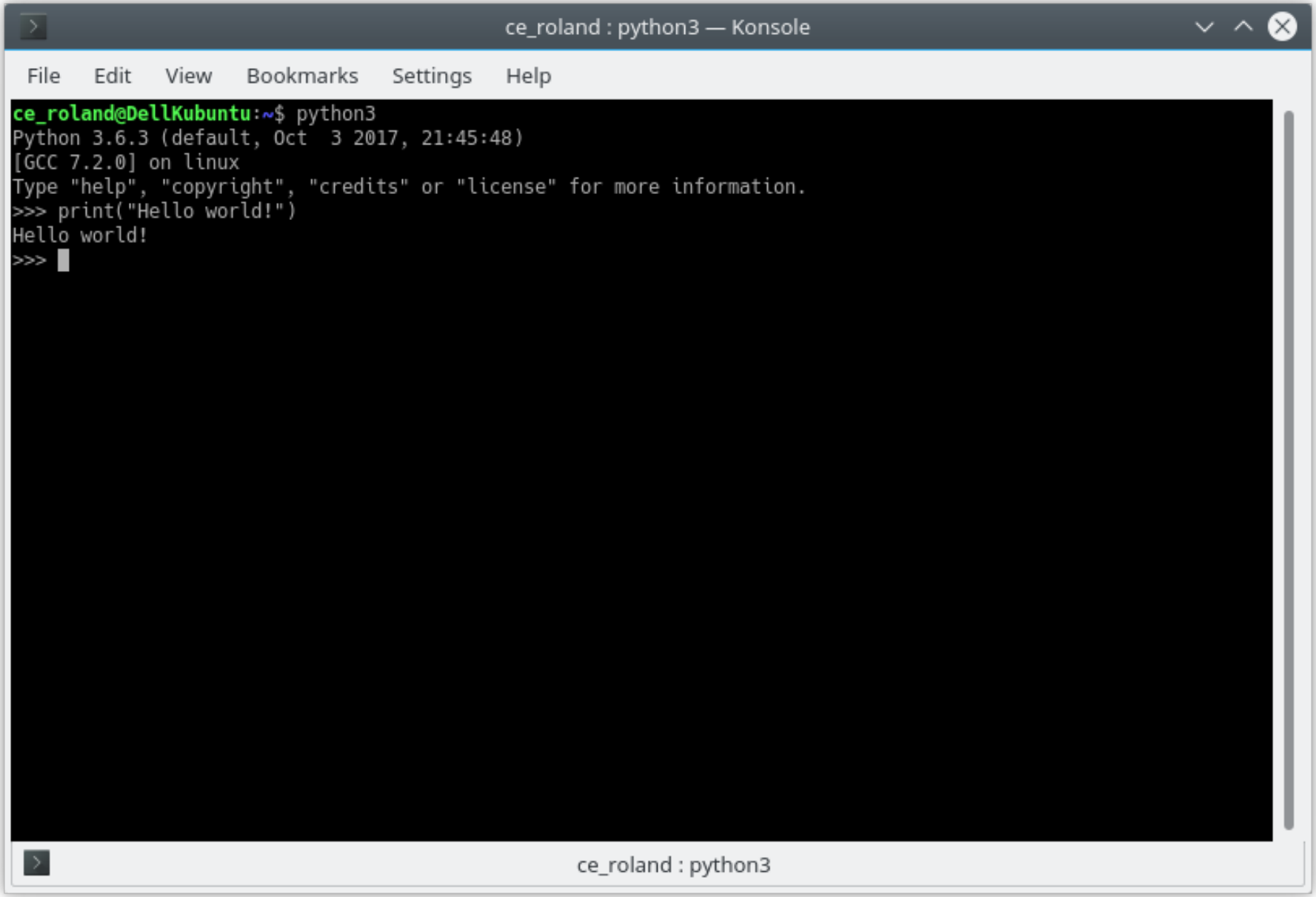
Conversando com Python - Sololearn



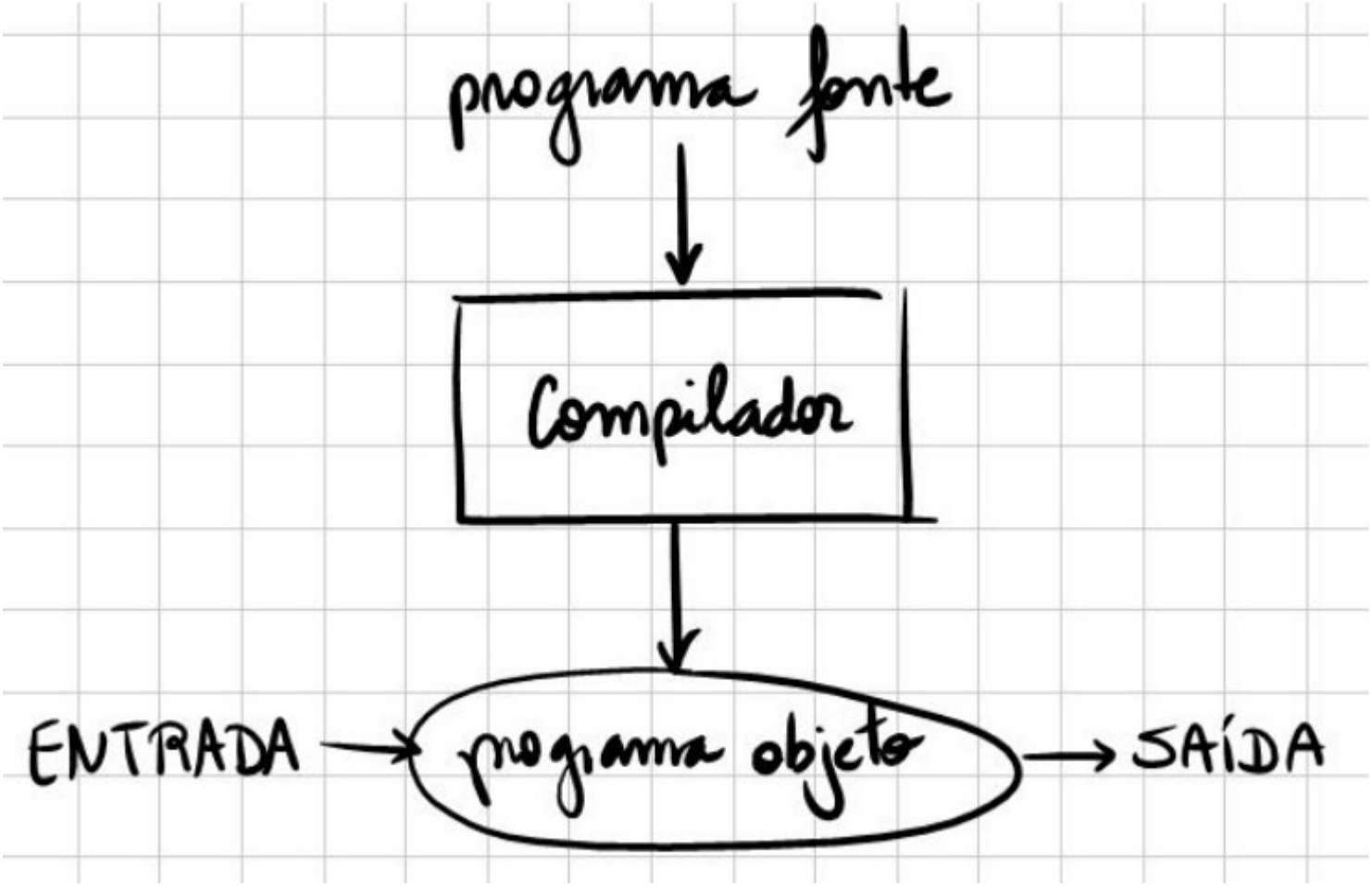
Conversando com Python - pyCharm



