

---

# JORNADA PYTHON FAIXA PRETA

---



ByLearn

# Seja Bem-Vindo!



Zarcky

Graduado em Ciência da Computação e Mestrando em  
Computação Aplicada – FFCLRP – USP.

---

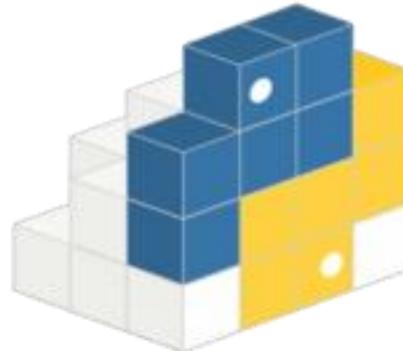
# BÔNUS: O *PIP* E OS PACOTES EXTERNOS

---



# PIP – O Que é

O PIP é o instalador de pacotes padrão do Python. Ele te permite instalar pacotes externos para auxiliar no seu desenvolvimento. Você pode buscar esses pacotes no [Pypi](#).



# Pacotes do Python

Em resumo, todo arquivo (script) do Python é um *módulo*, ou seja, esses códigos que nós desenvolvemos fazem parte de um módulo.

Já o *projeto* como um todo, a junção de todos arquivos, é um *pacote*.

Ou seja... Uma *solução* que fizermos no Python, como uma de *Gerenciar Alunos* pode ser considerado um pacote Python, no caso, posso chamá-lo de *schoolmanager*.

PS: Guardem bem esse nome!



# Pacotes Internos

O Python por conta própria já tem *vários pacotes prontos*, que chamamos de *built-ins*.

Esses pacotes, assim como vimos agora pouco, são *soluções completas*, certo?

Dessa forma, se nós utilizarmos esses pacotes, provavelmente iremos *acelerar o nosso desenvolvimento*.

Afinal, tudo lá já está *pronto para ser usado*!

Basta *importarmos* o pacote/módulo interno.



# Importando

Nós podemos importar pacotes, módulos e funções no Python para facilitar nosso código.

*Por exemplo:* Calcular qual o cubo da raiz quadrada de um número? Importa o *math*!

No caso, usamos *import* para falar que estamos *importando* algo e na frente passamos o *nome*.

No caso, importamos o *math*.

```
● ● ●  
import math  
  
numero = 25  
  
raiz_quadrada = math.sqrt(numero)  
ao_cubo = math.pow(raiz_quadrada, 3)
```

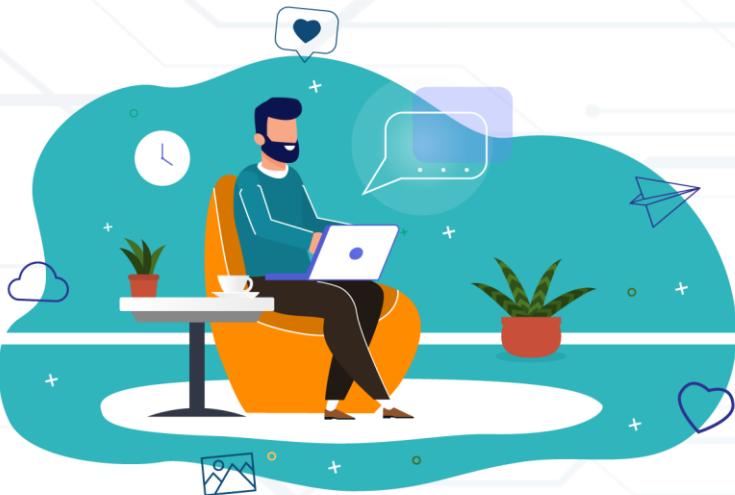
# Pacotes Externos

Assim como nós fazemos as *nossas próprias soluções*, muitos *outros programadores fazem as deles*, certo?

E se eu quiser *compartilhar* o *schoolmanager* com você? E se você quiser *compartilhar* sua solução comigo?

Como isso não vem por padrão no Python, nós chamamos esses pacotes de *pacotes externos*.

Ou seja: *Pacotes externos são soluções feitas por outros programadores*.



# Instalando

Pacotes externos precisam ser *instalados antes* de ser *importados* no nosso código.

Para isso, usamos o *PIP*, lembra dele?

A sintaxe é simples:

*pip install nome\_do\_pacote*

*Ex:*

*pip install schoolmanager*



```
# Esse comando deve ser feito no terminal  
# No windows, você pode usar o CMD.exe
```

```
pip install nome_do_pacote
```

Exemplos:

```
pip install numpy  
pip install pandas  
pip install pillow  
pip install tensorflow  
pip install ...
```

# Pacotes Externos

Sempre que você for iniciar um novo código é interessante buscar na internet se há algum pacote externo que pode te auxiliar.

Ex:

- Para requisições web, temos o *requests*.
- Para gerar pdf temos o *pdfkit*.
- Para criar gráficos temos o *matplotlib*.
- Para vetores e matrizes temos o *numpy*.
- Para imagens temos o *pillow (pil)*.



# REQUESTS



```
# Importando o módulo do Requests e JSON
import requests
import json

# Definindo nosso Payload
payload = {'name': 'Felipe Cabrera', 'job': 'developer'}

# Enviando nosso Payload via POST
response = requests.post('https://reqres.in/api/users', data = json.dumps(payload))

# Vamos verificar o que enviamos
print(response.request.body)
# Output: {"name": "Felipe Cabrera", "job": "developer"}

# Vamos verificar o código de resposta
print(response.status_code)
# Output: 201

# Vamos verificar a resposta
print(response.text)
# Output: {"id": "683", "createdAt": "2020-09-03T12:29:07.721Z"}
```

# PDFKIT

```
import pdfkit

# PDF de um Site
pdfkit.from_url('http://google.com', 'out.pdf')

# PDF de um Arquivo
pdfkit.from_file('test.html', 'out.pdf')

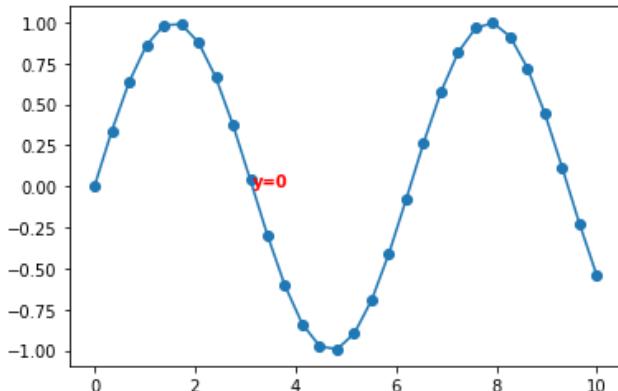
# PDF de um Texto
pdfkit.from_string('#ByLearnerFaixaPreta', 'out.pdf')
```

# MATPLOTLIB



```
import matplotlib.pyplot as plt
import numpy as np

plt.plot(x, np.sin(x), '-o')
plt.text(np.pi, 0, 'y=0', weight='bold', c='r')
plt.show()
```



# NUMPY



```
import numpy as np

# Criar matrizes randômicas
x1 = np.random.randint(10, size=6) # Matriz de uma dimensão
x2 = np.random.randint(10, size=(3, 4)) # Matriz bidimensional
x3 = np.random.randint(10, size=(3, 4, 5)) # Matriz tridimensional

print("Dimensões do x3: ", x3.ndim)
print("Shape (formato) do x3:", x3.shape)
print("Tamanho do x3", x3.size)

print('Primeira coluna do X2:', x2[:, 0])
print('Primeira linha do X2:', x2[0, :])
```

