

Universidade do Vale do Itajaí  
Escola do Mar, Ciência e Tecnologia  
Engenharia de Computação

# Introdução ao Python

**Douglas A. dos Santos - [douglasas@edu.univali.br](mailto:douglasas@edu.univali.br)**  
LEDS - Laboratory of Embedded and Distributed Systems

# Agenda

- Introdução
  - Instalação
  - Tipos de dados
  - Operações
  - Controle de fluxo
  - OpenCV
-

# Introdução

# História do Python

- Criada pelo holandês **Guido Van Rossum**
- Começou a criar no seu tempo livre em Dezembro de 1989
- Começou como um descendente da linguagem de programação ABC
- Nome baseado na série de comédia *The Monty Python's Flying Circus*
- Primeira versão em 1990



fonte: [https://python.swaroopch.com/about\\_python.html](https://python.swaroopch.com/about_python.html)

# Vantagens

- Python é fácil de aprender, altamente legível e fácil de usar
- Linguagem minimalista
- Suporte a avançados paradigmas de engenharia de software
- Códigos menores que os de outras linguagens
- Multiplataforma
- Grande quantidade de bibliotecas
- Integração de componentes de diferentes linguagens

- Desenvolvimento de aplicações WEB
  - Django, Flask, CherryPy e Bottle
- Computação científica e numérica
  - NumPy, SciPy, Pandas, matplotlib, e IPython
- Programação de GUI
  - wxWidgets, PyQt, PySide, Tkinter
- Prototipação de software
  - Pygame

# Características básicas

- Case-sensitive
- Variáveis não precisam ser declaradas
- Sem ponto-e-vírgula
- Blocos de código delimitados de acordo com a indentação