Conversando com Python - terminal

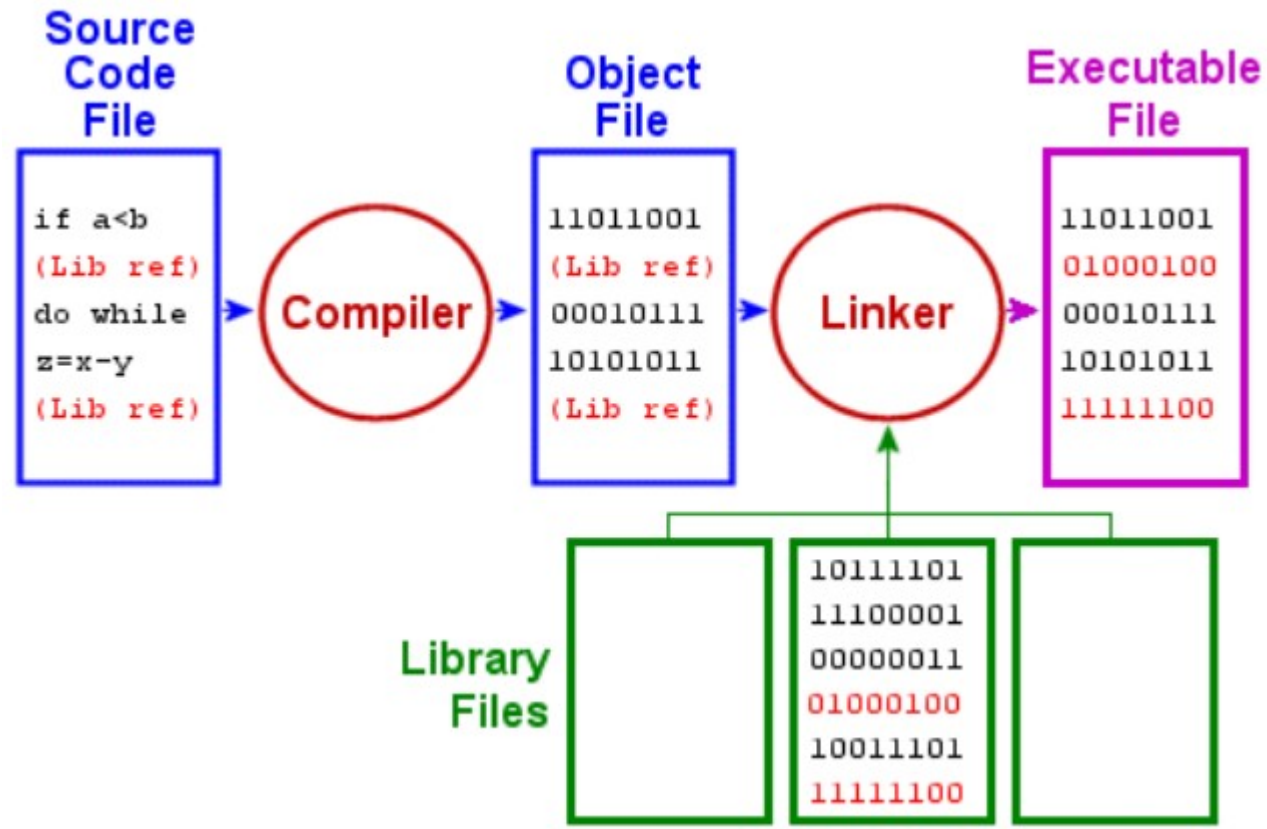


Compilador X Interpretador X Híbrido