# PILLOW (PIL)

```
from PIL import Image
from PIL import ImageFilter

# Manipula as imagens
imagem_original = Image.open('felipe.png')
imagem_com_filtro =
imagem_original.filter(ImageFilter.BLUR)
imagem_cinza = imagem_original.convert("LA")

# Mostra as imagens
imagem_original.show()
imagem_com_filtro.show()
imagem_cinza.show()

# Salva as imagens
imagem_original.save('original.png')
imagem_com_filtro.save('filtro.png')
imagem_cinza.save('cinza.png')
```



Original



Filtro



Cinza

---

# **EXTRA: RegEx**

## **(Expressões Regulares)**

---

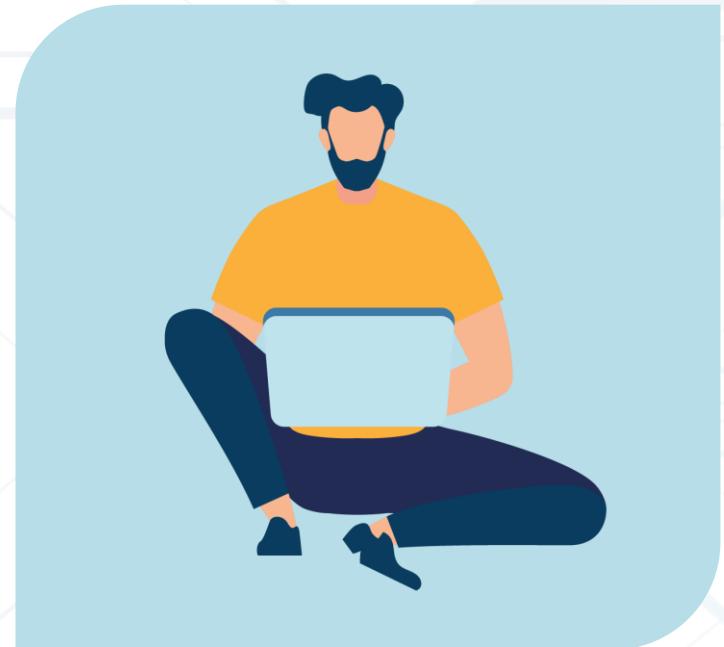


# Expressões Regulares

“Expressões Regulares são padrões utilizados para selecionar combinações de caracteres em uma string.”

De uma forma mais simples, podemos falar que as Expressões Regulares *são strings especiais para buscar padrões em textos.*

Algo como um *CTRL+F bem poderoso!*



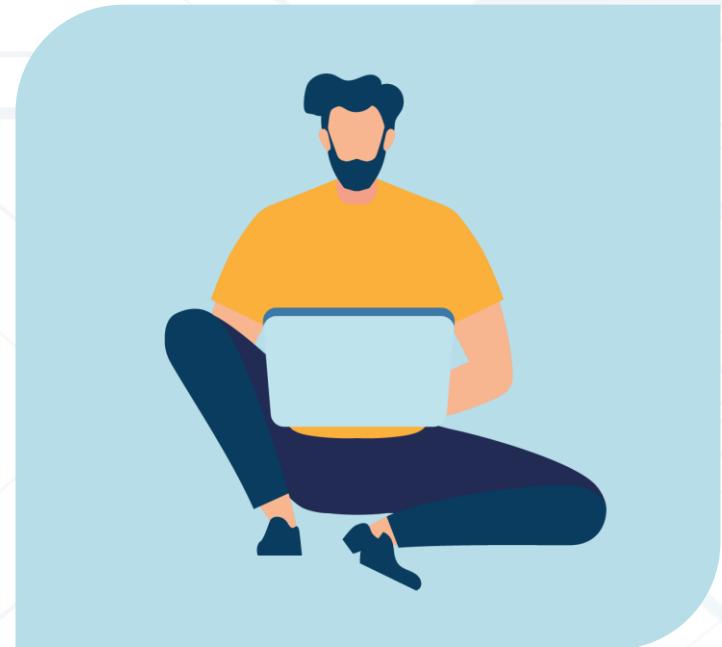
# Expressões Regulares

Seu uso é bem *amplo e importante*.

Na *grande maioria dos softwares* que trabalham com validação de dados e busca de padrões, nós *temos o RegEx sendo usado!*

Exemplos de uso:

- Web Scrapping
  - A coleta/mineração de dados
- Validar E-mails
- Encontrar Palavras Chaves
- Substituir Termos





```
import re

texto = "Venha se tornar um ByLearner!"
padrao = r"\bByLearn\w*\b"

resultado = re.search(padrao, texto)

if resultado:
    print("Encontramos o padrão no texto, entre os índices: "
          +f"{resultado.start()} e {resultado.end()}")
else:
    print("Não encontramos o padrão no texto")
```

*Saída:*

*Encontramos o padrão no texto,  
entre os índices: 19 e 28*



```
import re

texto = "O rato roeu a roupa do rei de Roma"
padrao = r"rato|roupa|rei"

ocorrencias = re.finditer(padrao, texto)

for ocorrencia in ocorrencias:
    print(f"Encontrei: {ocorrencia.group()} entre os índices {ocorrencia.span()}")
```

### Saída:

*Encontrei: rato entre os índices (2, 6)*  
*Encontrei: roupa entre os índices (14, 19)*  
*Encontrei: rei entre os índices (23, 26)*

```
● ● ●  
import re  
  
texto = "Um dois quatro quatro cinco"  
padrao = r"quatro"  
substituicao = r"tres"  
  
novo_texto = re.sub(padrao, substituicao, texto, 1)  
print(novo_texto)
```

*Saída:*

*Um dois três quatro cinco*



```
import re

texto = "O rato roeu a roupa do rei de Roma"
padrao = r"rato|roupa|rei"

ocorrencias = re.finditer(padrao, texto)

for ocorrencia in ocorrencias:
    print(f"Encontrei: {ocorrencia.group()} entre os índices {ocorrencia.span()}")
```

### Saída:

*Encontrei: rato entre os índices (2, 6)*  
*Encontrei: roupa entre os índices (14, 19)*  
*Encontrei: rei entre os índices (23, 26)*



```
import re

texto = "O rato roeu a roupa do rei de Roma"
padrao = r"[rR](\w*)"
substituicao = r"g\1"

novo_texto = re.sub(padrao, substituicao, texto)
print(novo_texto)
```

*Saída:*

*O gato goeu a goupa do gei de goma*

```
● ● ●  
import re  
  
texto = "Felipe Cabrera"  
padrao = r"\w+\s\w+"  
  
regex = re.fullmatch(padrao, texto)  
  
if regex:  
    print("Valido")  
else:  
    print("Inválido")
```

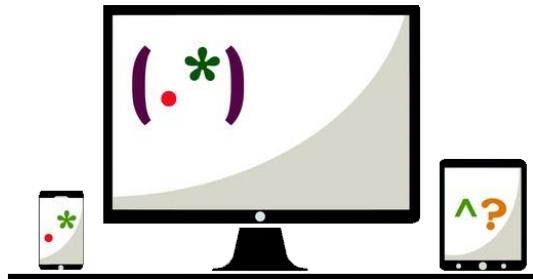
*Saída:*  
*Válido*

# EXPRESSÕES REGULARES

Agora que você já sabe a importância, que tal aprender de verdade as regras do RegEx?