# Instalação

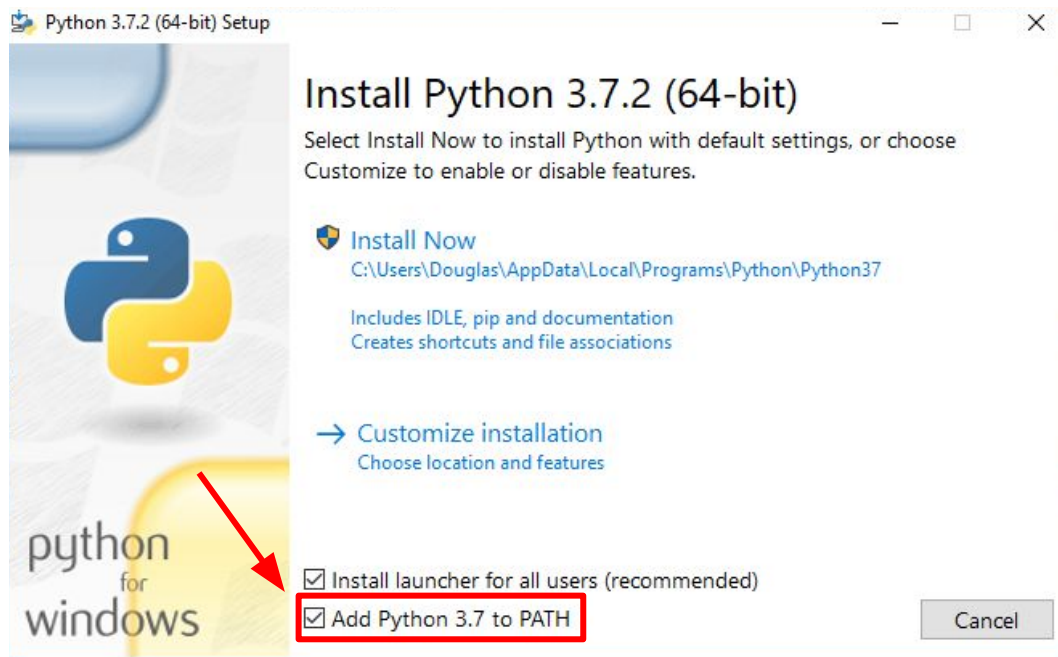


# Instalação no Linux

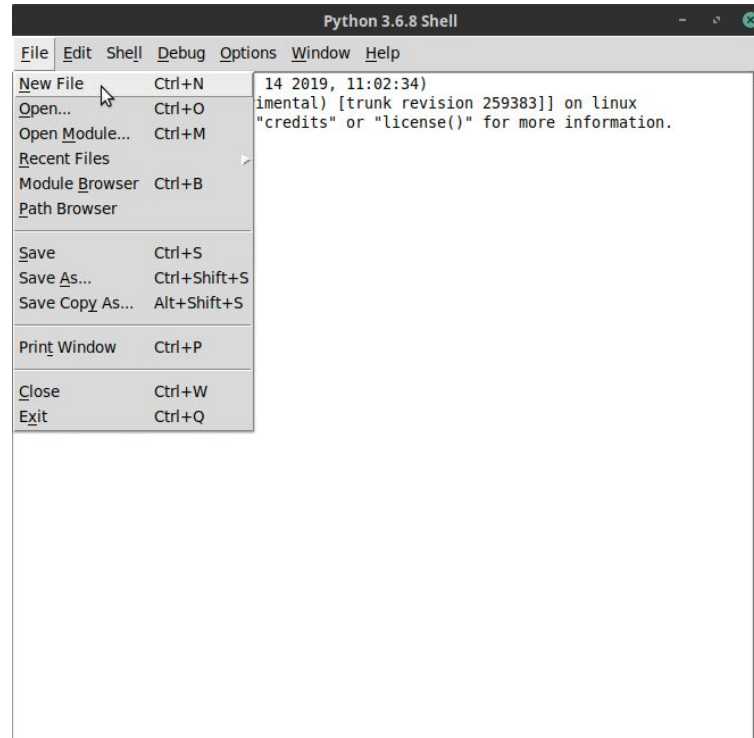
```
sudo apt install python3 pip3
```

# Instalação no Windows

Download no link: [python.org/downloads/](https://python.org/downloads/)



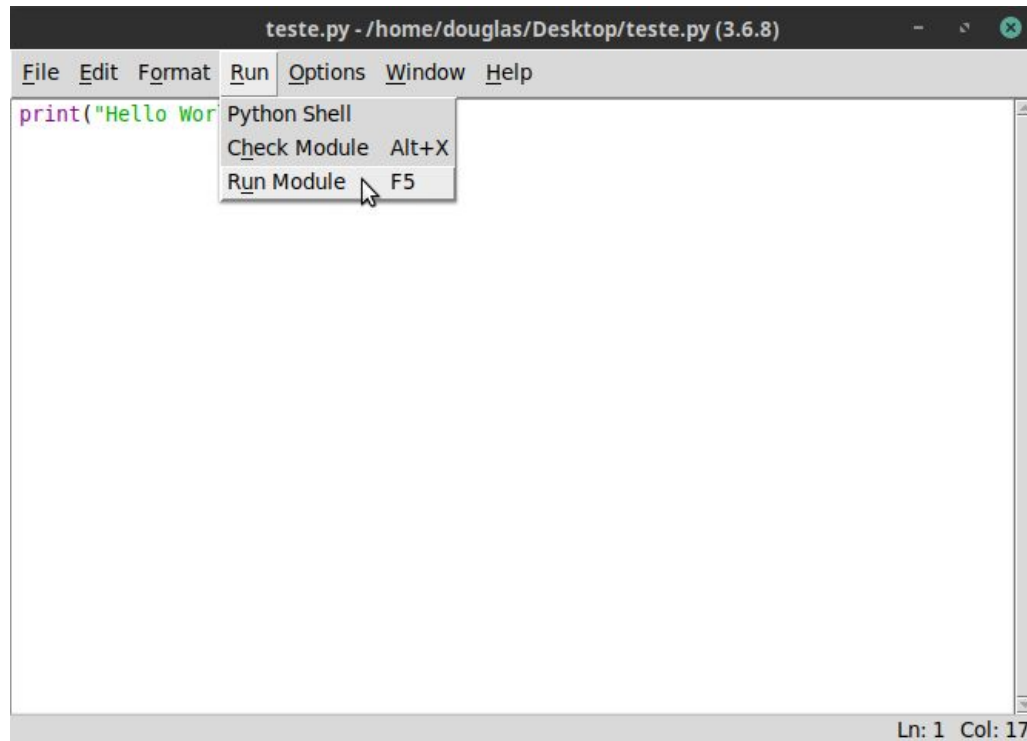
# Crie um novo arquivo



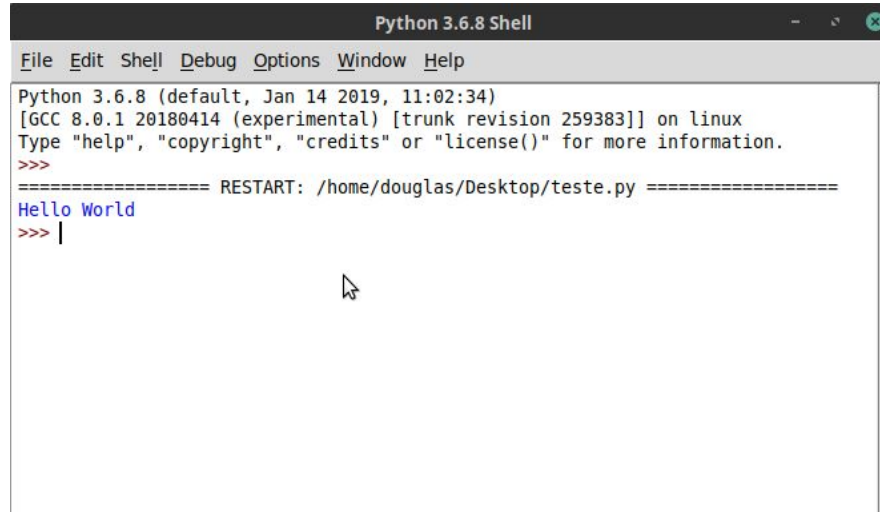
# Exemplo de código - Olá Mundo

```
print('Ola Mundo')
```

