

---

# JORNADA PYTHON

## FAIXA PRETA

---



**ByLearn**

# Seja Bem-Vindo!



**Felipe Cabrera**

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

---

# 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*.

## 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*.

# PROGRAMAÇÃO

## 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*.

## Python: O que é

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

## Python: Motivos

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

---

**BORA PRA  
PRÁTICA!**

---



# INSTALAÇÃO DO PYTHON

## Python x Anaconda

1) Site oficial:

<https://www.python.org/>

2) Pacote Anaconda:

<https://anaconda.org/>

A diferença é que o pacote Anaconda possui vários *adicionais inclusos*. Isso acelera na hora da configuração!



# PYTHON NA NUVEM

---

## Google Colab

Outra forma de se usar o Python é online, *sem precisar instalar nada*.

Para isso, podemos usar o *Colab*, que *utiliza a nuvem da Google* para executar nossos códigos.





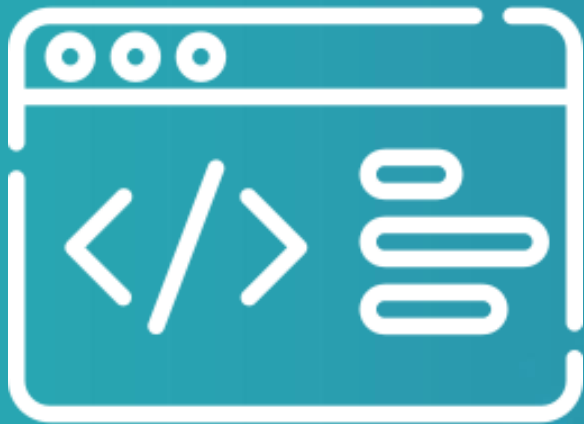
# O QUE É UMA *IDE*?

---

É um *ambiente de desenvolvimento integrado...*

Ok, agora em Português:  
É um *programa* que *auxilia* na hora do *desenvolvimento* (programação).





# O BÁSICO DA PROGRAMAÇÃO

**ÃO** Chegou a hora que tanto  
esperavam, não é mesmo?

# Vamos Codar?

A partir de agora iremos começar a aprender os assuntos básicos da programação para que você entenda os requisitos de um código de sucesso.

Caso você tenha *alguma dúvida*, não se assuste, *estamos aqui* para isso!

Iremos sempre *te ajudar*!



# SINTAXE

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

## Dicionário



**sintaxe**

/ss/

*substantivo feminino*

1. **GRAMÁTICA**

parte da gramática que estuda as palavras enquanto elementos de uma frase, as suas relações de concordância, de subordinação e de ordem.

2. **LINGÜÍSTICA**

componente do sistema linguístico que determina as relações formais que interligam os constituintes da sentença, atribuindo-lhe uma estrutura.

Definições de Oxford Languages

[Feedback](#)



Traduções e mais definições

# SINTAXE NA PROGRAMAÇÃO

Cada linguagem de programação possui a *sua sintaxe*, ou seja, a *sua forma de escrever*.  
Todos os exemplos abaixo fazem *a mesma coisa*, em *diferentes linguagens*.

```
1  class HelloWorld
2  {
3      public static void main(String[] args)
4      {
5          System.out.println("Hello World!!");
6      }
7  }
```

HelloWorld.java hosted with ❤ by GitHub

```
1  Module HelloWorld
2      Sub Main( )
3          System.Console.WriteLine("Hello world!")
4      End Sub
5  End Module
```

HelloWorld.vb hosted with ❤ by GitHub

```
1  #include<stdio.h>
2
3  int main(void)
4  {
5      printf("Hello, world!");
6      return 0;
7  }
```

HelloWorld.c hosted with ❤ by GitHub

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      cout << "\nHello World" << endl;
7      return 0;
8  }
```

HelloWorld.cpp hosted with ❤ by GitHub

```
1  using System;
2
3  namespace helloWorld
4  {
5      class HelloWorld
6      {
7          static void Main(string[] args)
8          {
9              Console.WriteLine("Hello World!");
10         }
11     }
12 }
```

HelloWorld.cs hosted with ❤ by GitHub

# AGORA EM PYTHON

