POLITÉCNICO DE LEIRIA | ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

MEE / MEEE

Visão Computacional / Computer Vision

Introduction to Python 3.x and OpenCV 4

1. Introduction

Python is a high-level, interpreted language (no prior compilation required before executing), interactive (can interact directly with the interpreter), object-oriented, and portable (can be executed on many hardware platforms). The final version of Python 2 is version 2.7, (released in 2010) and the latest version is Python 3 (released in 2008). Although there is version 3 of this language, Python 2.x is still widely used due to its development and information over time. There are some minor differences in programming syntax between versions 2 and 3.

For Python tests with standard libraries, if the user chooses to, online editors are available, for Python 2.7 or Python 3. We will use **Python 3** in this course.

2. Standalone installation (Python, OpenCV and libraries)

For Linux users, Python is native with the operating system, no installations are required, except eventual additional libraries to use.

For Windows or Mac OS X users, Python requires installation.

For Mac users, the installation is similar to the previously presented, so it is suggested also to check this online tutorial: Install OpenCV 4 on MacOS by Fahmi.

For Windows users it is suggested as follows:

IMPORTANT: The figures below present as example the installation steps for Python 3.7 and OpenCV 4. Both packages have newer versions, although the steps for installation are similar, so the figures may not be exactly equal to the ones presented, but are certainly similar.

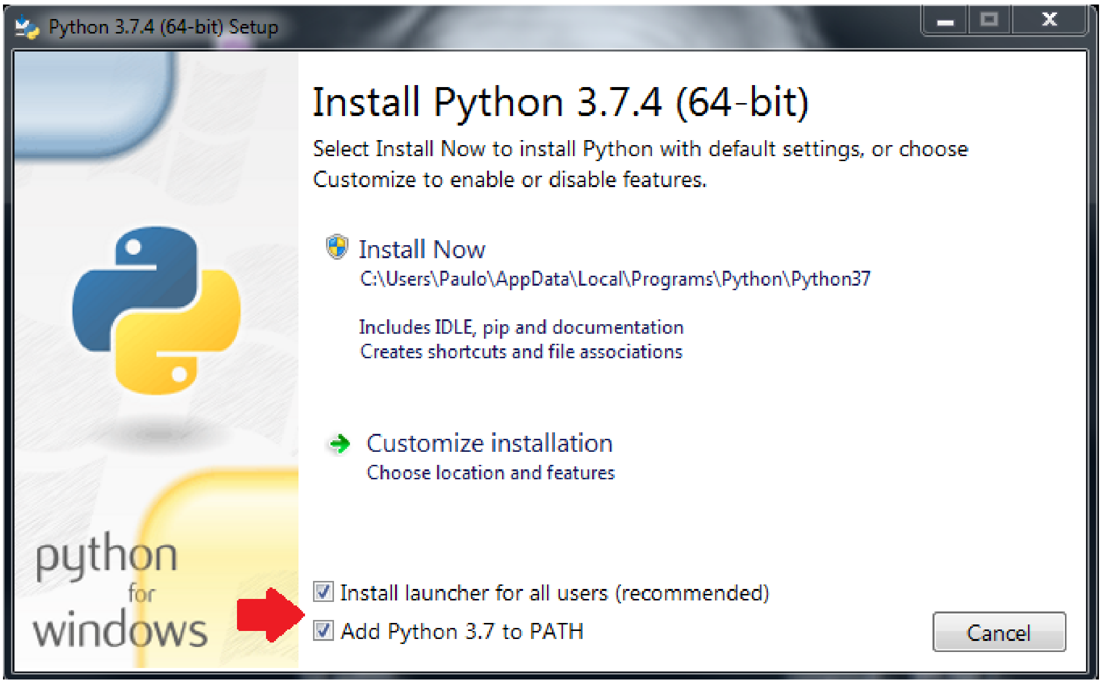
2.1 Python

- Download the installer from the last stable version of Python 3.x: https://www.python.org/downloads/, having in consideration if your machine is running on a 32 or 64 bits operating system.

- Start the installer and check "Add Python 3.7 to PATH" as presented in Figure 1. Afterwards, keep all the options checked in the Customize installation screen (Figure 2) and finally in Advanced Options screen check "*Install for all users*", "*Add*

Python to environment variables" and "*Precompile standard library*". Optionally, you can customize the install location to C:\Python37, similarly to Figure 3, and press Install.

