

# Python3

---

Seja bem vind@ ao modulo python  
básico

Feito por Ysrael Sacrati

Referencias : Aprenda Python Básico Rápido e Fácil de entender por Felipe Galvão ,  
w3schools, stackoverflow, cs50

# O que vamos aprender neste curso?

---

- instalação
- início
- Variáveis e dados
- condicionais
- Loops
- Funções

# Mas Python serve para?

---

- Desenvolvimento Web
- Sistemas embarcados
- Desktop
- App mobile
- IA (inteligencia artificial)
- Automação de processos
- Ciência de dados
- Infinitas possibilidades com código simples



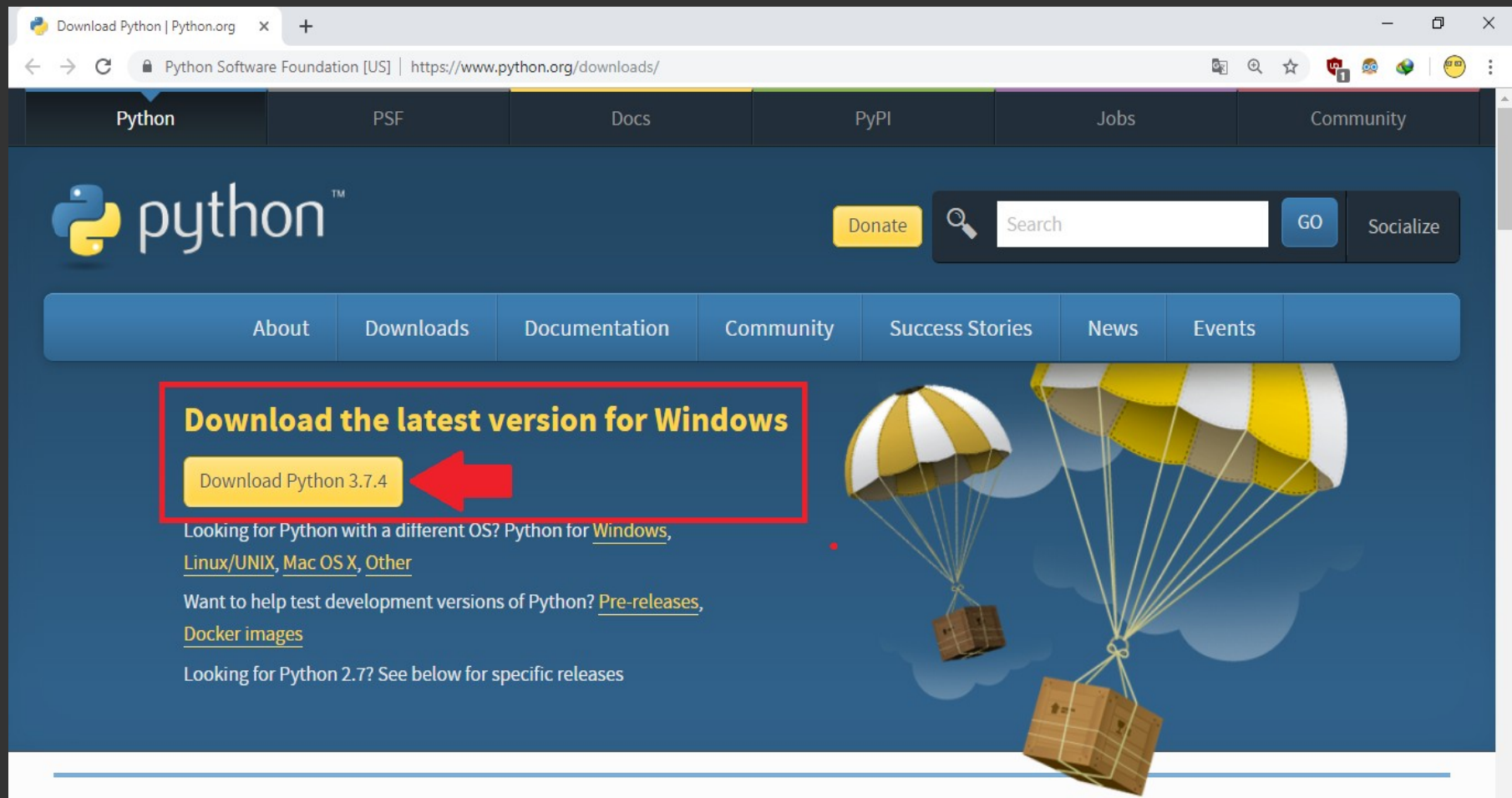
# Por que aprender Python?