```
1 print('Hello World')
```

hello\_world.py hosted with ❤ by GitHub

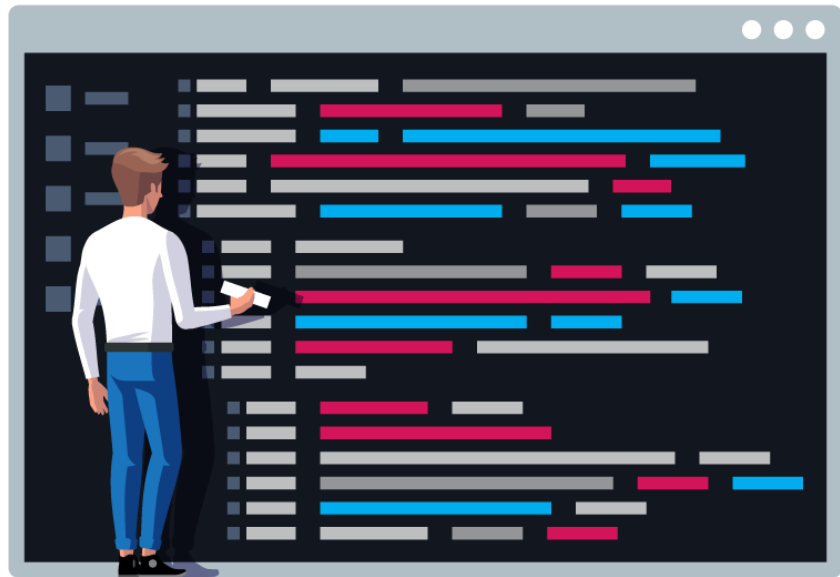
# Escrever em *Pythonês*

O python tenta ao máximo se aproximar da *linguagem de escrita humana* (como o *inglês*).

Ou seja, *esquerda pra direita* e de *cima pra baixo*, com *sintaxe* igual a de um *texto normal*.

Nós trabalhamos com níveis de *identação* (*espaçamento*).

*Nível* mais a *direita* *pertence* ao *nível* à sua *esquerda*.





Hoje eu brinquei com meus gatos:

- Doctor
- Pooky
- E Batman!

Também brinquei com a Shura, minha Furão.





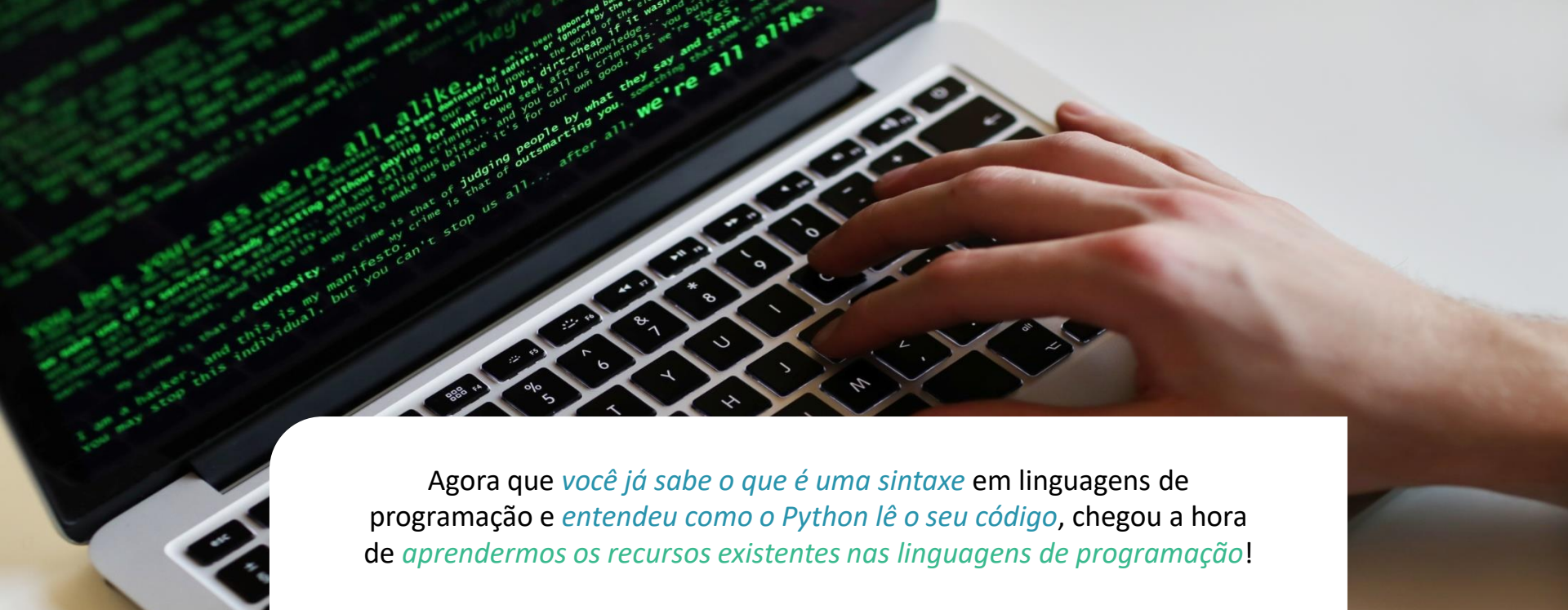
Eu estava com muita fome!

Fui na geladeira e achei o seguinte:

- Frutas:
  - Maçã
  - Pêra
  - Uva
- Doces:
  - Gelatina:
    - Framboesa
    - Morango
  - Sagu
  - Pudim

Decidi pegar a Gelatina de Morango...

Continuei com fome :(



Agora que *você já sabe o que é uma sintaxe* em linguagens de programação e *entendeu como o Python lê o seu código*, chegou a hora de *aprendermos os recursos existentes nas linguagens de programação!*

Daqui em diante você está na *parte prática do Python*.  
Agora é só *rodar esses códigos na sua IDE favorita!*

# Variáveis

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

Ou seja, é '*nome*' que podemos *atribuir* um *valor*.

Por exemplo, a variável *idade* vale *23*.

Ou seja, *atribuímos o valor* 23 para o *identificador* "Idade"



# Tipos de Dados

Cada dado irá possuir um *tipo*.

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

## *Variável x Tipo*

inteiro = *int*

decimal = *float*

texto = *string* (*str*)

verdadeiro\_falso = *booleano* (*bool*)