- Depending on the speed of your computer, it will take a few minutes to install and when the last page of the installer appears, you should also press the "*Disable path length limit"*, as presented in Figure 4.



Python 3.7.4 (64-bit) Setup

Install Python 3.7.4 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

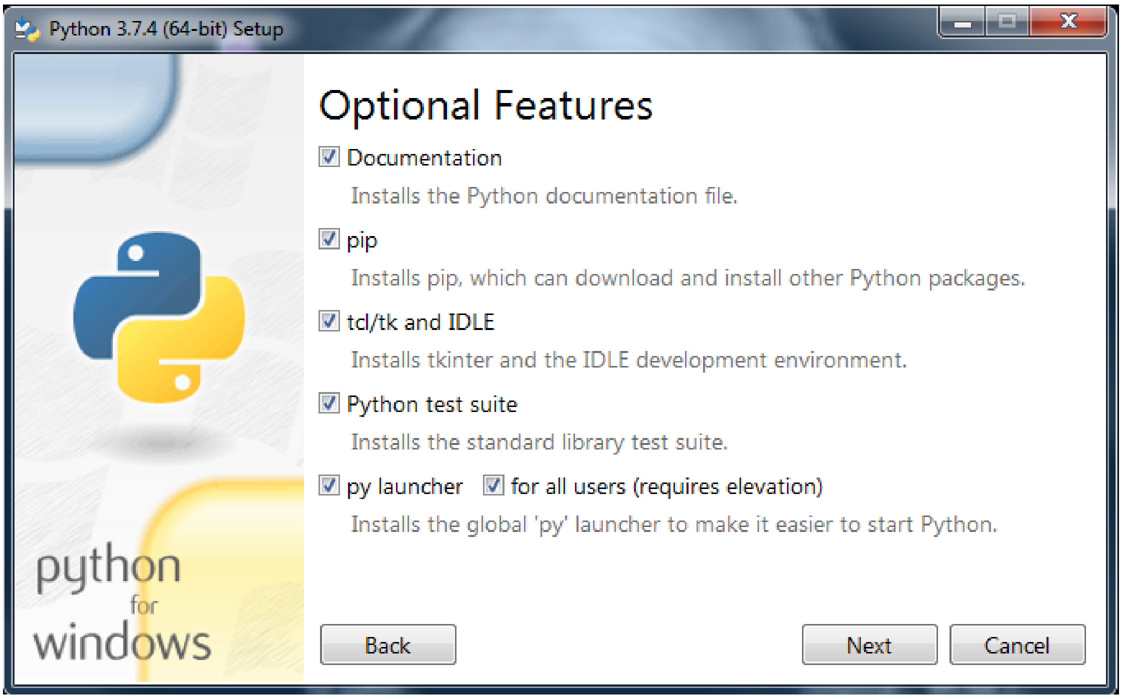
Install Now
C:\Users\Paulo\AppData\Local\Programs\Python\Python37

Includes IDLE, pip and documentation
Creates shortcuts and file associations

Customize installation
Choose location and features

☑ Install launcher for all users (recommended)
☑ Add Python 3.7 to PATH

Figure 1: Python installer initial screen.



Python 3.7.4 (64-bit) Setup

Optional Features

☑ Documentation
Installs the Python documentation file.

