
MARATONA PROFISSÃO PROGRAMADOR PYTHON



ByLearn

Seja Bem-Vindo!



Felipe Cabrera

Graduado em Ciência da Computação e Mestre 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*.

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*.

PYTHON

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

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*





CRONOGRAMA DA MARATONA

C R O N O G R A M A

O QUE É PROGRAMAÇÃO E QUAIS SÃO AS
OPORTUNIDADES TRABALHANDO COM O PYTHON



INICIANDO NA PROGRAMAÇÃO COM PYTHON:
CRIANDO NOSSO PRIMEIRO PROGRAMA



AVANÇANDO NA PROGRAMAÇÃO:
CRIANDO LÓGICA PARA SOFTWARES



C R O N O G R A M A

RUMO AO PROGRAMADOR FAIXA PRETA



PROJETO PRÁTICO + SEGREDO PESSOAL:
COMO GANHAR 5X MAIS EM OUTRA MOEDA



CONSTRUINDO UMA SOLUÇÃO 100%
PRÁTICA E REALISTA AO MERCADO



HORA DO SPOILER: NOSSAS ÚLTIMAS AULAS

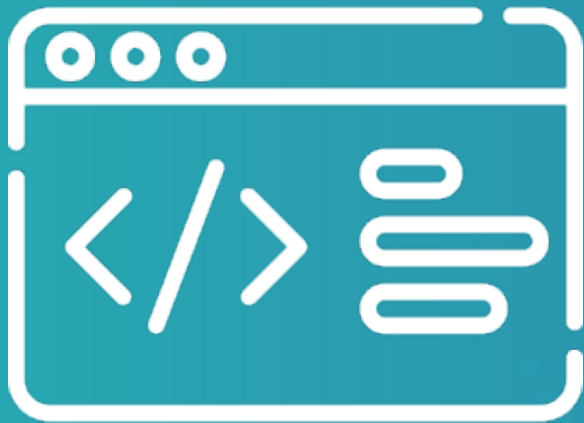
Projeto 1:

R\$ 764,10 (\$ 135,00)

Projeto 2:

R\$ 1017,00 (\$ 180,00)

Total:
R\$ 1781 (\$ 315,00)



O INTERMEDIÁRIO DA PROGRAMAÇÃO

*Bora continuarmos a aprender a
dominar o Python?*

CONDICIONAIS



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

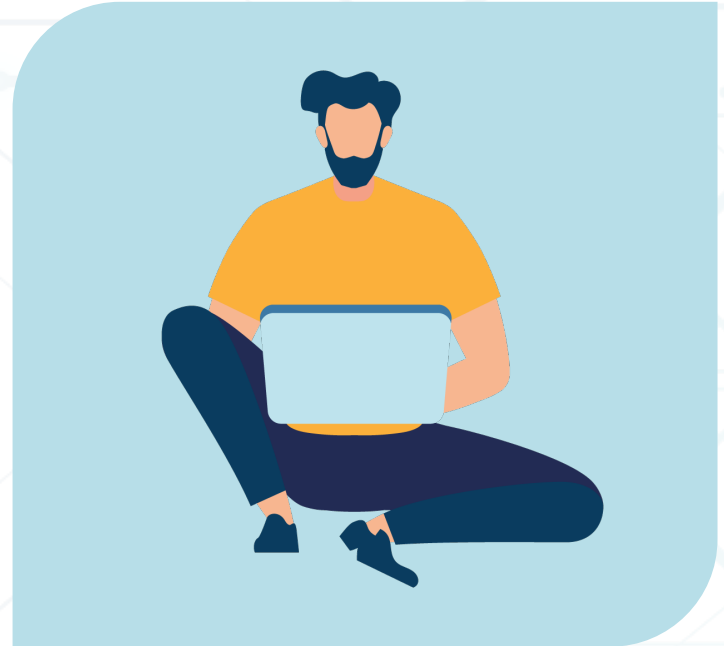
Na frente da palavra-chave você deve por a *expressão da sua condição* e um *dois pontos*:

Se o nome for Felipe

if nome == 'Felipe':

Se a idade for maior ou igual a 23

if idade >= 23:



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

Operadores Lógicos

Podemos usar operadores lógicos para criar *condicionais compostas*.

Por exemplo: Verificar se é de manhã *e* está chovendo ou se é sábado *ou* domingo.