Temos dois materiais fantásticos te esperando:

- 1) Aprenda sobre Expressões Regulares
- 2) Aprenda usar o RegEx com o Python



---

# RESUMO DA AULA 1

---



# CIÊNCIA DA COMPUTAÇÃO

---

## Programação

*Escrever um código que será transformado em um programa.*

## Algoritmo

*A sequência de instruções lógicas que chegam a solução de um problema.*

# CIÊNCIA DA COMPUTAÇÃO

---

## Linguagem de Programação

Conjunto de *regras, padrões e instruções* para comunicarmos com a *máquina* e gerarmos um *software*.

## Código

A *implementação* de um *algoritmo* usando uma *linguagem de programação*.

# CIÊNCIA DA COMPUTAÇÃO

---

## Compilar

*Traduzir* o código *escrito* (fonte) para o código lido pela *máquina* (alvo).

## Interpretar

O código escrito é passado para uma *linguagem intermediária* e executada pelo *interpretador*.

# CIÊNCIA DA COMPUTAÇÃO

---

## Python: O que é

*Linguagem de Programação interpretada*  
criada por Guido van Rossum.

## Python: Motivos

A linguagem que *mais cresce*, tem como objetivo ser *fácil, simples, gratuita e poderosa*. Alta *demand*a de empregos, *versátil* e com bons *salários*.

---

# RESUMO DA AULA 2

---



# PREPARAÇÃO

---

## Instalação

Você pode instalar o *Python* através do *site oficial* ou através do *pacote Anaconda*.

## Google Colab

Uma solução para utilizar o *Python na nuvem!*

O uso é totalmente online!

## IDE

É um *programa* que *auxilia* na hora do *desenvolvimento* (programação)

# PREPARAÇÃO

---

## IDE Escolhida

Nós escolhemos o *Google Colab*, para *ninguém precisar instalar* o Python durante a Jornada.

## Sintaxe

Basicamente, a *sintaxe* está ligada na *forma como escrevemos*. Ela quem dita a *maneira correta* de se escrever

## Sintaxe na Programação

Cada linguagem de programação possui a *sua sintaxe*, ou seja, a *sua forma de escrever*.

# PYTHON

---

## Pythonês

O Python é lido *da esquerda pra direita e de cima pra baixo*, com sintaxe *igual a de um texto normal*

## Indentação

No Python o *nível mais a direita pertence* ao *nível à sua esquerda*.

## Variáveis

Uma variável é um *identificador* que se *refere a um valor*

# PYTHON

---

## Tipos de Dados

Cada *dado* irá possuir um *tipo*.

É isso que definirá se ele é um *número*, *texto*, *booleano*...

## Métodos de Entrada

São formas de *enviar dados do usuário ao Python.*

## Métodos de Saída

São formas do Python de *mostrar dados ao usuário.*

# PYTHON

---

## Dados Sequenciais

Para trabalharmos com valores sequenciais *podemos usar listas*

## Elementos em Listas

Quando queremos usar um elemento da lista, *devemos utilizar o seu índice.*

## Índices x Posições

Posições iniciam em 1. Já os índices, iniciam em 0.

Basicamente:  
*Índice = Posição – 1*

# PYTHON

---

## Condições

Formas de realizar  
*tomadas lógicas*  
*de decisão.*

*Se... Então*  
*Senão, se... Então*  
*Senão... Então*

# GITHUB

## O que é o GitHub

É a principal plataforma para *hospedagem e compartilhamento* de *códigos* (de programação) do mundo!

Ótimo para *encontrar* projetos e  
*estudar*



---

# RESUMO DA AULA 3

---



# PYTHON

---

## Operadores Lógicos

Podemos usar operadores lógicos para criar *condicionais compostas*.

## Laços de Repetição

Excelente forma de *evitar repetição* de código.

## Repetições Dinâmicas

Podemos pegar o *tamanho* das sequências ou usar a *própria variável* nos *laços de repetição*.

# PYTHON

---

## Funções

Outra forma de evitar repetição de código, onde definimos uma *rotina para ser executada*.

## PIP

*Instalador de Pacotes* oficial do Python.

## Módulos e Pacotes

Todo arquivo é um *módulo*, a junção de todos módulos é o *pacote*.

# PYTHON

---

## Pacotes Externos

São soluções feitas  
por *outros  
programadores.*

## RegEx

São strings  
especiais para  
*buscar padrões*  
em textos

---

# VAMOS PRA AULA 4!

---



---

# O EXTRA DO BÔNUS DO BÔNUS

---





# Banco de Dados

*O que são e como usar?*



# Banco de Dados

Quando você precisa guardar seu dinheiro em um *lugar seguro e que consiga guardar* esse valor em dinheiro *por tempo indeterminado*, você vai até um *Banco*, certo?

Esse banco acima seria um “*Banco de Dinheiro*”.

Com dados, é a mesma coisa.

O *banco de dados* nada mais é que um lugar para *armazenar dados*.

Com dados queremos dizer:

- *Nome, Sobrenome, Idade, Login, Senha...*



# Banco de Dados

Agora falando de uma forma mais técnica, os banco de dados são *coleções de dados interligados entre si*.

Os *principais motivos* de usar banco de dados são:

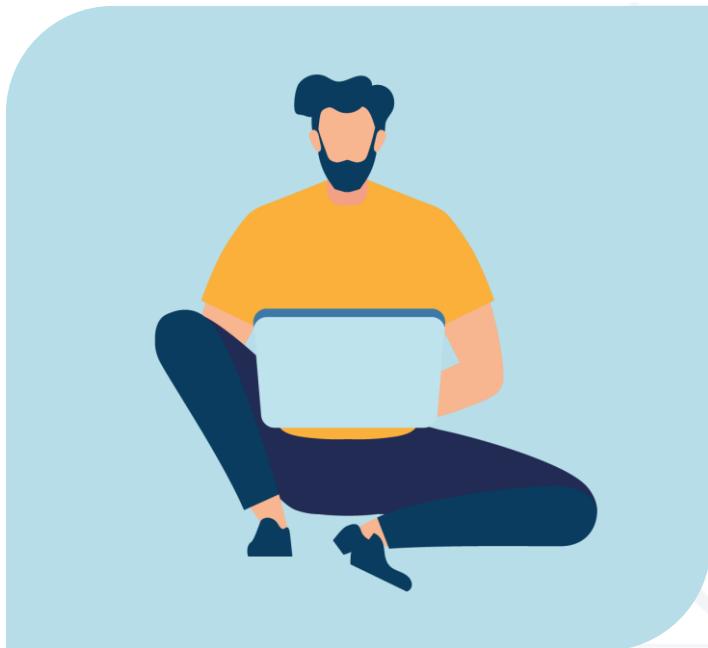
- *Salvar informações* mesmo após o término da aplicação.
- Guardar *dados de vários usuários* no seu próprio *servidor*.

Quer ler um artigo legal sobre o tema?  
Então [clique aqui!](#)!