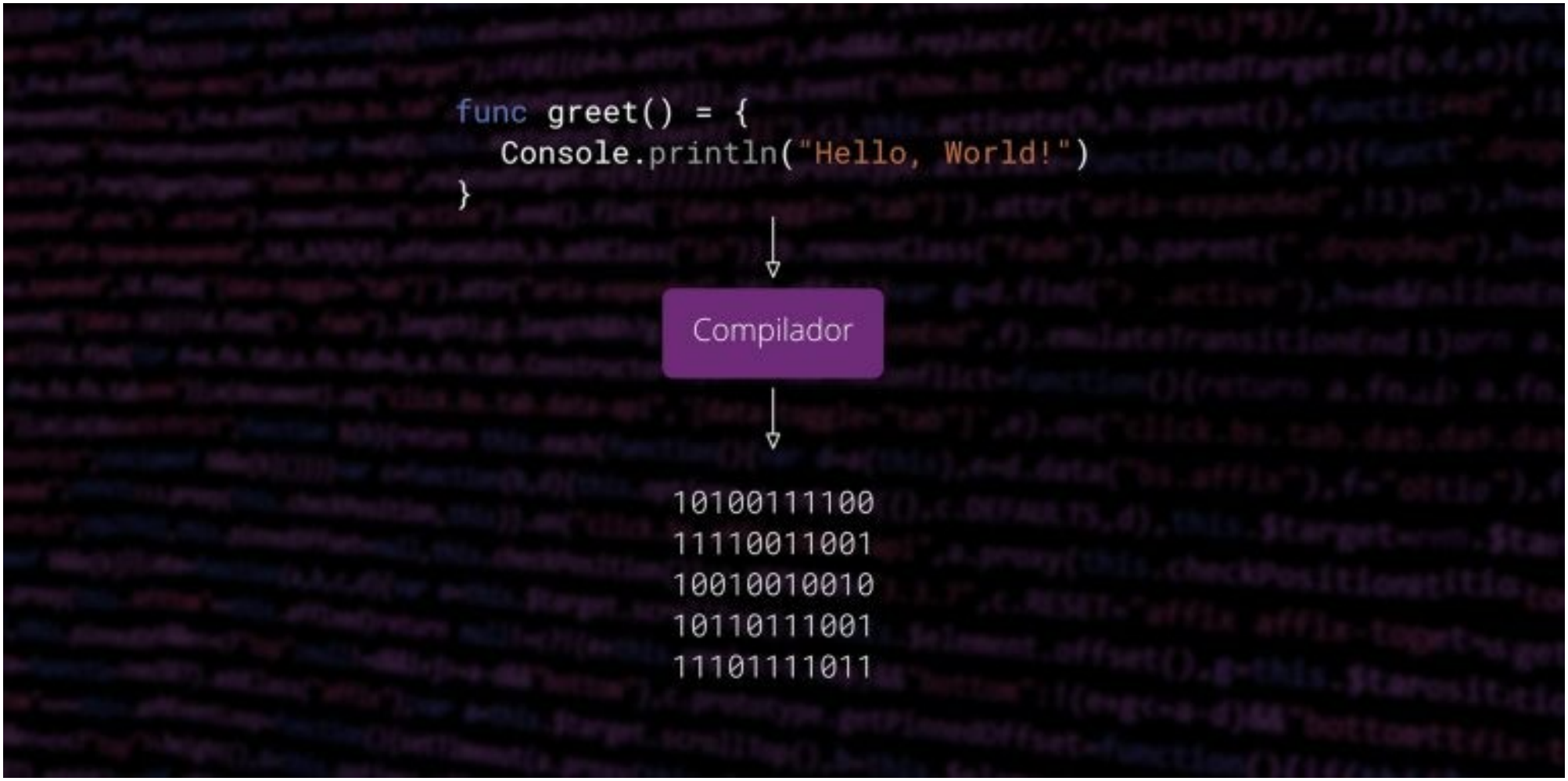
Fonte: http://www.ic.unicamp.br/~felice/ensino/1s2018/compiladores1/Topico01.Introducao.Slides.pdf