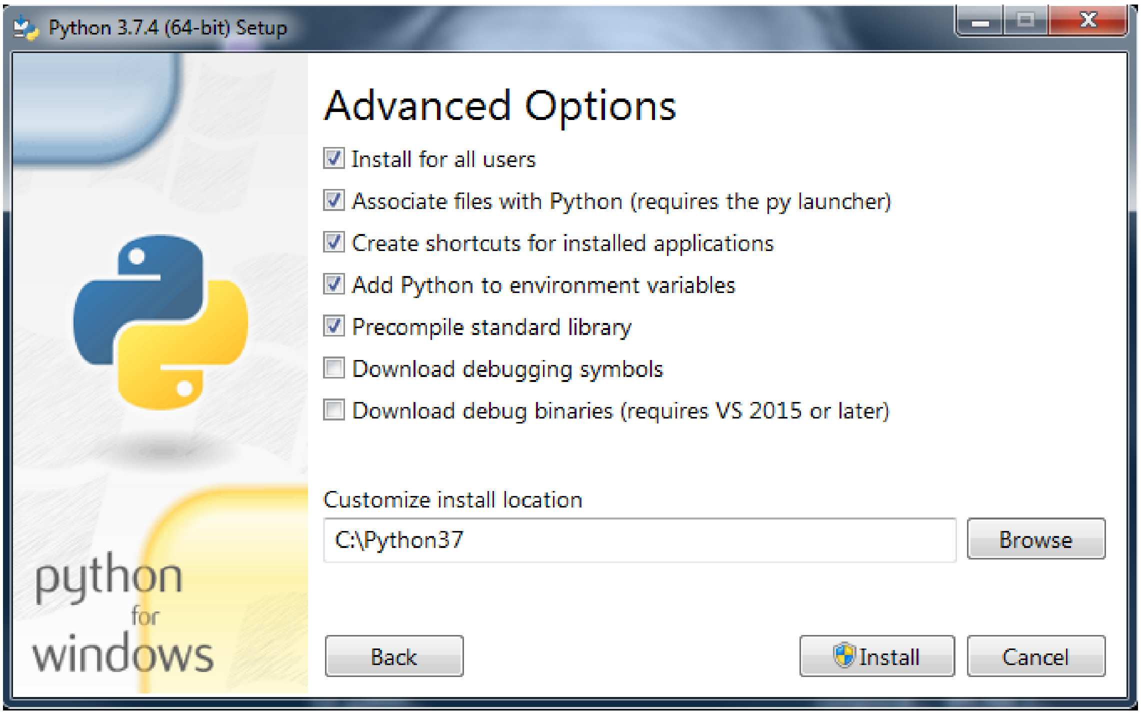
☑ pip
Installs pip, which can download and install other Python packages.

☑ tcl/tk and IDLE
Installs tkinter and the IDLE development environment.

☑ Python test suite
Installs the standard library test suite.

☑ py launcher ☑ for all users (requires elevation)
Installs the global 'py' launcher to make it easier to start Python.

Figure 2: Customize installation screen.



Advanced Options

☑ Install for all users

☑ Associate files with Python (requires the py launcher)

☑ Create shortcuts for installed applications

☑ Add Python to environment variables

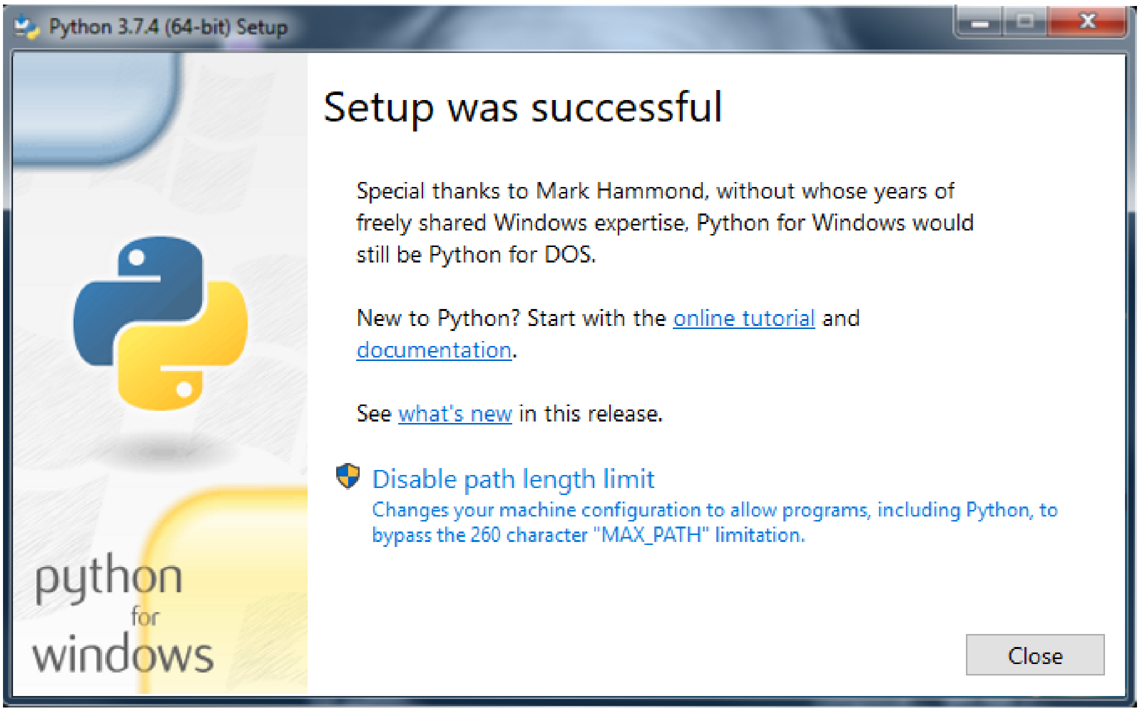
☑ Precompile standard library

□ Download debugging symbols

□ Download debug binaries (requires VS 2015 or later)

Customize install location: C:\Python37

Figure 3: Advanced Options screen.