# OPERAÇÕES COM BANCO DE DADOS

---



Você pode usar banco de dados para *diversas funções*, dentre elas:

- *Adicionar* um novo valor (linha)
- *Editar* um valor existente
- *Deletar* um valor
- *Criar* uma nova tabela
- *Editar* uma tabela
- *Deletar* uma tabela
- *Recuperar* valores
- Entre outras

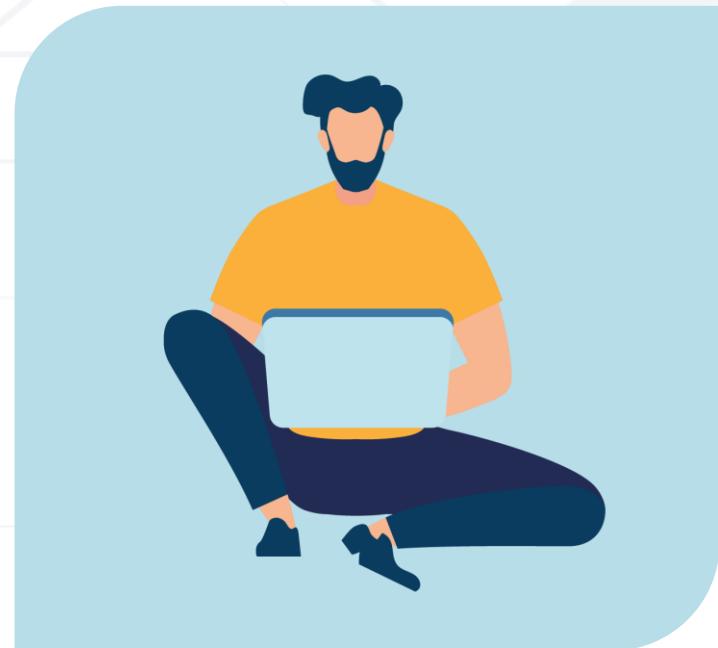
# TIPOS DE BANCOS DE DADOS

Basicamente, temos dois tipos de bancos:

- *SQL* (Relacionais)
- *NoSQL* (Não Relacionais)

Os *primeiros são criados com orientação a conjuntos* e por isso suas informações são organizadas em *tabelas*, integrando as *colunas* e *linhas*.

Já os segundos, são para *dados mais complexos e que não podem ser tabulados*, como imagens, mapas e dados mistos.



# Banco de Dados SQL

Para isso, nós temos as *DataBases*, que são nossas bases de dados, como:

*gerenciador\_aluno*

Temos as tabelas, que são nossas estruturas, como:

*alunos e notas*

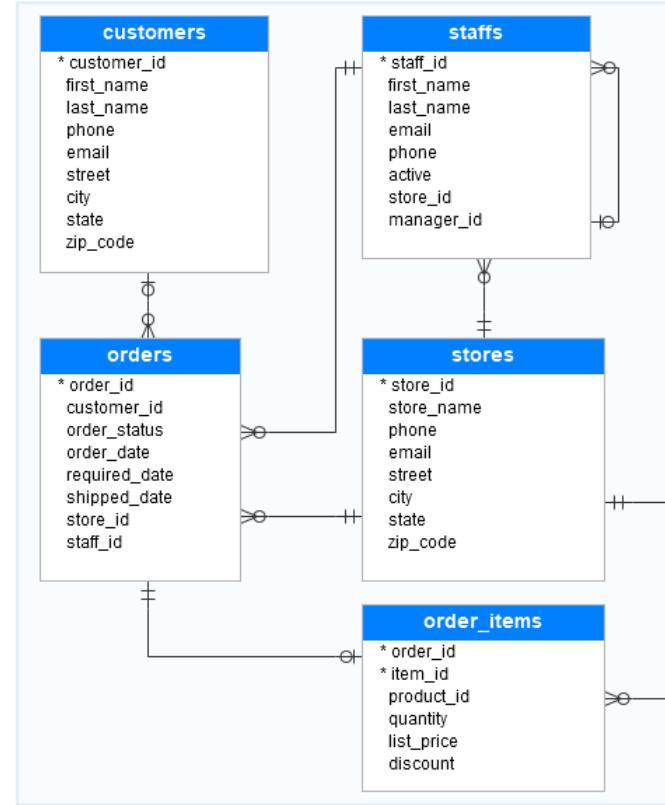
Temos as colunas, que são as propriedades das tabelas, como:

*nome, sobrenome e idade*

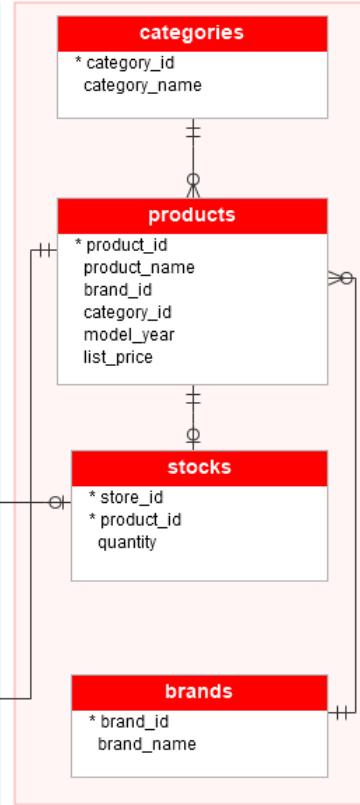
Por fim, temos *as linhas*, que são os *dados que inserimos nas tabelas*, com seus respectivos *valores para cada coluna*.



### Sales



### Production



## Data View - Employee ( C:\Program Files\Autodesk\AutoCAD 2016\Sample\Database Connec... )

— New Link Template — | — New Label T

|   | Emp_Id | Last_Name | First_Name | Gender | Title              |
|---|--------|-----------|------------|--------|--------------------|
| ▶ | 1000   | Torbati   | Yolanda    | F      | Programmer         |
|   | 1001   | Kleinn    | Joel       | M      | Programmer         |
|   | 1002   | Ginsburg  | Laura      | F      | President          |
|   | 1003   | Cox       | Jennifer   | F      | Programmer         |
|   | 1005   | Ziada     | Mauri      | M      | Product Designer   |
|   | 1006   | Keyser    | Cara       | F      | Account Executive  |
|   | 1010   | Smith     | Roxie      | M      | Programmer         |
|   | 1011   | Nelson    | Robert     | M      | Programmer         |
|   | 1012   | Sachsen   | Lars       | M      | Support Technician |
|   | 1013   | Shannon   | Don        | M      | Product Designer   |

Record 1

| First Name | Last Name | Address             | City     | Age |
|------------|-----------|---------------------|----------|-----|
| Mickey     | Mouse     | 123 Fantasy Way     | Anaheim  | 73  |
| Bat        | Man       | 321 Cavern Ave      | Gotham   | 54  |
| Wonder     | Woman     | 987 Truth Way       | Paradise | 39  |
| Donald     | Duck      | 555 Quack Street    | Mallard  | 65  |
| Bugs       | Bunny     | 567 Carrot Street   | Rascal   | 58  |
| Wiley      | Coyote    | 999 Acme Way        | Canyon   | 61  |
| Cat        | Woman     | 234 Purrfect Street | Hairball | 32  |
| Tweety     | Bird      | 543                 | Itotltaw | 28  |

# SQLite

---

A colorful illustration depicting a group of people working on computers. One person on the left sits at a desk with two monitors, looking at code snippets like '</>' and 'SQL'. Another person in the center stands behind a desk with multiple monitors, also looking at code. In the foreground, two more people are seated on bean bags, one using a laptop and the other a smartphone, both looking at code snippets. The background features abstract blue and green shapes.

Um exemplo bem famoso de Banco de Dados SQL é o *SQLite*.

Ele é bem *simples, leve e fácil*.

Saiba mais clicando [aqui](#).