---

Dadas a explosão de usos do Python, há muitas razões do porquê que aprender Python é um grande investimento do seu tempo. Não somente o Python é extremamente versátil e útil, está também em alta demanda por profissionais, com ótimos salários, e mais.

# Instalando no windows

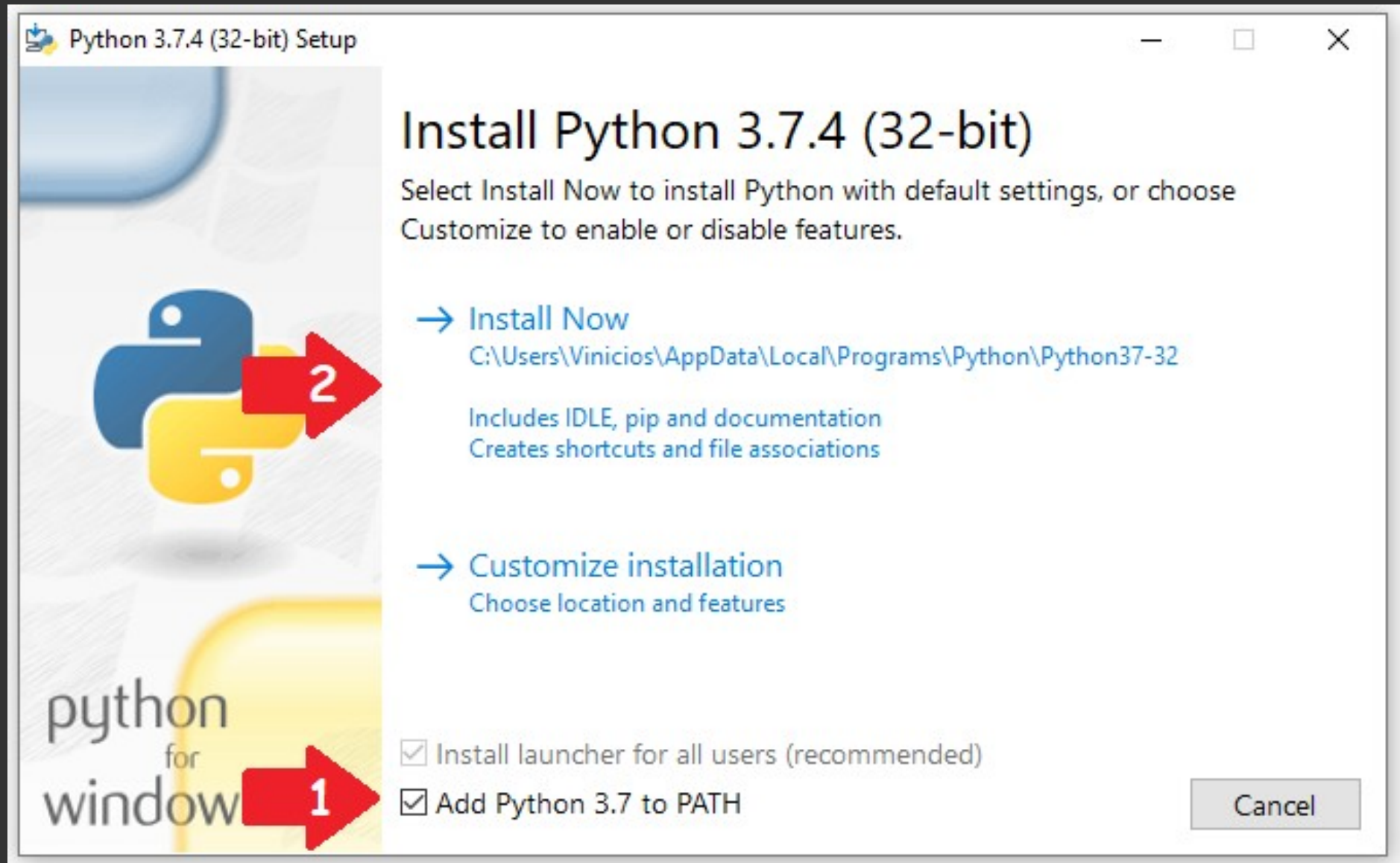
Para instalar o Python no seu sistema operacional Windows, você precisa baixar o instalador. Acesse o site oficial e clique em download.