Compilador X Interpretador X Híbrido

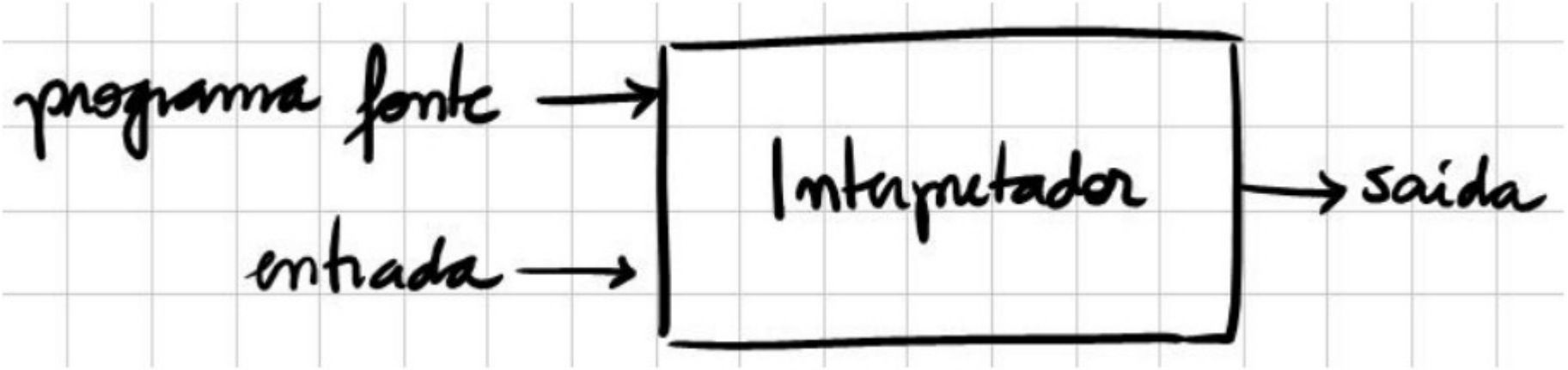


Fonte: http://edirlei.3dgb.com.br/aulas/clp/CLP_Aula_01_Introducao_2015.pdf

Compilador X Interpretador X Híbrido

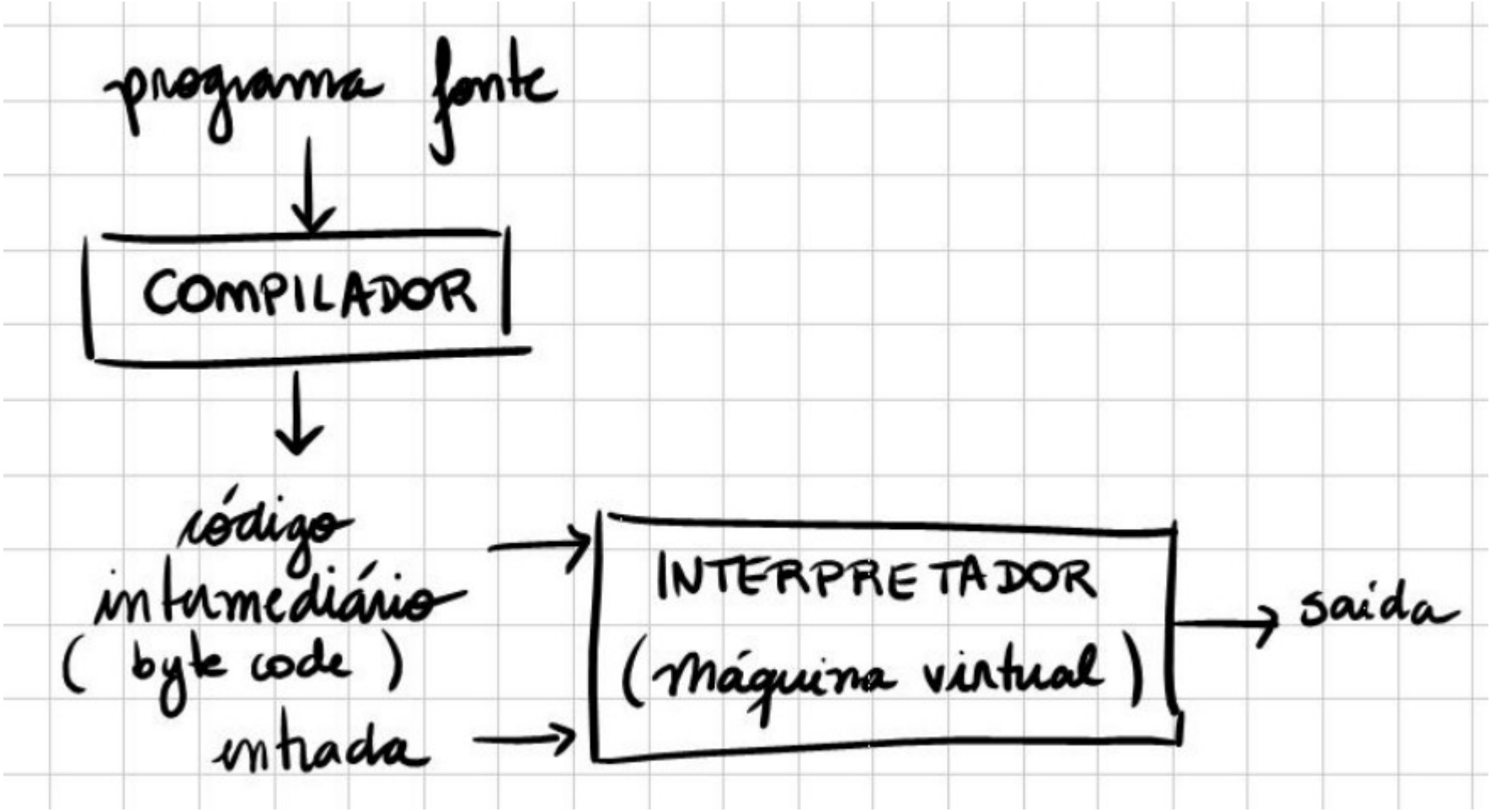


Compilador X Interpretador X Híbrido



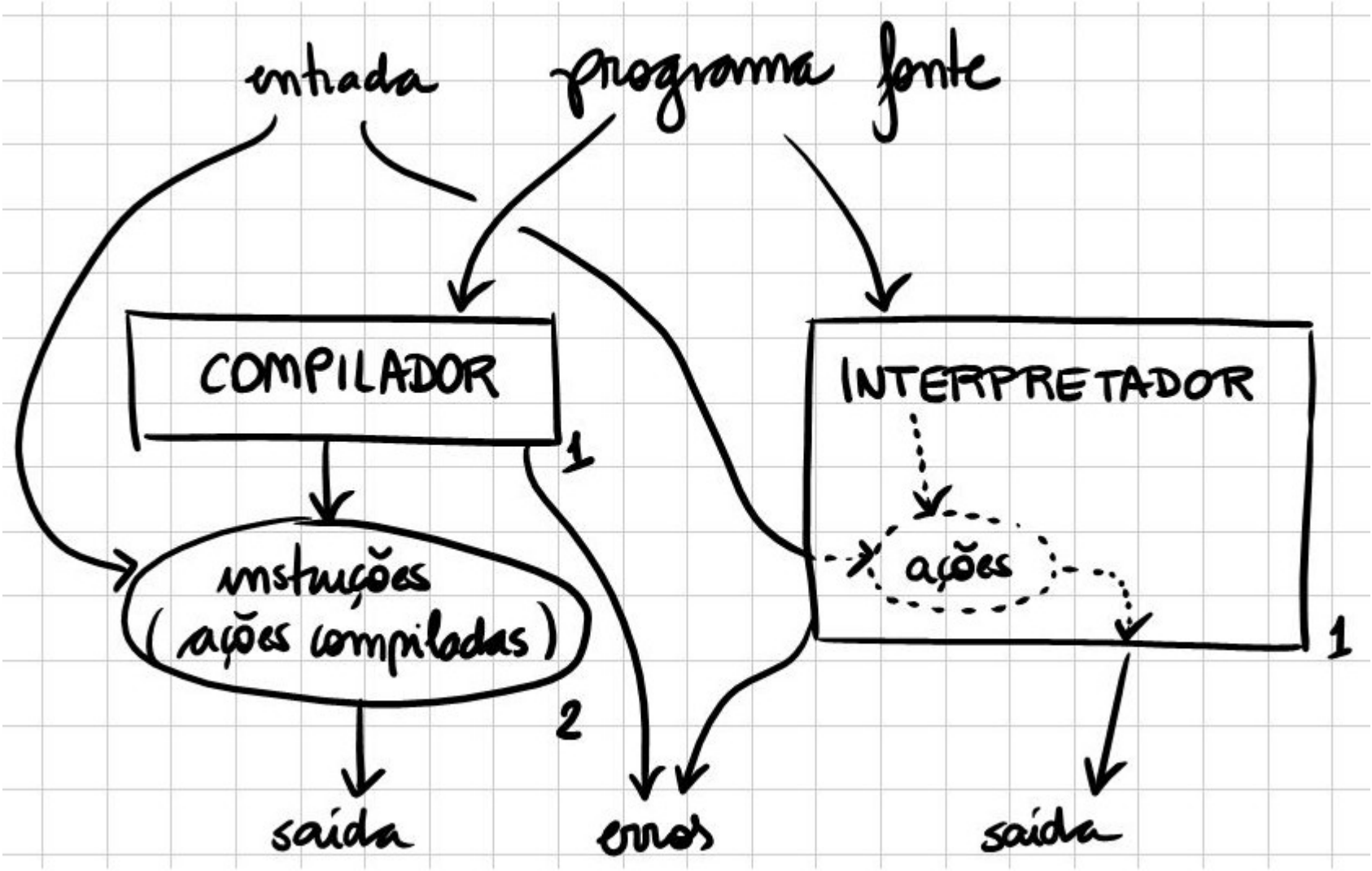
Fonte: http://www.ic.unicamp.br/~felice/ensino/1s2018/compiladores1/Topico01.Introducao.Slides.pdf

Compilador X Interpretador X Híbrido



Fonte: http://www.ic.unicamp.br/~felice/ensino/1s2018/compiladores1/Topico01.Introducao.Slides.pdf

Compilador X Interpretador



Fonte: http://www.ic.unicamp.br/~felice/ensino/1s2018/compiladores1/Topico01.Introducao.Slides.pdf

Java X Python

Java

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

Python

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

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

Fonte: https://pythonconquerstheuniverse.wordpress.com/2009/10/03/python-java-a-side-by-side-comparison/

Java X Python

Java

```
int    myCounter = 0;
String myString = String.valueOf(myCounter);
if (myString.equals("0")) ...
```

Python

```
myCounter = 0
myString = str(myCounter)
if myString == "0": ...
```

```
// print the integers from 1 to 9
for (int i = 1; i < 10; i++)
{
   System.out.println(i);
}
```

```
 print the integers from 1 to 9
for i in range(1,10):
    print i
```

Fonte: https://pythonconquerstheuniverse.wordpress.com/2009/10/03/python-java-a-side-by-side-comparison/

Java X Python

Java

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

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

Python

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

Fonte: https://pythonconquerstheuniverse.wordpress.com/2009/10/03/python-java-a-side-by-side-comparison/

Java X Python

JAVA

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

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

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

   //--------- constructor #3 -------------
   public Employee(String EmployeName,
         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__

Fonte: https://pythonconquerstheuniverse.wordpress.com/2009/10/03/python-java-a-side-by-side-comparison/

Java X Python

JAVA

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

PYTHON

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

Fonte: https://pythonconquerstheuniverse.wordpress.com/2009/10/03/python-java-a-side-by-side-comparison/