# CRIANDO O BANCO

---

```
import sqlite3  
  
arquivo_db = "bylearn.db"  
conexao = sqlite3.connect(arquivo_db)
```

# CRIANDO A TABELA

```
● ● ●  
import sqlite3  
  
sql_criar_tabela = """ CREATE TABLE IF NOT EXISTS alunos (  
                    id integer PRIMARY KEY AUTOINCREMENT,  
                    nome text NOT NULL,  
                    nota integer NOT NULL  
                ); """  
  
conexao = sqlite3.connect('bylearn.db')  
cursor = conexao.cursor()  
cursor.execute(sql_criar_tabela)
```

# INSERINDO DADOS

```
import sqlite3

sql_inserir_aluno_felipe = "INSERT INTO alunos (nome, nota) VALUES ('Felipe',10)"
sql_inserir_aluno_jose = "INSERT INTO alunos (nome, nota) VALUES ('José',8)"

conexao = sqlite3.connect('bylearn.db')
cursor = conexao.cursor()

cursor.execute(sql_inserir_aluno_felipe)
cursor.execute(sql_inserir_aluno_jose)

conexao.commit()
```

# BUSCANDO DADOS

```
● ● ●  
import sqlite3  
  
sql_buscar_alunos = "SELECT * FROM alunos"  
  
conexao = sqlite3.connect('bylearn.db')  
cursor = conexao.cursor()  
  
cursor.execute(sql_buscar_alunos)  
alunos = cursor.fetchall()  
  
for aluno in alunos:  
    print("O aluno", aluno[1], "tirou nota", aluno[2])
```

---

# PRINCIPAIS DÚVIDAS

---



# PRINCIPAIS DÚVIDAS

Algumas dúvidas são únicas, outras muitos ByLearners possuem.  
Reunimos as mais comuns aqui, para que você tenha [todas as respostas!](#)

## Por quanto tempo eu terei acesso?

Você terá acesso vitalício ao curso!

É isso mesmo:

*Será Python hoje, Python amanhã e  
Python SEMPRE!*

## Eu realmente aprenderei Python?

Sim! *Temos mais de 30.000 alunos que comprovam nossa qualidade* e dão feedback para nossa melhoria contínua de didática e conteúdo.

Ensinamos com didática exclusiva e validada, desde o Zero Absoluto até o Avançado do Python.

# PRINCIPAIS DÚVIDAS

Algumas dúvidas são únicas, outras muitos ByLearners possuem.  
Reunimos as mais comuns aqui, para que você tenha [todas as respostas!](#)

## Eu terei alguma garantia?

Sim! Isso é algo extremamente necessário!

Aqui na ByLearn nós damos uma *garantia incondicional de 7 dias*.

Se a qualquer momento você quiser seu investimento de volta, é só avisar!

## Como vou acessar o curso?

Assim que a matrícula for aprovada você, *automaticamente*, receberá um *e-mail com os dados* para acessar o curso.

# PRINCIPAIS DÚVIDAS

Algumas dúvidas são únicas, outras muitos ByLearners possuem.  
Reunimos as mais comuns aqui, para que você tenha [todas as respostas!](#)

Vou ter algum suporte?

Claro! É o que eu sempre digo, pode contar comigo sempre!

Uma vez ByLearner...  
sempre ByLearner!

Você terá *suporte completo e vitalício!*

Terei um certificado?

Opa, terá sim!

Ao final do curso você poderá pegar seu certificado, para *comprovar que você aprendeu* todo o conteúdo que ensinamos.

# PRINCIPAIS DÚVIDAS

Algumas dúvidas são únicas, outras muitos ByLearners possuem.  
Reunimos as mais comuns aqui, para que você tenha [todas as respostas!](#)

## O curso é atualizado?

Sim! A tecnologia sempre evolui e precisamos nos atualizar sempre, para que você nunca fique para trás!

Essas atualizações são tanto para as aulas existentes, *melhorando sua qualidade*, quanto para *novos conteúdos*.

## Quantas horas de duração?

Atualmente temos cerca de 30 horas de duração, lembrando que temos atualizações constantes!

Nossa didática é bem direta ao ponto e inovadora, você aprenderá todo o necessário para se *Destacar no Mercado de Trabalho* e muito mais!

# PRINCIPAIS DÚVIDAS

Algumas dúvidas são únicas, outras muitos ByLearners possuem.  
Reunimos as mais comuns aqui, para que você tenha [todas as respostas!](#)

## O curso é mesmo para mim?

Se você é fã de tecnologia, quer aprender a programar, ou já conhece uma linguagem de programação e deseja migrar para o Python, [o curso é para você!](#)

Se assim como nossos 30.000 alunos você deseja entrar de vez no mercado de trabalho com salários altíssimos, [então é sim para você!](#)

## O curso é totalmente online?

Sim! Nossa curso é [% online e multiplataformas.](#)

Você pode assistir pelo seu celular, tablet, computador ou até televisão.

# PRINCIPAIS DÚVIDAS

Algumas dúvidas são únicas, outras muitos ByLearners possuem.  
Reunimos as mais comuns aqui, para que você tenha [todas as respostas!](#)

Terei exemplos práticos?

Sem sombras de dúvidas!

*Nosso curso é totalmente prático,*  
afinal, programação só se aprende  
praticando!

O que vocês ensinam?

Ensinaremos desde o básico até o  
avançado do Python!

Terminando o *curso você poderá*  
*facilmente entrar no mercado* de  
trabalho e terá o necessário para  
iniciar sua jornada em qualquer área  
que utilize o Python.

# PRINCIPAIS DÚVIDAS

Algumas dúvidas são únicas, outras muitos ByLearners possuem.  
Reunimos as mais comuns aqui, para que você tenha [todas as respostas!](#)

É em videoaulas?

Pode ter certeza!

E tudo gravado em Full-HD, com a melhor qualidade possível, tanto de vídeo quanto áudio.

*Fazendo uso de todas as boas práticas de gravação e usabilidade de usuário.*

Eu terei apostilas?

Boa pergunta! Esse é um *diferencial da ByLearn!*

Todos os módulos acompanham uma *apostila exclusiva* para você sempre continuar aprendendo.

# PRINCIPAIS DÚVIDAS

Algumas dúvidas são únicas, outras muitos ByLearners possuem.  
Reunimos as mais comuns aqui, para que você tenha [todas as respostas!](#)

## O certificado é valido onde?

Nosso certificado é totalmente válido e oficial, portanto, ele é reconhecido em *todo território nacional*.

Aceito tanto nas maiores empresas quanto em qualquer faculdade!

## Porque escolher o Python?

Além de ser *muito fácil e poderosa*, o Python é a *linguagem que mais cresce no momento!*

As vagas estão bombando, com uma *demandas enorme* e pagando *salários altíssimos!*

# PRINCIPAIS DÚVIDAS

Algumas dúvidas são únicas, outras muitos ByLearners possuem.  
Reunimos as mais comuns aqui, para que você tenha [todas as respostas!](#)

Poderei sair do curso trabalhando?

Sim! Certamente!

*Nós temos inúmeros alunos* que nos agradecem pelo conteúdo e dizem *que conseguiram ótimos empregos* com o conteúdo que aprenderam no nosso curso!

Pra que o Python é usado?

*Praticamente para tudo* no mundo digital!