Podemos usar *and* para que *todas* sejam verdade.

Podemos usar *or* para que *ao menos* uma seja verdade.

```
dia = 'sabado'
horario = 'manha'
chovendo = False

if horario == 'manha' and chuvendo == False:
    print("O sol está lindo")

if dia == 'sabado' or dia == 'domingo':
    print("É fim de semana")
```

Laços de Repetição

Excelente forma de evitar repetição de código.

Por exemplo: Mostrar todos os números até 10.

Para usar laços de repetição nós podemos usar o *for*.

PS: Como na programação iniciamos em 0, *range(10)* vai de 0 até 9.

```
for numero in range(10):  
    print(numero)  
    # De 0 até 9  
  
for numero in range(1,11):  
    print(numero)  
    # De 1 até 10
```

Tamanho de Listas

Podemos também pegar o tamanho de listas e usá-los no nosso código.

Por exemplo: Pegar quantos nomes tem na lista e mostrar um a um (pelo índice).

Sabermos o tamanho de uma variável torna nosso código mais dinâmico.

No Python nós usamos a função *len* para pegar o tamanho da sequência.

```
nomes = ['Felipe', 'Fabiana', 'Paulo', 'Katia']  
  
tamanho = len(nomes)  
  
for indice in range(tamanho):  
    print("O nome no índice", indice, "é", nomes[indice])
```

Repetições Dinâmicas

Além de usar o tamanho da variável, podemos também usar a própria sequência na repetição.

Por exemplo: Para todas as *espécies* da lista *animais* eu quero *printar* a *espécie*

Para isso, usamos o *for* com a *sequência* que queremos trabalhar, como:

for elemento in sequencia:

```
animais = ['Gato', 'Furão', 'Cachorro', 'Coelho', 'Pássaro', 'Peixe']

for animal in animais:
    print("Espécie:", animal)

numeros = [5, 10, 15, 20, 25, 30]

for numero in numeros:
    print("O valor", numero, "é múltiplo de 5!")
```


Funções

Outra forma de evitar repetição de código é através de funções, onde definimos uma rotina para ser executada.

Por exemplo: Definir o que fazer para verificar se um aluno foi aprovado.

Para definirmos uma função usamos o *def*.

Funções podem *retornar valores*.

Podemos *enviar valores* para funções.

Funções

Outra forma de evitar repetição de código é através de funções, onde definimos uma rotina para ser executada.

Por exemplo: Definir o que fazer para verificar se um aluno foi aprovado.

Para definirmos uma função usamos o *def*.

Funções podem *retornar valores*.

Podemos *enviar valores* para funções.

```
# Definimos a função
def mostrar_nome_felipe():
    print('Felipe')

# Função com Parâmetro
def mostrar_nome(nome):
    print(nome)

# Função com retorno
def proximo_numero(numero):
    return numero + 1

# Chamando as funções:
mostrar_nome_felipe()
mostrar_nome('ByLearner')
cinco = proximo_numero(4)
```

```

# Definimos a funcao da media
def calcular_media(nota1, nota2):
    soma = nota1 + nota2
    media = soma / 2

    return media

# Definimos a função da Aprovacao
def verificar_aprovacao(media):

    if media >= 6:
        print('O Aluno foi aprovado')
    else:
        print('O Aluno foi reprovado')

# Chamamos a função Calcular Média
# Enviamos as notas 7.75 e 4.5
# Pegamos o valor do seu retorno...
# ... e colocamos em media_aluno
media_aluno = calcular_media(7.75, 4.5)

# Chamamos a função Verificar Aprovação
# Enviamos a média calculada acima
verificar_aprovacao(media_aluno)
```