# Execução de códigos



# Resultado Esperado



```
Python 3.6.8 Shell
File Edit Shell Debug Options Window Help
Python 3.6.8 (default, Jan 14 2019, 11:02:34)
[GCC 8.0.1 20180414 (experimental) [trunk revision 259383]] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: /home/douglas/Desktop/teste.py =====
Hello World
>>> |
```

# Exemplos de código

# Exemplo de código

```
nome = input('Digite seu nome: ')\nif nome == 'Batata':\n    print('Ola Batata')\nelse:\n    print('Seu nome é', nome)
```



# Tipos de Dados

# Tipos básicos de dados

Tipo do dado	Comando	Criação
Texto	str	<code>var = 'Teste' ou var = "Teste"</code>
Inteiro	int	<code>var = 123</code>
Complexo	complex	<code>var = 1j</code>
Ponto flutuante	float	<code>var = 123. ou var = 123.0 ou var = .123</code>
Lista	list	<code>vet = [] ou vet = list()</code>
Conjunto	set	<code>vet = set()</code>
Dicionário	dict	<code>vet = {} ou vet = dict()</code>

Adaptado dos slides de Vinícius A. dos Santos

# Operações

# Operações aritméticas e lógicas

Aritméticas:

```
x = x + 1
```

```
x += 1
```

```
x = x ** 2  
x = x // 2
```

Lógicas:

```
'a' in ['a', 'b', 'c']
```

```
'a' in {'a': 23, 'b': 54}
```

```
a is b
```

```
a is not b
```

```
a == b
```

```
not a == b
```

```
a and b
```

```
a or b
```

Adaptado dos slides de Vinícius A. dos Santos

# Ferramentas de controle de fluxo

# Desvios condicionais

```
if x is True:  
    print("True")  
elif x is not False:  
    print("Still true")  
else:  
    print("False")
```

# Laços de repetição

```
x = 0
while x < 10:
    print(x)
    x += 1
```

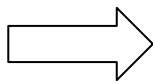
```
for i in range(10):
    print(i)
```

*# Percorrer vetores*

```
words = ['cat', 'window', 'defenestrate']
for w in words:
    print(w)
```

- Faça um contador de valores
  - O algoritmo deve percorrer toda a matriz
  - Deve ser contada a quantidade de vezes que cada número aparece na matriz
  - O resultado para cada número deve ser armazenado num vetor, na sua posição correspondente
  - O maior valor possível é 255