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

Áreas de Cursos

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

Home > Programação > Curso de Programação em Python Online Grátis

Curso de Programação em Python Online Grátis

Curtir 222 | Tweet | G+

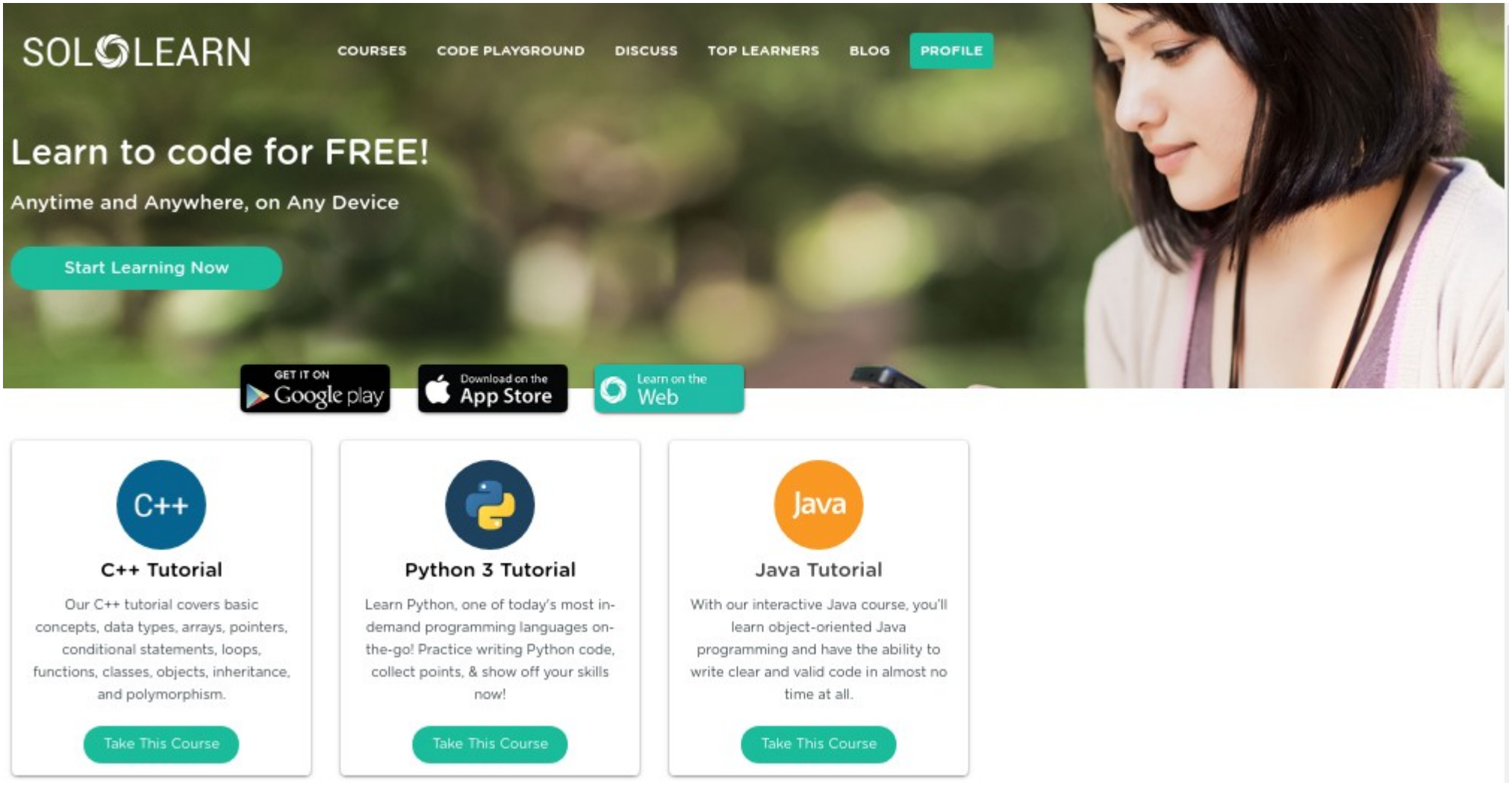
Este curso é **GRÁTIS**. Aproveite!.
Totalmente livre de mensalidades.

ASSINAR GRÁTIS*

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

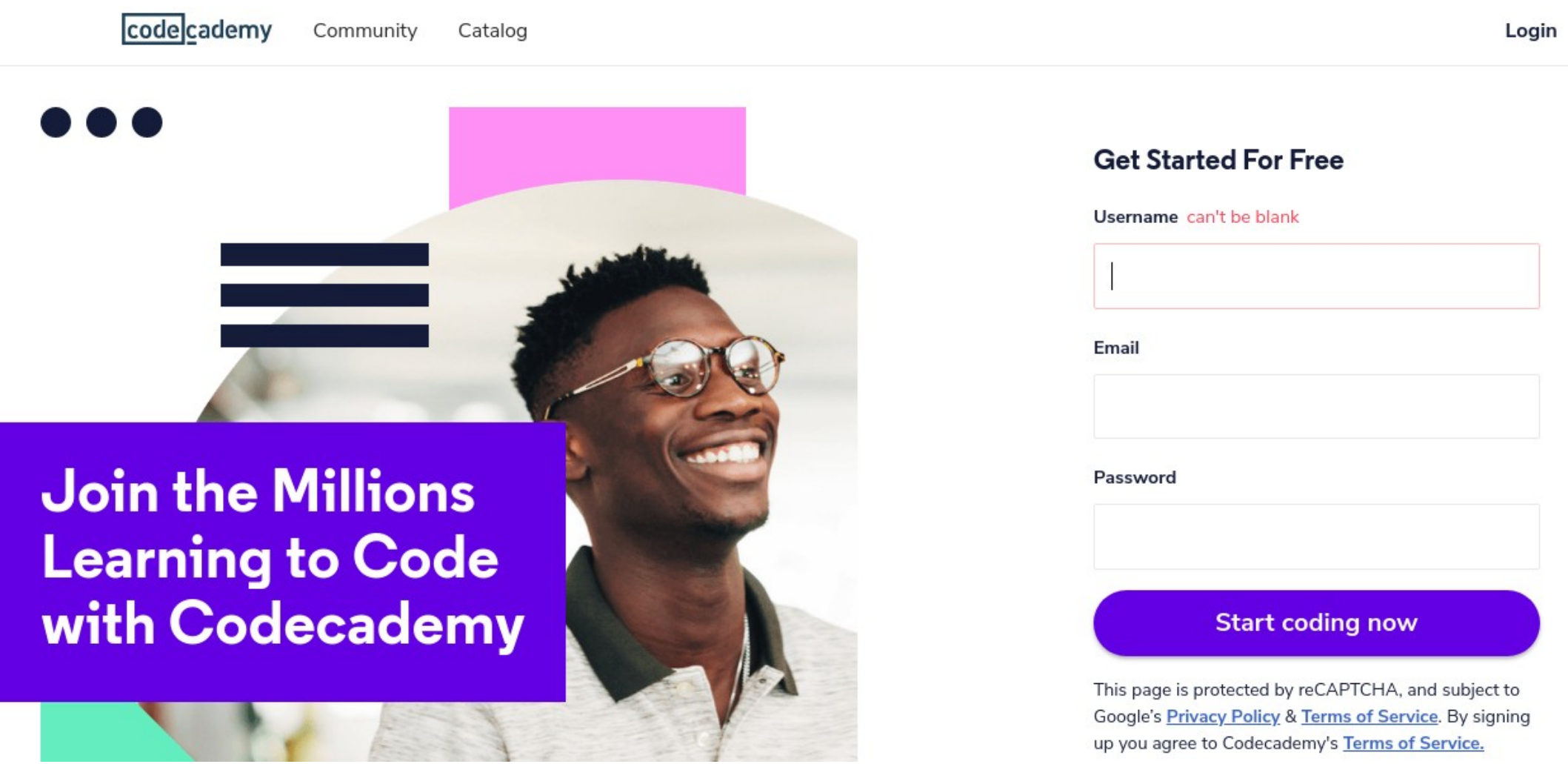
Uma linguagem de **programação** é um conjunto de regras e instruções que, ao ser escrita dentro de uma determinada sintaxe, poderá ser interpretada pelo computador e dessa forma poderemos dar comandos a ele. Desde os comandos mais simples como somar 10 e 20 até os mais complexos, como desenvolvimento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