EXEMPLOS PRÁTICOS





```
nota1 = 7.5
nota2 = 4.8

# Definimos o que é "Verificar uma Aprovação"
def verificar_aprovacao():
    media = calcular_media([nota1, nota2])

    if media >= 6:
        print("O Aluno Foi Aprovado!")
    else:
        print("O Aluno Foi Reprovado")

# Definimos o que é "Calcular a Média"
def calcular_media(notas):
    quantidade = len(notas)

    soma = 0
    for nota in notas:
        soma = soma + nota

    media = soma / quantidade

    return media

# Chamamos (executamos) a função de Verificar Aprovação
verificar_aprovacao()
```



```
frutas = ['Maça', 'Banana', 'Pera', 'Uva']
guloseimas = ['Bolacha', 'Batata', 'Fini', 'Chocolate']
comidas = ['Arroz', 'Feijão', 'Carne']
bebidas = ['Refrigerante', 'Suco de Laranja', 'Água']

categorias = ['Frutas', 'Guloseimas', 'Comidas', 'Bebidas']
compras = [frutas, guloseimas, comidas, bebidas]

for indice, categoria in enumerate(categorias):
    print('Você precisa comprar', len(compras[indice]), categoria+':')
    for compra in compras[indice]:
        print('-', compra)
```

```
def validar_idade(idade):
    if idade < 18:
        print('\nDesculpe, você não tem idade para prosseguir,', nome)
        return False
    else:
        print('\nÓtimo! Podemos prosseguir,', nome)
        return True

def escolher_carta():
    print("Digite uma das opções abaixo:")
    print("1 - Carro\n2 - Moto\n3 - Carro e Moto")

    return int(input())

def calcular_preco(escolha):
    valor_carro = 1500
    valor_moto = 1000

    if escolha == 1:
        return valor_carro
    elif escolha == 2:
        return valor_moto
    else:
        return valor_carro + valor_moto

def desconto(valor):
    return valor - (valor * 0.10)

nome = input('Digite o seu nome: ')
idade = int(input('Digite sua idade'))

if validar_idade(idade):
    escolha = escolher_carta()

    print('\nPerfeito! Vou calcular o valor')
    valor = calcular_preco(escolha)

    print('\n'+nome, 'o valor total é de', valor, 'reais')
    print('Mas vou ver com meu gerente se posso dar um desconto...')
    valor = desconto(valor)

    print('\nCom desconto eu consigo fazer por', valor, 'reais.')

    print('Te interessa?\n1 - Sim\n2 - Não')
    interesse = int(input())
    if interesse == 1:
        print('\nPerfeito! Começaremos amanhã!')
    else:
        print('\nTudo bem :(Me avise se mudar de ideia.')
```



```
animais = []

animal = input("Digite o nome do seu animal de estimação ou digite 0 se não tiver nenhum: ")

while animal != '0':
    especie = input("Digite a Espécie desse animal: ")
    animais.append([animal, especie])
    animal = input('Se tiver mais animais, digite o nome dele. Ou digite 0 se não tiver: ')