Setup was successful

Special thanks to Mark Hammond, without whose years of freely shared Windows expertise, Python for Windows would still be Python for DOS.

New to Python? Start with the online tutorial and documentation.

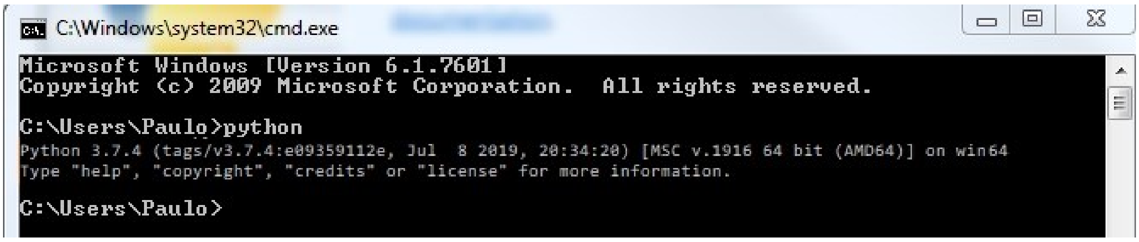
See what's new in this release.

Disable path length limit

Changes your machine configuration to allow programs, including Python, to bypass the 260 character "MAX_PATH" limitation.

Figure 4: Setup final screen.

- To check if Python was correctly installed, open a Command Prompt window (type “cmd” in search box), type “*python*” command and press enter – Figure 5.



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Paulo>python
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win64
Type "help", "copyright", "credits" or "license" for more information.

C:\Users\Paulo>
```

Figure 5: Verification of installation.

- Exit from the Python interpreter by writing “*quit()*” and pressing the enter key.

2.2 Additional libraries

In order to add libraries useful to the course, we can install them through Command window. We will install **NumPy** (**Num**erical **Py**thon: package that provides a high-performance multidimensional array object, and tools for working with these arrays), **SciPy** (**Sci**entific **Py**thon: package that uses NumPy arrays as basic structure and adds modules useful to scientific programming - linear algebra, calculus, differential equation solving, and signal processing) and **Matplotlib** (library used to create 2D graphs and plots by using python scripts):

```
python -m pip install numpy --user
python -m pip install scipy --user
python -m pip install matplotlib --user
```

After the execution of each command you should see a “Successfully installed” message.

2.3 OpenCV

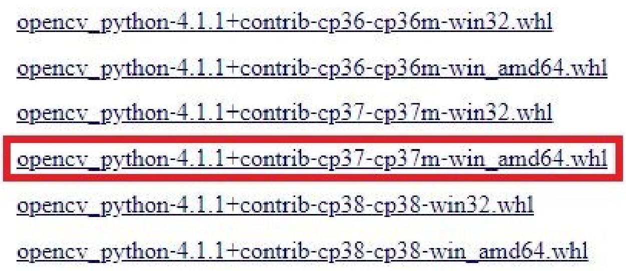
OpenCV is an open-source computer vision and machine learning software library. To install this library in your computer, the following steps are advisable:

```
python -m pip install opencv-contrib-python --user
```

The installation is complete. You can test it by running Python and importing the OpenCV library, using the command “import cv2”.

As alternative method (only use if you didn’t manage with the previous command), you can also use:

- Download the OpenCV binary files from here. Take in account that you must select a version compatible with the Python version installed in your system. So, what is the correct version?



opencv_python-4.1.1+contrib-cp36-cp36m-win32.whl

opencv_python-4.1.1+contrib-cp36-cp36m-win_amd64.whl

opencv_python-4.1.1+contrib-cp37-cp37m-win32.whl

opencv_python-4.1.1+contrib-cp37-cp37m-win_amd64.whl

opencv_python-4.1.1+contrib-cp38-cp38-win32.whl

opencv_python-4.1.1+contrib-cp38-cp38-win_amd64.whl

Figure 6: OpenCV version selection.

For the latest version of OpenCV (say version 4.1) and for Python 3.7.x installed in your system. An example is presented using Figure 6:

cp36m-win32.whl means that OpenCV is for python 3.6 (32-bit version)

cp37m-win_amd64.whl means that OpenCV is for python 3.7 (64-bit version)

Just need to choose the correct one that suits your Python installation (say its contrib-cp37m-win_amd64.whl).

The use of a version with the "*contrib*" attributes in the name of the installer, will be useful in later stages of this curricular unit, so it is advised to install this version.

- Run the Command Prompt window (type "cmd" in search box), **go the directory where the OpenCV binaries are located**, for example c:\Users\Paulo

- Install using the command:

```
python -m pip install opencv_python-4.1.1+contrib-cp37-cp37m-win_amd64.whl
```

- The installation is complete. You can test it by running Python and importing the OpenCV library, using the command "import cv2".

3. Python from distributions

Python is also available in platforms (such as Anaconda, etc.) that you can use. If you are already using Anaconda for example, you can use it in the course. If you wish to upgrade Python and all the other libraries to the latest version the following commands are advised, running the **"*Anaconda prompt*" with administrative privileges** (or similar command line for other distributions).

```
# (optional) Update conda and/or pip installers
conda update conda
pip install --user --upgrade pip

# (optional) Update anaconda
conda update anaconda
```

```
# update Python to the latest version (say version 3.7.4)
conda install python=3.7.4

# update scientific libraries
pip install -U numpy --user
pip install -U scipy --user
pip install -U matplotlib --user
```

To perform OpenCV installation, proceed accordingly to section 2.3.

4. Examples

Some examples using the previously mentioned libraries are presented as follows:

4.1 Draw a curve

```
# import libraries
import numpy as np
# import scipy as sp
import matplotlib.pyplot as plt


# define 100 points in [0,1] interval
t = np.linspace(0, 1, 100)
# t = sp.linspace(0, 1, 100)

# plot a parabola curve in the defined interval
plt.plot(t, t**2)
plt.show()
```

The code will result in a similar curve to the one shown in Figure 7.

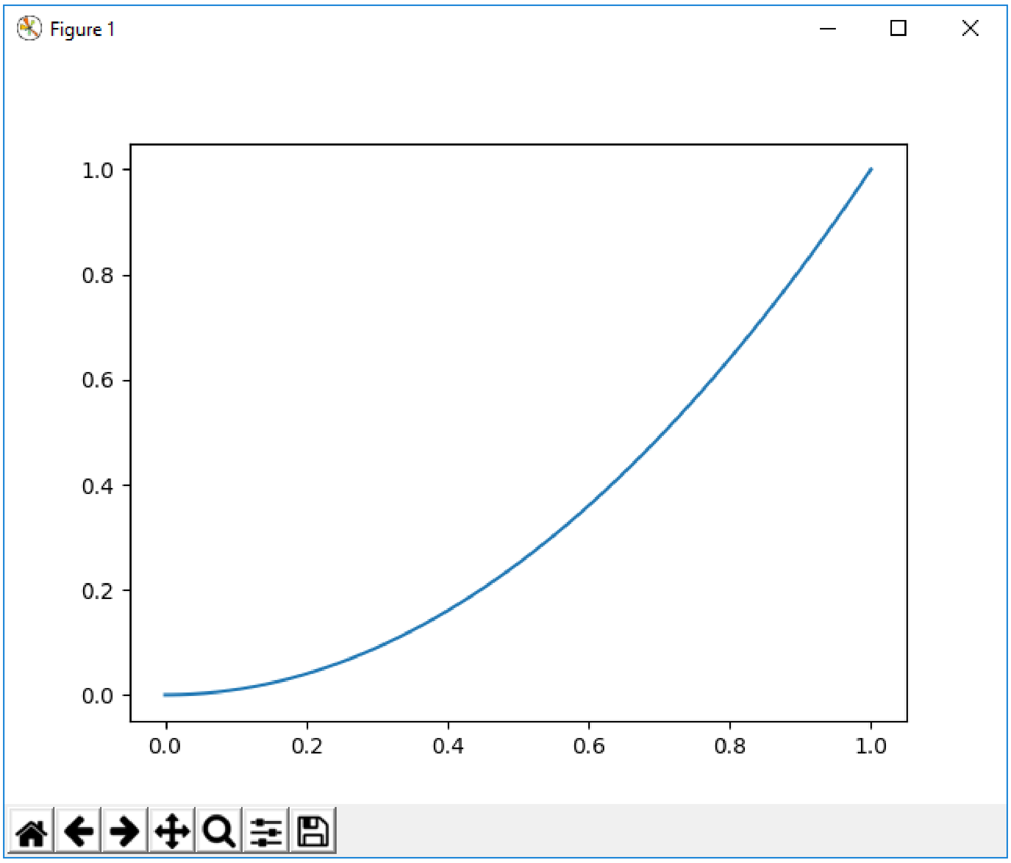


Figure 7: Parabola curve in (0,1) interval.

4.2 Open and display an image with OpenCV library