Onde aprender?



Onde aprender?

Onde aprender?

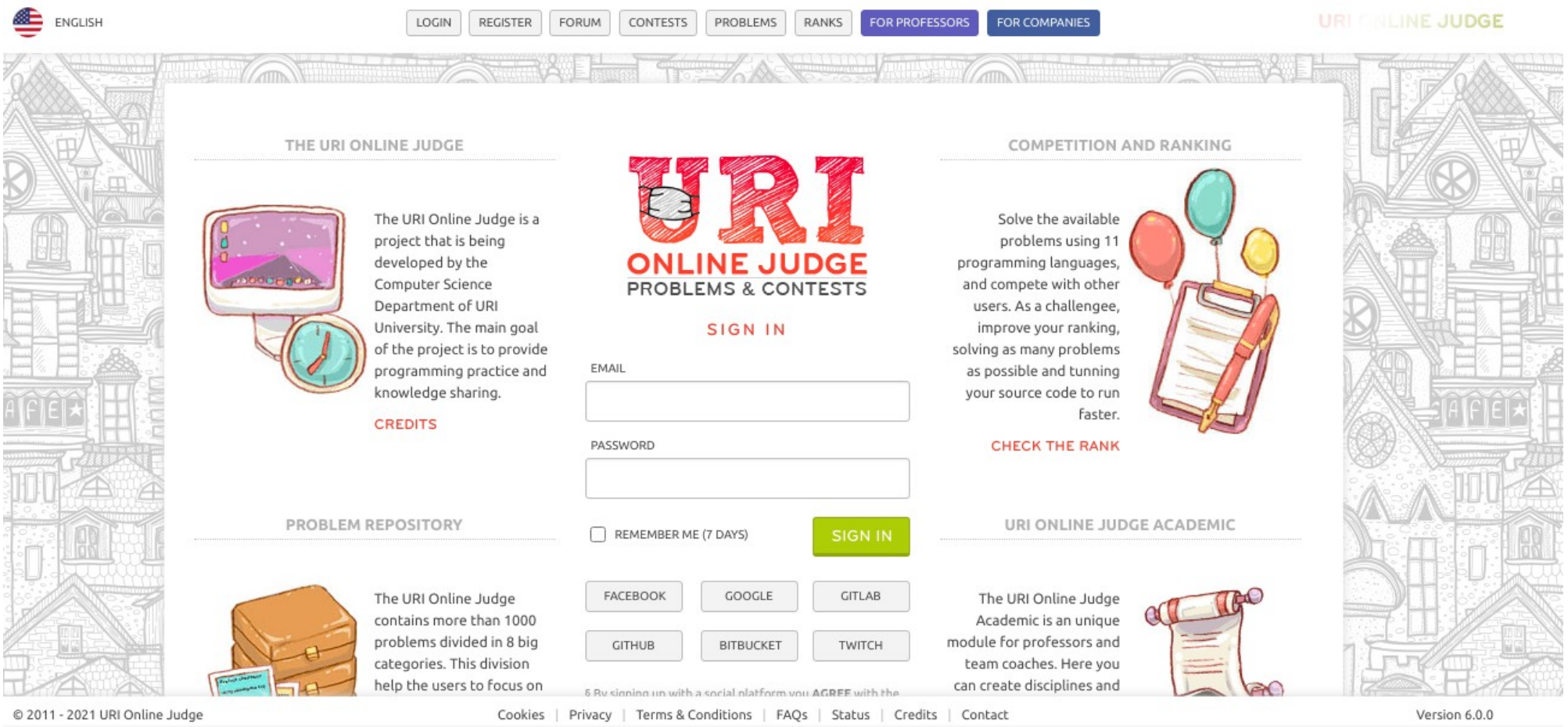


Onde aprender?