O processo de instalação é bem simples.

1. Marque a opção "Add Python to PATH"

2. Clique em "Install Now"



# Instalando no linux

---

Apenas digite `sudo apt-get install python3`  
no teu terminal



# Testando o python

---

- Para verificar se a instalação do Python foi bem-sucedida, pesquise no menu iniciar por "cmd" terminal no linux e clique duas vezes para abri-lo.
- Digite o seguinte comando:  
`python --version`

Este comando retornará a versão do python que está instalada em sua máquina.

# IDE

- Visual studio code usaremos a seguinte extensão.



# Vamos codar!!

---

- Como bons aprendizes de programação vamos aprender a exibir uma mensagem com python.
- Para fazer isso digitamos o seguinte código

```
Print (“ola mundo”)
```

# Mas que raios acabamos de fazer??

---

Nos usamos a função print que basicamente, “imprime” na tela, ou seja, exibe, aquilo que está dentro dos parenteses.

Essa função já esta definida na linguagem python.

# Mas que diacho é função??

---

Uma função no contexto da programação é pedaços de códigos predefinidos que podem ser reaproveitados.

Como fizemos anteriormente usando a função `print()` para exibir a nossa mensagem .

# Comentários no código

---

Comentários são auto-explicativos. São anotações no código que não são rodadas junto com o próprio, e servem para ilustrar, explicar ou incluir informações adicionais nele.

Para definirmos um comentário de uma linha em python nos usamos o '#'. Já os de varias linhas usamos '''.

```
# isso é um comentario
```

```
'''
```

```
isso é um comentario
```

```
'''
```

# Tipos de Dados

---

- Strings - são dados definidos entre aspas duplas (") ou simples ('). ex : "ola mundo"
- Números Inteiros (Integer) - são números sem casas decimais. Ex : 1, 2, 3
- Números decimais (float) - números com casas decimais. Ex : 1.50, 2.40, 10.90
- Boolean - utilizado para representar os conceitos de Verdadeiro ou Falso. Ex : True & False

# variáveis

---

variáveis são elementos que nos permitem guardar valores para reutilizar posteriormente.

Para definir uma variável no Python, basta digitar um nome para a própria, o sinal de igual e o valor que se deseja que ela armazene.

```
Mensagem = 'ola mundo '
```



# A função input()

---

Esta função é utilizada para captar informações enviadas pelo usuário. Ela pode receber um parâmetro, que será uma mensagem exibida ao usuário antes que ele insira o que deseja.

```
nome = input("Olá, qual o seu nome?\n")  
print("Olá + " " + nome)
```

# Exercício 1

---

Jã que nos aprendemos o básico de python escreva o seu primeiro programa.

O programa pega o nome e sobrenome do usuário e depois exhibi uma mensagem com o nome e o sobrenome.

# Sinais matemáticos

---

- + soma
- - subtração
- \* multiplicação
- / divisão
- % resto da divisão
- \*\* potenciação

# Exercício 2

---

Sabendo os sinais e variáveis , escreva um programa em que calcula uma media.

# Listas

---

são estruturas de dados sequenciais do Python que servem para armazenar outros diferentes tipos de dados. As listas podem armazenar qualquer tipo de dado, como números inteiros, strings, números decimais e até outras listas

Para criar uma lista, basta definir seus itens entre colchetes “[ ]” e separados por vírgula. Também é possível criar uma lista vazia, definindo apenas os colchetes, sem nenhum item dentro deles.

```
alunos = ["José", "João", "Luiz"]
```

# Exercício 3

---

Faça um Programa que leia um vetor(lista) de 5 números inteiros e mostre-os.

# Tuplas

---

As tuplas são bem parecidas com as listas. A grande diferença é que ao contrário de uma lista, uma tupla é imutável(constante).

Você não pode modificar os dados que esta armazenado em uma tupla.

```
tupla1 = ("Gato", "Cachorro", "Papagaio", "Tartaruga")
```

```
# para definir uma tupla usamos () em vez de []
```

# Dicionários

---

Os dicts mais uma estrutura para armazenar dados. A diferença dele para a lista é que o índice é um nome que você define, ao invés dos índices numerados que as listas possuem.