```python
# importing OpenCV(cv2) module
import cv2

# Read image
img = cv2.imread('python_opencv.png')

# Output img with window name as 'image'
cv2.imshow('Image loaded with OpenCV', img)

# Maintain output window until user presses a key
cv2.waitKey(0)

# present windows on screen
cv2.destroyAllWindows()
```

The code will display the picture depicted in Figure 8.

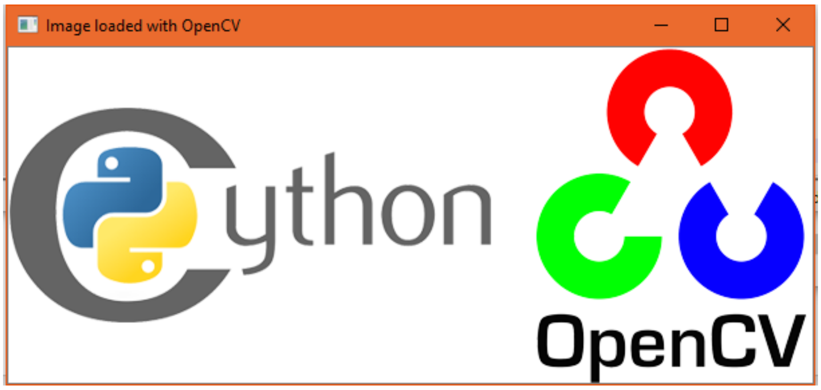


Figure 8: Example of displaying a picture.

4.3 Open and display an image with matplotlib library

```
# importing matplotlib modules
import matplotlib.image as mpimg
import matplotlib.pyplot as plt

# Read image
img = mpimg.imread('python_opencv.png')

# Output Images
plt.imshow(img)
# Maintains the output window visible
plt.show()
```

The code will also display a similar picture to the depicted in Figure 8.

4.4 Open, modify and save the result in a new image

```
# importing OpenCV(cv2) module
import cv2

# Read image
image = cv2.imread('python_opencv.png')
# Converts to grayscale
image_gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# Saves the resulting image
cv2.imwrite('python_opencv_gray.png', image_gray)
# Outputs the images
cv2.imshow('Original image', image)
cv2.imshow('Gray image', image_gray)

# Maintain output window until user presses a key
cv2.waitKey(0)
# present windows on screen
cv2.destroyAllWindows()
```

The new grayscale image will be displayed as presented in Figure 9.

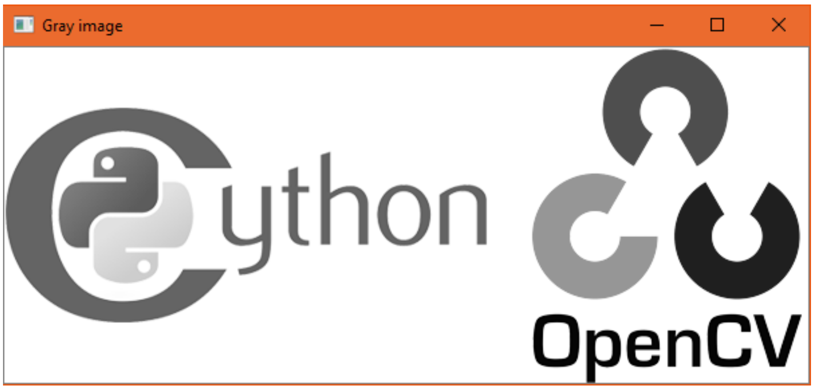


Figure 9: Grayscale image.