```
inteiro = 23
decimal = 10.5
texto = 'Felipe'
verdadeiro_falso = True
```

# Operações Matemáticas

Podemos executar todas as operações matemáticas no Python

É tão fácil que parece *Matemática!*

## *Operações x Simbolo*

soma = +

subtração = -

multiplicação = \*

divisão = /

```
soma = 10 + 10
subtracao = 15 - 10
multiplicacao = 15 * 3
divisao = 10 / 2

# Reparou na falta de acentos?
# Eles não são usados na programação!
```

# Métodos de Entrada

São formas de enviar dados do usuário ao Python.

*Por exemplo:* Posso pedir ao usuário digitar o nome e a idade dele.

Usamos a função *input* para isso, com uma *mensagem* (*opcional*).

Detalhe: *Converte* a entrada para o *tipo* de dado *desejado*, pois o *input* por padrão é uma *string* (*texto*).



```
nome = input("Insira seu nome")  
idade = int(input("Insira sua idade"))
```

# Métodos de Saída

São formas do Python de mostrar dados ao usuário.

*Por exemplo:* Posso mostrar o valor de uma soma.

Usamos a função *print* para isso.

Dentro do *print* iremos colocar o *texto* ou a *variável* que queremos mostrar a *valor*.

*PS: Separando* os dados *por vírgula*!

```
numero1 = 10
numero2 = 20

soma = 10 + 20

print("A soma vale", soma)
```



```
nome = input("Insira seu nome: ")

nota1 = float(input("Sua primeira nota: "))
nota2 = float(input("Sua segunda nota: "))

print("Olá", nome, "suas notas foram:")
print(nota1, "e", nota2)

soma = nota1 + nota2
media = soma / 2

print("Sua média é", media)
```

**Saída:**

*Olá Felipe suas notas foram:  
8.25 e 7.75  
Sua média é 8.0*



# Dados Sequenciais

Para trabalharmos com valores sequenciais podemos usar *listas*.

*Por exemplo:* Podemos colocar todos os itens para compra em uma lista.

Para criar uma lista, precisamos colocar o valor dentro de *Colchetes* `[]`.

PS: Separamos os dados por *vírgula*.  
Podendo ser *qualquer* tipo de *dado*!