Para criar um dicionário, usamos as chaves (`{}`). Os elementos seguem o formato “chave” : “valor”. Vejamos no exemplo para que fique mais claro:

```
aluno = {"nome": "José", "idade": 20, "nota": 9.2}
```



# Sets

---

Os sets são listas ou coleções não ordenadas de itens únicos.

```
lista1 = ["Luiz", "Alfredo", "Felipe", "Alfredo", "Joana", "Carolina", "Carolina"]
set1 = set(lista1)
print(set1)
set2 = {"cachorro", "gato", "papagaio", "cachorro", "papagaio", "macaco", "galinha"}
print(set2)
> {'Felipe', 'Alfredo', 'Joana', 'Carolina', 'Luiz'}
> {'cachorro', 'papagaio', 'gato', 'galinha', 'macaco'}
```

# Comparações em python

Sinal	Operação	Exemplo	Definição
==	Igual a	>>> 4 == 5 False >>> 4 == 4 True	Indica se dois valores são iguais
!=	Diferente de	>>> 6 != 3 True >>> 5 != 5 False	Indica se dois valores não são iguais
>	Maior que	>>> 9 > 8 True >>> 3 > 5 False	Indica se o número anterior ao sinal é maior que o posterior
>=	Maior ou igual a	>>> 10 >= 10 True >>> 13 >= 15 False	Indica se o número anterior ao sinal é maior ou igual ao posterior
<	Menor	>>> 63 < 65 True >>> 150 < 140 False	Indica se o número anterior ao sinal é menor que o posterior
<=	Menor ou igual a	>>> 5 <= 5 True >>> 12 <= 8 False	Indica se o número anterior ao sinal é menor ou igual ao posterior

# Estruturas de Controle(if)

---

Estruturas de Controle são utilizadas para organizar e adicionar complexidade à um programa.

O “ if” é uma estrutura de controle que executa um determinado bloco de código baseado em uma condição. Se a condição estipulada for verdadeira, o bloco logo abaixo será executado.

```
Nome = “Mario”  
If Nome == “mario” :  
    print (“cade o Luigi”)
```

# Estrutura de Controle (else)

---

O else é sempre executado se a condição estipulada no if não for verdadeira:

```
nome = input("Olá, qual o seu nome?\n")
```

```
if nome == "Felipe":  
    print("Olá," + nome)
```

```
else:  
    print("Olá, visitante")
```

# Estruturas de Controle (elif)

---

Podemos usar o “elif” quantas vezes quisermos, para testar quantas condições forem necessárias.

```
if nome == "Felipe":  
    print("Olá, Felipe")  
elif nome == "João":  
    print("Oi, João")  
elif nome == "Carlos":  
    print("E aí, Carlos?")  
else:  
    print("Olá, visitante")
```

# Exercício 4

---

Crie um algoritmo em PYTHON que leia a idade de uma pessoa e informe a sua classe eleitoral:

- - não eleitor (abaixo de 16 anos);
- - eleitor obrigatório (entre a faixa de 18 e menor de 65 anos);
- eleitor facultativo (de 16 até 18 anos e maior de 65 anos, inclusive).

# Loops

---

Os loops são estruturas de controle para repetição

# Loops - for

---

Imagine se eu quisesse escrever um programa que imprima todos os números de 1 a 10, ou que imprima separadamente cada item de uma determinada lista. Você poderia criar uma linha com um print para cada número, ou para cada item. Mas isso seria repetitivo, tedioso. Tem de haver uma maneira melhor de fazer isso. E têm. Para facilitar, o Python possui o loop “for”.



# Loops - while

---

O loop while é semelhante ao for, no sentido que ele leva a repetição de um bloco de código. Entretanto, ele executa este código enquanto uma certa condição definida for verdadeira.

# Funções

---

Nessa aula entraremos em detalhes sobre como funcionam as funções no Python. Como já vimos no início do curso, funções são blocos de código que podem ser reutilizados quantas vezes forem necessárias, para que você não tenha que ficar escrevendo o mesmo código várias vezes.

# Definindo uma função

---

A estrutura básica para definir uma função é através da declaração “def”, depois o nome da função e parênteses.

- Vamos lá para o vscode !!