Inteligência Artificial, Ciência de Dados, Big Data, Desenvolvimento Web, Desenvolvimento de Automações, Cyber-Segurança, Servidores em Nuvem *e muito mais...*

---

# POR QUE A BYLEARN?

---



- 
- Centrais do Aluno
  - Milhares de avaliações  
5 estrelas
  - Mais de 30 mil  
alunos formados
  - Diploma de Conclusão
  - Acesso Vitalício

# POR QUE ESCOLHER A BYLEARN?

Atualizações Constantes



Supporte completo



Multiplataformas



Material didático  
complementar



Garantia de Qualidade



# POR QUE ESCOLHER A BYLEARN?

# POR QUE A BYLEARN?

Falar de nós é fácil, e até parece falsidade... Então deixo aqui alguns depoimentos:



**William Tomé**

ALUNO DOS CURSOS DA BYLEARN | DEPOIMENTO VIA FACEBOOK

" Recomendo a todos que amam tecnologia, principalmente programação os cursos da ByLearn, pois a didática deles é ótima, explicações são claras e com muita mão na massa.Valeu ByLearn! Continuem com essa boa qualidade! "



**Rafael Gimenez**

ALUNO DOS CURSOS DA BYLEARN | DEPOIMENTO VIA FACEBOOK

" Cursos excelentes recomendo muito! Com o curso VS Code Produtividade Infinita pude aprender a utilizá-lo de forma eficaz no meu dia a dia. A didática é muito boa, com suporte exclusivo e os instrutores são muito atenciosos. Recomendo demais a ByLearn! "



**Ag Pessoa**

ALUNO DOS CURSOS DA BYLEARN | DEPOIMENTO VIA FACEBOOK

" Estou fazendo o curso VS Code: Produtividade Infinita. Hoje por causa dos conhecimentos adquiridos tenho um domínio fantástico da ferramenta. Parabéns ByLearn seus cursos são fantásticos. Super indico a quem quer aprender a desenvolver de verdade. Show 5 estrelas! "

# POR QUE A BYLEARN?

Falar de nós é fácil, e até parece falsidade... Então deixo aqui alguns depoimentos:



**Vinicius Virgulino**

ALUNO DOS CURSOS DA BYLEARN | DEPOIMENTO VIA FACEBOOK

“ Fiz o curso de VS Code oferecido pela ByLearn pois queria melhorar minhas habilidades com o VSCode, achei o curso muito bom, recomendo para todos aqueles que fazem uso do VS code, não irão se decepcionar. Parabéns pelo excelente curso. ”



**Celso Cortes**

ALUNO DOS CURSOS DA BYLEARN | DEPOIMENTO VIA FACEBOOK

“ Fiz o curso Android 9.0 e adorei a didática do curso, simples e com clareza, recomendo muito para quem quer iniciar a programar. ”



**Martins Gouveia**

ALUNO DOS CURSOS DA BYLEARN | DEPOIMENTO VIA FACEBOOK

“ Excelente plataforma de estudos, com uma didática leve e muito simples de se aprender. Amei ”

# POR QUE A BYLEARN?

Falar de nós é fácil, e até parece falsidade... Então deixo aqui alguns depoimentos:



**Charles**

ALUNO DOS CURSOS DA BYLEARN | DEPOIMENTO VIA FACEBOOK

" Excelente plataforma para estudos, professores usam uma didática muito boa, sempre que possível entro em sua plataforma para ver as novidades, quem está no ramo de TI sabe a importância de se manter atualizado e ByLearn faz isso por nós. "



**Luis Augusto**

ALUNO DOS CURSOS DA BYLEARN | DEPOIMENTO VIA FACEBOOK

" Fiz o curso de VS Code: Produtividade Infinita, os professores tem uma ótima didática, os cursos são bem completos com explicações claras e coesas, o que aumenta o interesse em aprender o que está sendo ensinado. Parabéns ByLearn, estão no caminho certo!!! "



**Regiane Badia**

ALUNA DOS CURSOS DA BYLEARN | DEPOIMENTO VIA FACEBOOK

" Muito bom os cursos recomendaria muito.."

# POR QUE A BYLEARN?

Falar de nós é fácil, e até parece falsidade... Então deixo aqui alguns depoimentos:



**Maycon Cardoso**

ALUNO DOS CURSOS DA BYLEARN | DEPOIMENTO VIA FACEBOOK

“ Recomendo muito, Professor bem didático.”



**Kálil Maciel**

ALUNO DOS CURSOS DA BYLEARN | DEPOIMENTO VIA FACEBOOK

“ A didática do professor é muito interessante. Ele sabe seguir um ritmo bom tanto pra novatos quanto para experientes.”



**Emílio C. Dias**

ALUNO DOS CURSOS DA BYLEARN | DEPOIMENTO VIA FACEBOOK

“ Muito didático e com conteúdo realmente agregador, recomendo!”

# POR QUE A BYLEARN?

Falar de nós é fácil, e até parece falsidade... Então deixo aqui alguns depoimentos:



**Anthony Oliveira**

ALUNO DOS CURSOS DA BYLEARN | DEPOIMENTO VIA FACEBOOK

" Uns dos melhores cursos que já fiz na internet, Como sou novo em desenvolvimento foi crucial para conhecer o software mais atuado "



**Thiago Pereira**

ALUNO DOS CURSOS DA BYLEARN | DEPOIMENTO VIA FACEBOOK

" Cursos muito bem feitos , excelente didatica "



**Kriva Junior**

ALUNO DOS CURSOS DA BYLEARN | DEPOIMENTO VIA FACEBOOK

" Melhor custo beneficio e ótimas aulas "

# POR QUE A BYLEARN?

*PREFERE EM VÍDEO? ENTÃO TUDO BEM...*



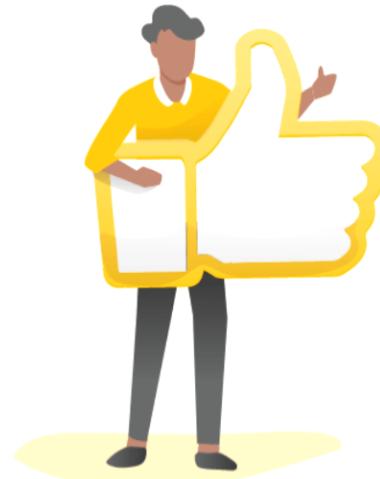
Caso o vídeo acima não tenha carregado:

<https://youtu.be/bFKMvMu5Sel>

# **POR QUE A BYLEARN?**

Para verificar *mais centenas* de depoimentos, clique na imagem abaixo:

**Todos os  
Depoimentos**



Caso você não seja redirecionado com a imagem acima:

<https://bylearn.com.br/depoimentos>

---

# SE LIGA NESSA SUPER OFERTA

---



---

# PACOTE PYTHON

# FAIXA PRETA

---



ByLearn



Curso Python Completo  
do Júnior ao Sênior



Curso Integrando o  
Python com o Word



Curso de Boas  
Práticas com Python



Curso Integrando o  
Python com o Excel

**PACOTE  
PYTHON**

**FAIXA**

**PRETA**



Curso Integrando o Python com PDF



Curso Integrando o Python com o Sistema Operacional



Curso Banco de Dados com o Python



Curso Expressões Regulares com Python

# PACOTE PYTHON FAIXA PRETA



Curso Git com Github



Gravação da Jornada  
Python Faixa Preta



Live sobre o curso  
Python Completo



Live sobre Banco de  
Dados com Python

# PACOTE PYTHON FAIXA PRETA



Live sobre RegEx  
com o Python



Todos os módulos  
possuem Apostilas



Live sobre Git com GitHub



O Curso Completo  
também possui Apostila!

# PACOTE PYTHON FAIXA PRETA

# ACHOU POUCO?

---

ENTÃO SE LIGA SÓ NO BÔNUS  
EXCLUSIVO QUE OS *100 PRIMEIROS*  
*ALUNOS QUE FAZEM A MATRÍCULA*  
IRÃO RECEBER *INTEIRAMENTE*  
*GRÁTIS!*





## Mentoria VIP sobre o Mercado de Trabalho da TI.

Ensinarei meus segredos para se dar  
bem numa entrevista de emprego!