```
lista_compras = ['arroz', 'feijão', 'carne']  
lista_compras2 = ['bolacha', 'biscoito', 'batata']  
lista_compras3 = ['alface', 'maçã', 'suco natural']
```

# ELEMENTOS EM LISTAS

<i>Lista</i>	Cachorro	Gato	Coelho	Furão	Pássaro	Peixe
--------------	----------	------	--------	-------	---------	-------

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

Enquanto *posições iniciam em 1* (“Primeira Posição”, “Segunda Posição”)...

Os *Índices iniciam em 0* (“Índice Zero”, “Índice Um”)...

<i>Elemento</i>	Cachorro	Gato	Coelho	Furão	Pássaro	Peixe
<i>Posição</i>	Primeira	Segunda	Terceira	Quarta	Quinta	Sexta
<i>Índice</i>	<i>Zero</i>	<i>Um</i>	<i>Dois</i>	<i>Três</i>	<i>Quatro</i>	<i>Cinco</i>

# ELEMENTOS EM LISTAS

Ou seja, os índices podem ser entendidos como:

$$\text{Índice} = \text{Posição} - 1$$

Elemento	Cachorro	Gato	Coelho	Furão	Pássaro	Peixe
Posição	1	2	3	4	5	6
Índice	0	1	2	3	4	5

Dessa forma, o *Furão* é o *índice 3* dessa nossa lista.

Para *pegar um elemento pelo índice*, devemos colocar o *índice entre colchetes*:

```
furao = lista[3]
```



```
animais = ['Cachorro', 'Gato', 'Coelho', 'Furão', 'Pássaro', 'Peixe']  
  
furao = animais[3]  
gato = animais[1]  
  
print("A Shura é uma", furao)  
print("O Doctor é um", gato)
```

**Saída:**


*A Shura é uma Furão  
O Doctor é um Gato*

---

# **BÔNUS:** ***CONDICIONAIS***

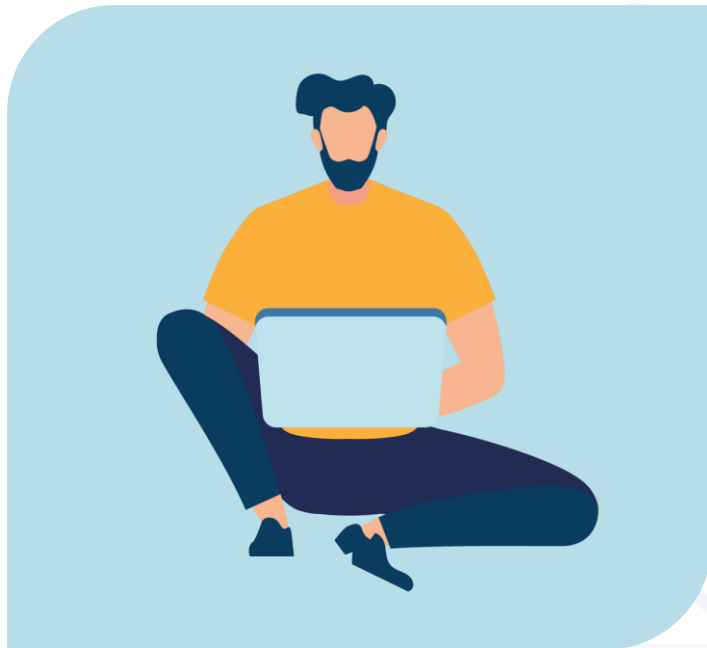
---

# **S**



# CONDICIONAIS

---



Praticamente todo código em algum momento terá que tomar uma decisão.

*“Se isso... então aquilo!”*

*“Senão, se... então isso!”*

*“Senão... Então isso outro!”*

Em outras palavras (de 99% das mães):

*“Se estiver nublado, então leve guarda-chuva”*

*“Senão, se estiver frio, então saia agasalhado”*

*“Senão, então você que sabe...”*

# CONDICIONAIS

---



Para isso, nós usaremos as palavras-chaves:

*If => se*

*Elif => senão, se*

*Else => senão*

Na frente da palavra-chave você deve por a *expressão da sua condição* e um *dois pontos*:

**Se o nome for Felipe**

*if nome == 'Felipe':*

**Se a idade for maior ou igual a 23**

*if idade >= 'Felipe':*

# CONDICIONAIS

---



Continuando, nós devemos lembrar o que vimos sobre *identação*, lembra?

No Python falamos que *um bloco de código pertence a outro* usando *níveis de indentação*.

Onde o *mais a direita pertence ao mais a esquerda*.

Ou seja, damos um espaçamento (ou tab) nas linhas de código que pertence a essa condição:

```
if nome == 'Felipe':  
    print("É o Felipe!")
```





```
horario = 'noite'

if horario == 'dia':
    print('O sol está quente')
elif horario == 'tarde':
    print('O por do sol está chegando')
else:
    print('A lua está linda')
```

*Saída:*

*A lua está linda*



```
temperatura = 22

if temperatura <= 30:
    print('Não está tão quente')
else:
    print('Está muito quente!')
```

**Saída:**

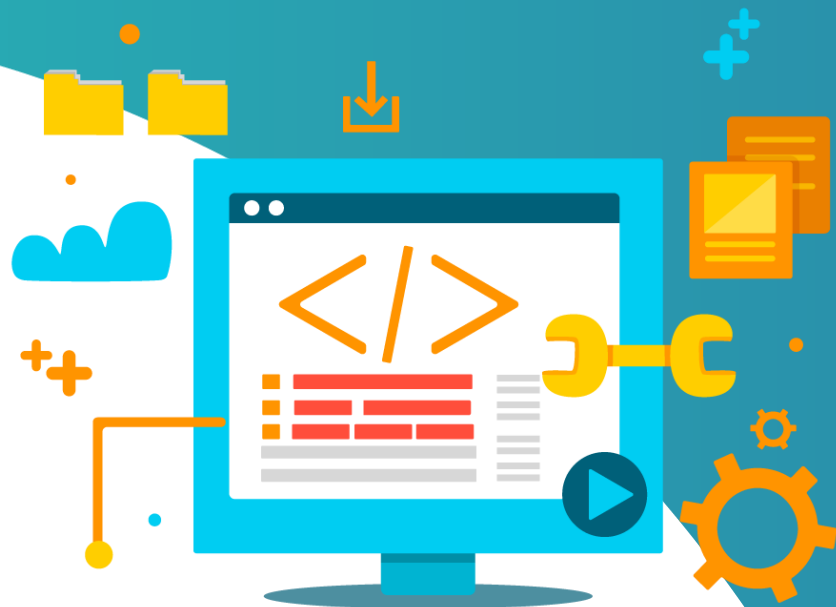
*Não está muito quente*



**GITH  
UB**

# GITHUB

---



- É a principal plataforma para *hospedagem e compartilhamento de códigos* (de programação) do mundo!
- Possui mais de *40 milhões* de programadores e *100 milhões* de projetos.
- Projetos podem ser *públicos ou privados*, o desenvolvedor escolhe.
- Ótimo para *encontrar* projetos e *estudar*.

# COMPARTILHAR

No GitHub você pode compartilhar seus códigos com a comunidade. Programadores encontrarão seu código e irão usá-los em projetos ou em estudo.

# DIVULGAR

Você também pode usar como forma de divulgar seus conhecimentos. Diversas empresas verificam o GitHub dos candidatos para ver seus projetos e validar se sabem mesmo programar.



# EXEMPLOS DE PROJETOS

## Inteligência Artificial

- [Reconhecer Captcha](#)
- [Processamento de Linguagem Natural](#)

## Ciência de Dados

- [Recomendação de filmes Netflix](#)
- [Duração de corrida de taxi NY](#)

## Web

- [Blog](#)
- [Clone do Twitter](#)

## Nuvem

- [Boto3 \(AWS\)](#)
- [Google Cloud Client](#)

## Mobile

- [Python Brasil App](#)
- [Lista de Tarefas](#)

## Interfaces Gráficas

- [Calculadora](#)
- [Loja de Livros](#)

## Banco de Dados

- [Controlar Equipamento Médico](#)
- [Agenda de Contatos](#)

## Automação

- [Envio de E-mail de Lembretes](#)
- [Automação Residencial](#)

## Interação com Office

- [Gerador de DocX usando Templates](#)
- [Importar dados do Excel](#)

## Cyber Segurança

- [Detector de Phishing](#)
- [Analisador de Dados de Rede](#)

## Expressões Regulares

- [Analisar Vulnerabilidade em .PHP](#)
- [Detector de Spam](#)

## Jogos

- [Jogo da Velha](#)
- [Xadrez](#)

# NOSSAS REDES SOCIAIS

## Conteúdos Diários

Deseja conteúdos diários, gratuitos e de qualidade sobre Python?

Então siga-nos nas redes sociais:

- [Facebook](#)
- [Instagram](#)
- [YouTube](#)





# ByLearn

Obrigado por escolher nossos cursos!