if len(animais) == 0:
    print('\n\nVocê não tem animais')
else:
    print("\n\nVocê tem os seguintes animais:")
    for animal in animais:
        print("- Nome:", animal[0], "| Espécie:", animal[1])
```




```
horario = 'manhã'
clima = 'ensolarado'
temperatura = 'quente'

if horario == 'manhã' or horario == 'tarde':
    if clima == 'ensolarado' and temperatura == 'quente':
        print("Uma piscina cairia bem")

    if (clima == 'ensolarado' or clima == 'nublado') and (temperatura == 'amena' or temperatura == 'frio'):
        print("Seria legal praticar algum esporte")

    if clima == 'chuvoso':
        print("Aproveite para treinar seu Python")
else:
    if clima == 'chuvoso':
        print("Que tal um filme, série ou jogatina?")
    else:
        print("Um jantar fora parece interessante...")
```



```
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*

O que você vai receber?

✓ **Mentoria de 03 meses com acompanhamento e desafios semanais**

✓ **A trilha gravada do método triplo01**

Curso completo de Python do Zero a Pro

Git com Github

Regex com Python

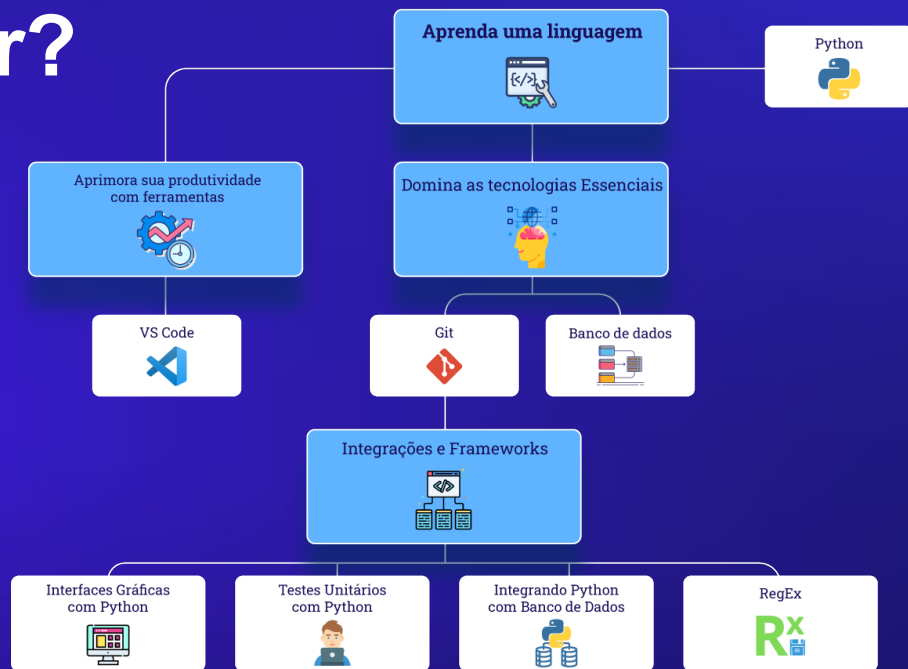
Vscode produtividade infinita

Banco de dados

Testes Unitários

Interfaces gráficas

Todo conteúdo conta com Apostila



Bônus

✓ 05 Masterclass comigo e outros Profissionais

1. Mercado de Trabalho
2. Personal Branding
3. Marketing para Programadores
4. Dev. Gringo (Receba 5x mais - Ganhe em dólar)
5. Desenvolvimento de Software (Renda Passiva)



Bônus

✓ **Módulo Avançado: Expert em Python**

Integração com Word

Integração com Excel

Integração com PDF

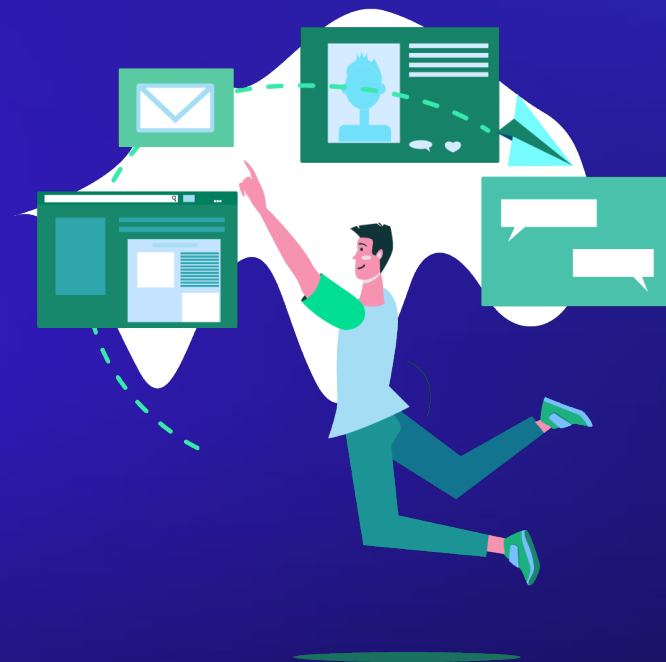
Integração com Sistemas Operacionais

Boas Práticas com Python

✓ **Grupo de Networking**

✓ **Certificado**

✓ **Garantia de 15 dias**



~~De 1497~~
por apenas
12x 116,62
1197

Acesso a todo material durante 02 anos. Podendo acessar por celular,
tablet e computador e assistir mesmo sem internet no celular

BÔNUS 20 PRIMEIROS

ATENÇÃO

- ✓ Os 20 primeiros Mentorandos terão um ***SUPER BÔNUS***
- ✓ Acesso particular e individual a uma reunião exclusiva comigo com tema totalmente livre *por 30 minutos!*



DUPLA GARANTIA

Se dentro de 03 meses você fizer tudo o que eu mostro e não aprender nada eu devolvo o seu dinheiro e te pago ainda mais **R\$ 500,00** como pedido de desculpas.



DUPLA GARANTIA

Se dentro de 03 meses você fizer tudo o que eu mostro e não aprender nada eu devolvo o seu dinheiro e te pago ainda mais **R\$ 500,00** como pedido de desculpas.

Mas você terá que me provar que assistiu todas as aulas e participou de todas as mentorias ao vivo e não aprendeu nada de programação.





ByLearn

Obrigado por escolher nossos cursos!