0	4	2
2	4	2
3	0	5



0	1	2	3	4	5
2	0	3	1	2	1

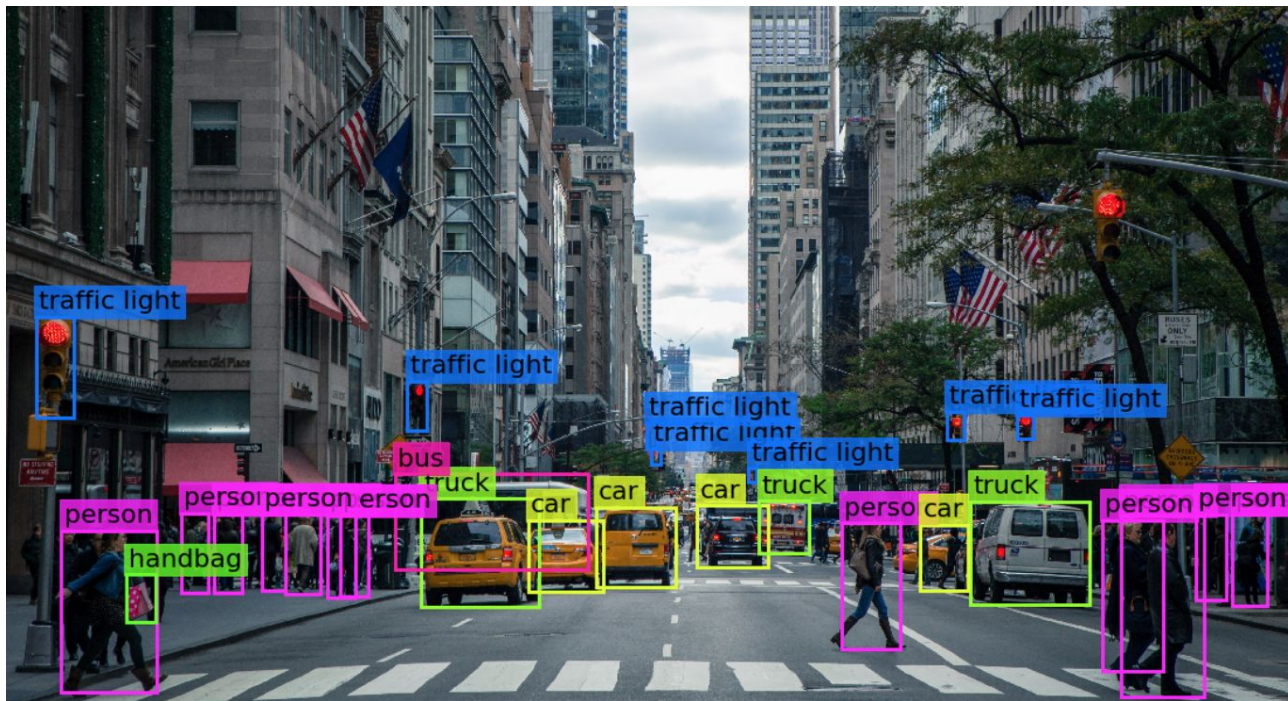


# OpenCV

# Introdução

- Iniciada em 1999 por Gary Bradsky, disponibilizado a partir de 2000
- Suporta várias linguagens: C++, Python, Java, etc
- Disponível para os sistemas: Windows, Linux, OS X, Android, iOS, etc
- OpenCV-Python é a versão da API do Python do OpenCV.
- OpenCV-Python é a ferramenta apropriada para prototipação rápida de problemas de visão computacional





# Instalação dos módulos

```
pip3 install opencv-python
```

# Carregando imagem

```
import cv2
```

```
# Ler imagem
```

```
# cv2.IMREAD_COLOR : Carrega imagem com cores
```

```
# cv2.IMREAD_GRAYSCALE : Carrega imagem em escala de cinza
```

```
img = cv2.imread("lena_color.png", cv2.IMREAD_COLOR)
```

```
# show image
```

```
cv2.imshow('lena', img)
```

```
# wait until the key 0 is pressed to destroy all windows
```

```
cv2.waitKey(0)
```

```
cv2.destroyAllWindows()
```

Imagem no repositório:  
[github.com/Douglasas/python-intro](https://github.com/Douglasas/python-intro)

# Detectando bordas

- Detector de bordas Canny

```
import cv2
```

```
img = cv2.imread("lena_color.png", cv2.IMREAD_COLOR)
```

```
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

```
canny = cv2.Canny(gray, 80, 200)
```

```
cv2.imshow('lena', canny)
```

```
cv2.waitKey(0)
```

```
cv2.destroyAllWindows()
```

```
cv2.imwrite("lena_result.png", img)
```

# Funções do OpenCV

```
canny = cv2.Canny(imagem, 80, 200)
```

```
stylization = cv2.stylization(img)
```

```
blur = cv2.blur(imagem, (5, 5))
```

```
enhance = cv2.detailEnhance(img)
```

```
filtered = cv2.edgePreservingFilter(img)
```

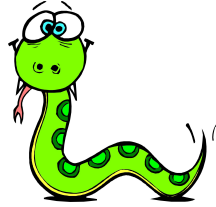
# Canny aplicado em tempo real

```
import cv2
# capture video
# from camera device (device index)
# from file (name of video file)
cap = cv2.VideoCapture(0)
while True:
    # Capture frame-by-frame
    ret, frame = cap.read()
    # Our operations on the frame come here
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    canny = cv2.Canny(gray, 100, 200)
    # Display the resulting frame
    cv2.imshow('frame', canny)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
# When everything done, release the capture
cap.release()
cv2.destroyAllWindows()
```



# Conclusão

- Python é legal



- É possível trabalhar com processamento de imagens com Python
- Adicione a skill ao seu LinkedIn

# Referências

## Wiki do Python

- <https://wiki.python.org/>

## Wiki do NumPy

- <https://docs.scipy.org/doc/numpy/dev/>

## OpenCV-Python Tutorials

- <https://opencv-python-tutroals.readthedocs.io>

## A byte of Python

- <https://python.swaroopch.com/>

Universidade do Vale do Itajaí  
Escola do Mar, Ciência e Tecnologia  
Engenharia de Computação

# Introdução ao Python

**Douglas A. dos Santos - [douglasas@edu.univali.br](mailto:douglasas@edu.univali.br)**  
LEDS - Laboratory of Embedded and Distributed Systems