*Bora conseguir aquela sonhada vaga?*



Mentoria VIP sobre *Personal Branding*!

Te ensinarei como eu faço para as vagas  
virem até mim!

*Você encontrará a vaga desejada sem  
nem precisar procurar!*

# GOSTOU DAS LIVES?

---

SE VOCÊ GOSTOU DAS LIVES DA JORNADA, TENHO UMA SUPRESA:

OS 100 PRIMEIROS TAMBÉM PODERÃO ESCOLHER O TEMA DE UMA LIVE COMIGO!





**ENTÃO CORRA!**

---

POIS SERÁ POR TEMPO LIMITADO E A  
*AMPULHETA JÁ ESTÁ CONTANDO!*

VOCÊ  
NÃO  
VAI QUERER  
FICAR FORA  
DESSA TURMA!



# BÔNUS: PESQUISANDO DÚVIDAS CORRETAMENTE

*Tudo tem uma resposta no Google,  
quer saber como?*



# RESUMA TUDO

---



Encontre a “Raiz do Problema”, ou seja, apenas as palavras chaves do seu problema.



Não use “Stopwords” nas suas buscas, o Google não vai dar bola para elas mesmo...

Infelizmente o Google não quer conversar com você, então nada de bate-papo!

Pesquise apenas as palavras-chaves do seu problema, como por exemplo:

Errado: “Eu gostaria de acrescentar um dia na minha variável de datas no Python”

Certo: “Python **acrescentar dia data**”

Veja um resultado legal dessa busca clicando [aqui](#).

# EVITE STOPWORDS!

```
In [10]: lista_Stop = nltk.corpus.stopwords.words('portuguese')
np.transpose(lista_Stop)
```

```
Out[10]: array(['de', 'a', 'o', 'que', 'e', 'do', 'da', 'em', 'um', 'para', 'com',
   'não', 'uma', 'os', 'no', 'se', 'na', 'por', 'mais', 'as', 'dos',
   'como', 'mas', 'ao', 'ele', 'das', 'à', 'seu', 'sua', 'ou',
   'quando', 'muito', 'nos', 'já', 'eu', 'também', 'só', 'pelo',
   'pela', 'até', 'isso', 'ela', 'entre', 'depois', 'sem', 'mesmo',
   'aos', 'seus', 'quem', 'nas', 'me', 'esse', 'eles', 'você', 'essa',
   'num', 'nem', 'suas', 'meu', 'ás', 'minha', 'numa', 'pelos',
   'elas', 'qual', 'nós', 'lhe', 'deles', 'essas', 'esses', 'pelas',
   'este', 'dele', 'tu', 'te', 'vocês', 'vos', 'lhes', 'meus',
   'minhas', 'teu', 'tua', 'teus', 'tuas', 'nosso', 'nossa', 'nossos',
   'nossas', 'dela', 'delas', 'esta', 'estes', 'estas', 'aquele',
   'aquela', 'aqueles', 'aquelas', 'isto', 'aquilo', 'estou', 'está',
   'estamos', 'estão', 'estive', 'esteve', 'estivemos', 'estiveram',
   'estava', 'estávamos', 'estavam', 'estivera', 'estivéramos',
   'esteja', 'estejamos', 'estejam', 'estivesse', 'estivéssemos',
   'estivessem', 'estiver', 'estivermos', 'estiverem', 'hei', 'há',
   'havemos', 'hão', 'houve', 'houvemos', 'houveram', 'houvera',
   'houvéramos', 'haja', 'hajamos', 'hajam', 'houvesse',
   'houvéssemos', 'houvessem', 'houver', 'houvermos', 'houverem',
   'houverei', 'houverá', 'houveremos', 'houverão', 'houveria',
   'houveríamos', 'houveriam', 'sou', 'somos', 'são', 'era', 'éramos',
   'eram', 'fui', 'foi', 'fomos', 'foram', 'fora', 'fôramos', 'seja',
   'sejamos', 'sejam', 'fosse', 'fôssemos', 'fossem', 'for', 'formos',
   'forem', 'serei', 'será', 'seremos', 'serão', 'seria', 'seríamos',
   'seriam', 'tenho', 'tem', 'temos', 'tém', 'tinha', 'tínhamos',
   'tinham', 'tive', 'teve', 'tivemos', 'tiveram', 'tivera',
   'tiverámos', 'tenha', 'tenhamos', 'tenham', 'tivesse',
   'tivéssemos', 'tivessem', 'tiver', 'tivermos', 'tiverem', 'terei',
   'terá', 'teremos', 'terão', 'teria', 'teríamos', 'teriam'],
  dtype='<U12')
```



A ByLearn certifica que

FELIPE CABRERA

---

concluiu com êxito a Jornada Python Faixa Preta, no período de  
05/10/2020 a 08/10/2020, com carga horária de 10 horas.

FCR Santos

**FELIPE CABRERA**  
(PROFESSOR)



# USE O INGLÊS

É fato que você encontrará muitas coisas em português, mas certamente você *encontrará muito mais em inglês!*

Mesmo que você não saiba inglês, é possível usar o *Google Translate* para isso.

Com o tempo, *você aprenderá o padrão* usado no inglês de desenvolvedores, isso é facinho!

Por exemplo, para o nosso slide anterior, eu buscara por “*Python Add Days Date*”

Você pode ver um resultado legal clicando [aqui](#).





Não tenha medo de abrir abas, o  
Chrome vai usar sua RAM de  
qualquer jeito!

Use *mais de um site* como referência.



Alguns sites são sites-chaves para encontrar soluções, como o *StackOverflow*.

# Inurl

Você também pode forçar apenas resultados de um determinado site, usando o inurl.

*Por exemplo:* Python sort list inurl:stackoverflow

No caso, só serão aceitos resultados onde na URL existe stackoverflow

The screenshot shows a search results page for the query "Python sort list inurl:stackoverflow". The results are filtered to show only links from stackoverflow.com. The first result is a question titled "Sort a list in python - Stack Overflow" with the following snippet:  
5 de out. de 2013 - Use abs as key to the sorted function or `list.sort`: >>> lis = [-1,-5,10,6,3,-4,-9]  
>>> sorted(lis, key=abs, reverse=True) [10, -9, 6, -5, -4, 3, 1].  
3 respostas

Below this, there are four more questions listed with their titles, number of answers, and last update date:

- How to sort a list of strings? 11 respostas 30 de ago. de 2008
- Python Sort List 5 respostas 11 de jan. de 2011
- sorting list in python 5 respostas 30 de mar. de 2011
- sorting lists in python 3 3 respostas 1 de jun. de 2013

A link "Mais resultados de stackoverflow.com" is also present.