AQUI!

Onde treinar?

URI ONLINE JUDGE - Problems & Contests

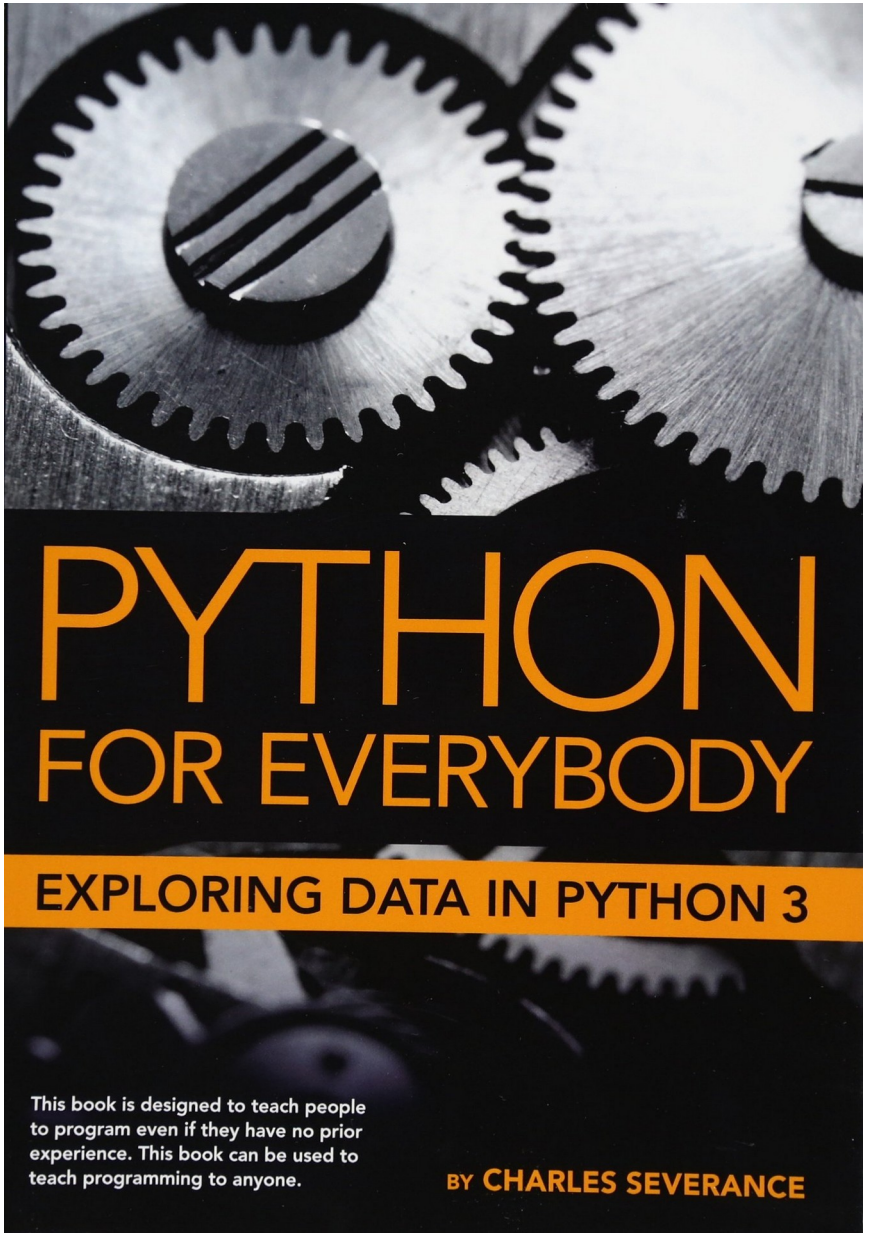


Sobre

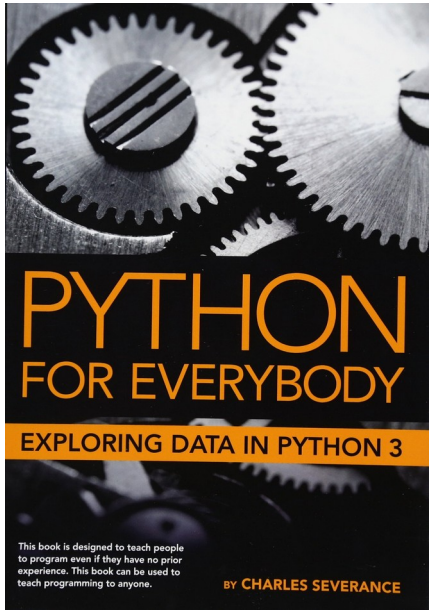
Engenheiro Eletrônico (1983)
Cientista da Computação (1984)
Esp Des. Software para WEB (2005)
Mestre em Desenvolvimento Regional (2011)
Claretinas (2006) e ETEC (2008)
FATEC Franca (2010) e Uni-FACEF (2011)
Fotógrafo, coralista, marido, pai
Avô do Bento

roland@facef.br
carlos.roland@fatec.sp.gov.br
https://www.facebook.com/carlos.e.roland.3

Referência


PYTHON
FOR EVERYBODY
EXPLORING DATA IN PYTHON 3
This book is designed to teach people to program even if they have no prior experience. This book can be used to teach programming to anyone.
BY CHARLES SEVERANCE

Referência



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!’)

>primt(‘Hello world!’)

>primt ‘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

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

O que pode dar errado?

Tipos de erros	Causas
sintaxe	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.
lógica	Scripts sintaticamente corretos, mas em sequência errada, provocando erros de execução. É necessário rever e corrigir o algoritmo.
semântica	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!