The second result is a question titled "Confusion with list.sort() in python - Stack Overflow" with the following snippet:  
6 de mar. de 2019 - The list `sort` is pretty self explanatory as it is just sorting the numbers numerically. In list2 the values are stored as strings. So it is comparing ...  
4 respostas

The third result is a question titled "How to sort a list of strings numerically? - Stack Overflow" with the following snippet:  
I know that this sounds trivial but I did not realize that the `sort()` function of **Python** was weird. I have a `list` of "numbers" that are actually in string form, so I first ...  
13 respostas

The fourth result is a question titled "Sort a list by multiple attributes? - Stack Overflow" with the following snippet:  
22 de abr. de 2016 - ... the same using itemgetter (which is faster and avoids a **Python** function call): ... I had a `list` of tuples that needed sorting 1st by descending integer values ... This will use the generated lambda function `sort` the `list` by object.  
6 respostas

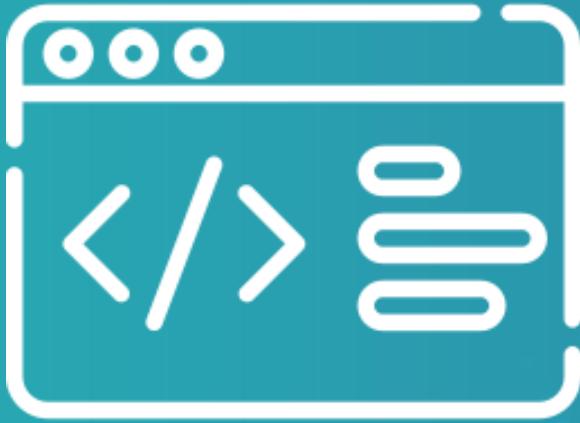
# STACKOVERFLOW

---

Sem sombras de dúvidas a StackOverflow é a *maior comunidade de desenvolvedores* do mundo!

É lá *onde* você *encontrará* a grande maioria das suas *respostas*.





# CONHECENDO AS IDES

*Qual será sua favorita?*



# VS Code

---

Na verdade, o VS Code é considerado um *Editor de Códigos*.

Embora seja muito *poderoso* e *completo*, com inúmeras extensões, ele é muito *rápido* e *leve*.

A escolha perfeita para scripts leves e simples



# Spyder

Bem similar ao *R-Studio*, uma excelente alternativa para programadores R acostumados com ele.

Possui um visualizador de variáveis bem fácil e poderoso, ótimo para ter uma *Preview* dos dados.

Se interessou? Veja nosso vídeo sobre ele [aqui](#).



# PyCharm

---

Uma *IDE dedicada apenas ao Python*, feita pelo Jetbrains.

Bem *poderosa*, porém, bem *pesada*, com *inúmeras funcionalidades*.

Geralmente usada em projetos grandes que não envolvam visualização de dados, como, por exemplo, *Desenvolvimento Web*.



# Jupyter

---

Muito parecido com o Colab, afinal, o *Jupyter foi a inspiração para o Google Colab.*

Trabalha com células de código e é muito utilizado em *Data Science e Machine Learning.*

Muito *poderoso, simples e organizado.*



---

# TÁ, MAS E O CERTIFICADO?

---



# ESTÁ AQUI!

---

Chegou a tão esperada hora do certificado...

Para isso, basta você [clicar aqui](#) e realizar a sua inscrição no evento.

Com isso, você pode baixar seu certificado de forma gratuita e automática



# DÁ UMA OLHADA NELE



A ByLearn certifica que

FELIPE CABRERA

---

concluiu com êxito a Jornada Python Faixa Preta, no período de  
05/10/2020 a 08/10/2020, com carga horária de 10 horas.

FCR Santos

---

**FELIPE CABRERA**  
(PROFESSOR)



# Passo 1

Na parte das *Inscrições* aperte no botão verde escrito *Realizar Inscrição*

*Inscrições*

Participantes

Grátis

até 16/10/2020

REALIZAR INSCRIÇÃO

Lembre-se de colocar  
dados verdadeiros

# Passo 2

Irá abrir uma nova tela, com um *Formulário*, basta inserir o seus dados.

Você tem *15 minutos* para isso.

Lembre-se de colocar  
dados verdadeiros

14:55

Fique ligado no tempo. Após esse prazo, esta reserva será cancelada.

## Dados do participante

Inscrição nº 1 - Participantes

Nome completo

Nome completo

Email

Email

Confirmar email

Confirmar email

Idade 

# Passo 3

Agora que você enviou o formulário,  
apenas *Confirme Seus Dados* e clique  
em *Continuar*.

Tá quaaaase...

Felipe Cabrera  
felipe@bylearn.com.br  
Inscrição nº 1 - Participantes

[Editar](#)

[Continuar →](#)

# Passo 4

Aqui você só precisa apertar em  
*Concluir Inscrição*.



Tá quaaaase...

# Passo 5

Pronto! Você está inscrito no evento,  
agora falta se inscrever nas *Atividades!*  
Para isso, clique em *Participar das  
Atividades* (o botão verde).

O mais difícil já foi, hehehehe!

## Jornada Python Faixa Preta

05/10/2020 – 08/10/2020

20:00 - 23:59

Este é um evento online



### Atividades

Crie [uma conta](#) para participar das atividades do evento, tenha acesso a programação completa e muito mais.

[Participar das atividades](#)

# Passo 6

Será solicitado que você crie uma conta, para isso, você só precisa colocar uma *Senha*.

Dica de senha: Shura\_Fofa

### CRIAR CONTA

País

Brasil

Nome completo

Felipe Cabrera

Este nome será utilizado em todos os documentos emitidos pela plataforma

Email

felipe@bylearn.com.br

Senha

Senha

Campo obrigatório

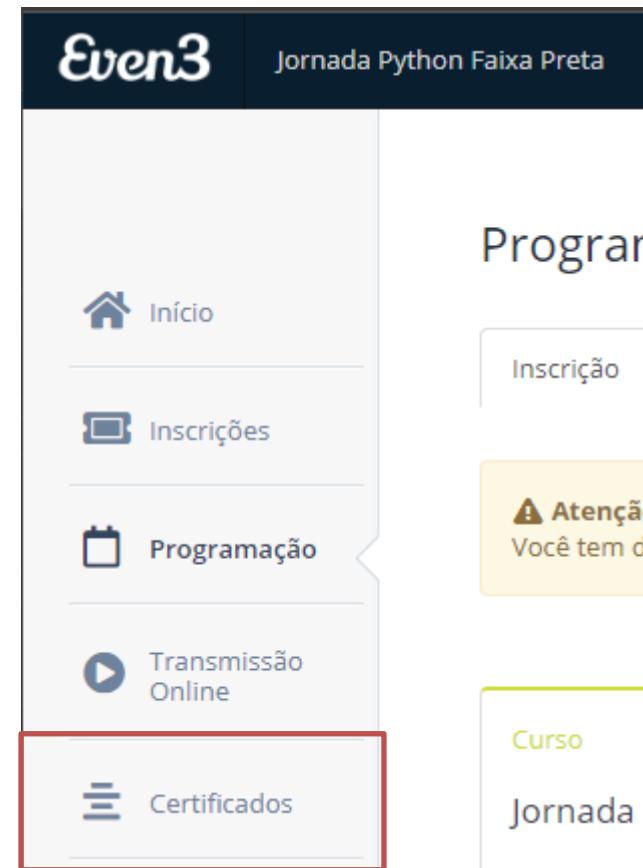
CRIAR CONTA

# Passo 7

Feito! Você já está *inscrito no evento e na atividade*.

Agora, no menu lateral esquerdo, clique em *Certificados*.

É agora... Preparem-se...



# Passo 8

Esse é o passo final!

*Agora é só você apertar em Download do Certificado.*

CONSEGUIMOOOOOS!!!

---

AÇÃO

---



Download do Certificado

# Só de garantia...

## Certificados

TÍTULO

Certificado de Participação

AÇÃO

 Download do Certificado

Em caso de dúvidas ou certificados faltando,  
entre em contato com a organização

---

MUITO OBRIGADO  
PELA PARTICIPAÇÃO

---



ByLearn



# ByLearn