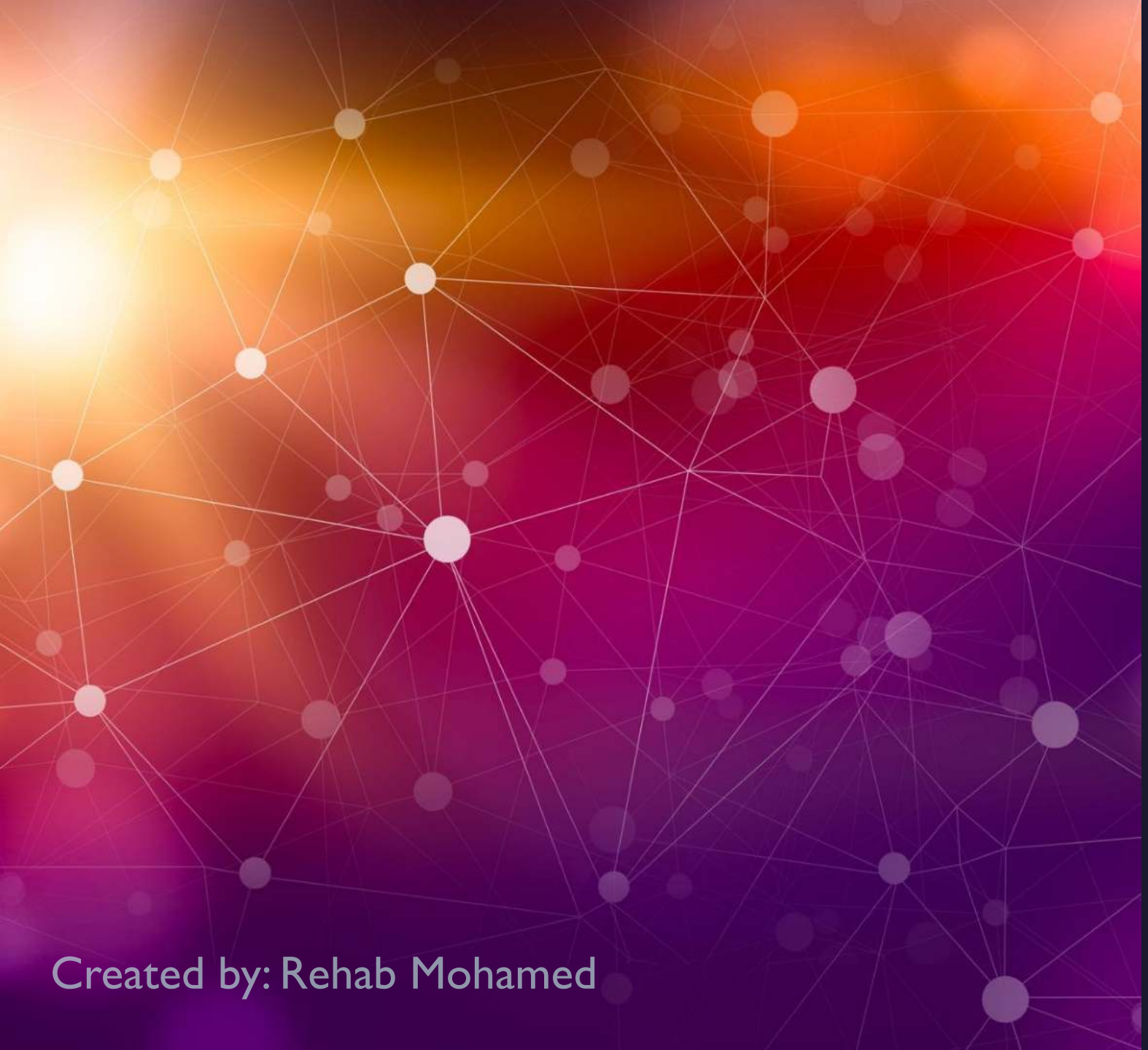


Image Processing in Python

Section 6

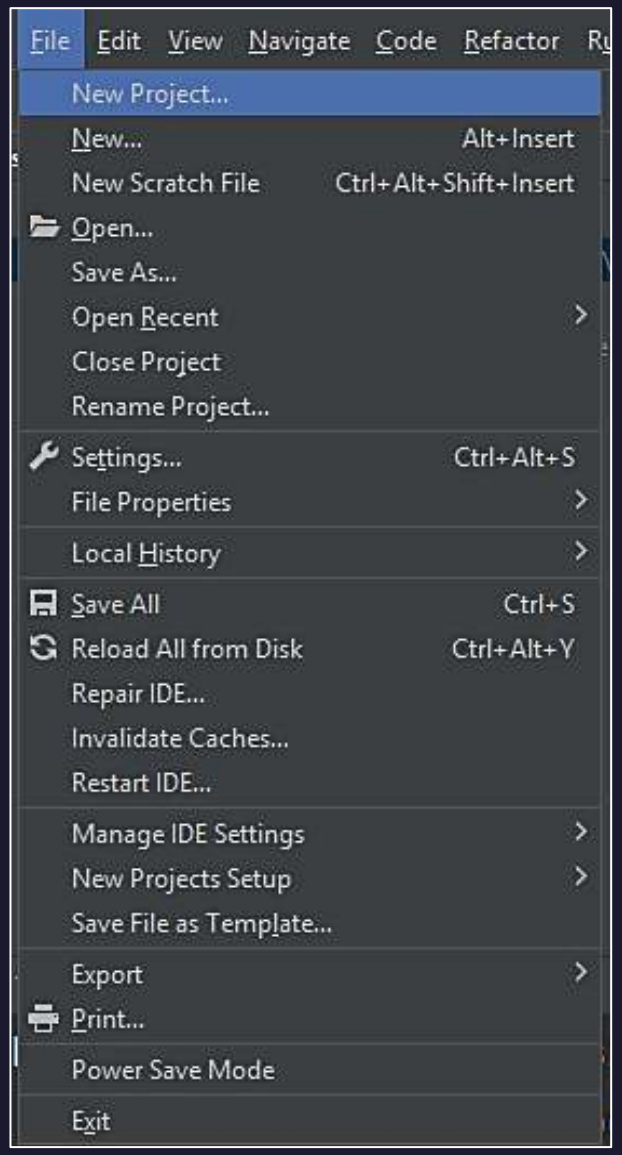
Created by: Rehab Mohamed

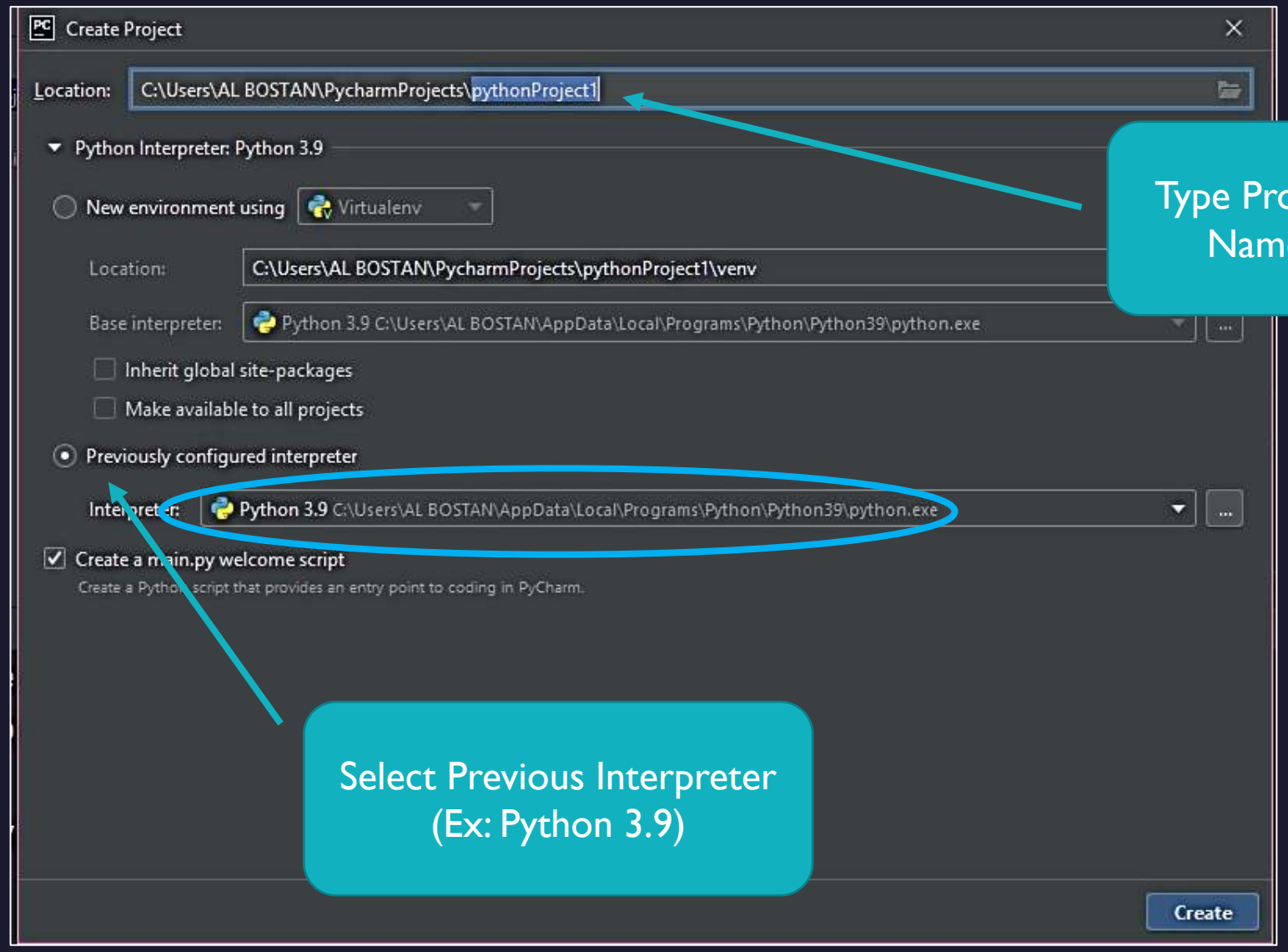
Python provides lots of libraries for image processing, including:

- **Python Imaging Library (PIL)** − To perform basic operations on images like create thumbnails, resize, rotation, convert between different file formats etc.
- **Install required library**

Our first step will be to install the required library, like pillow or other which we wants to use for image processing.

Creating New Project in Pycharm

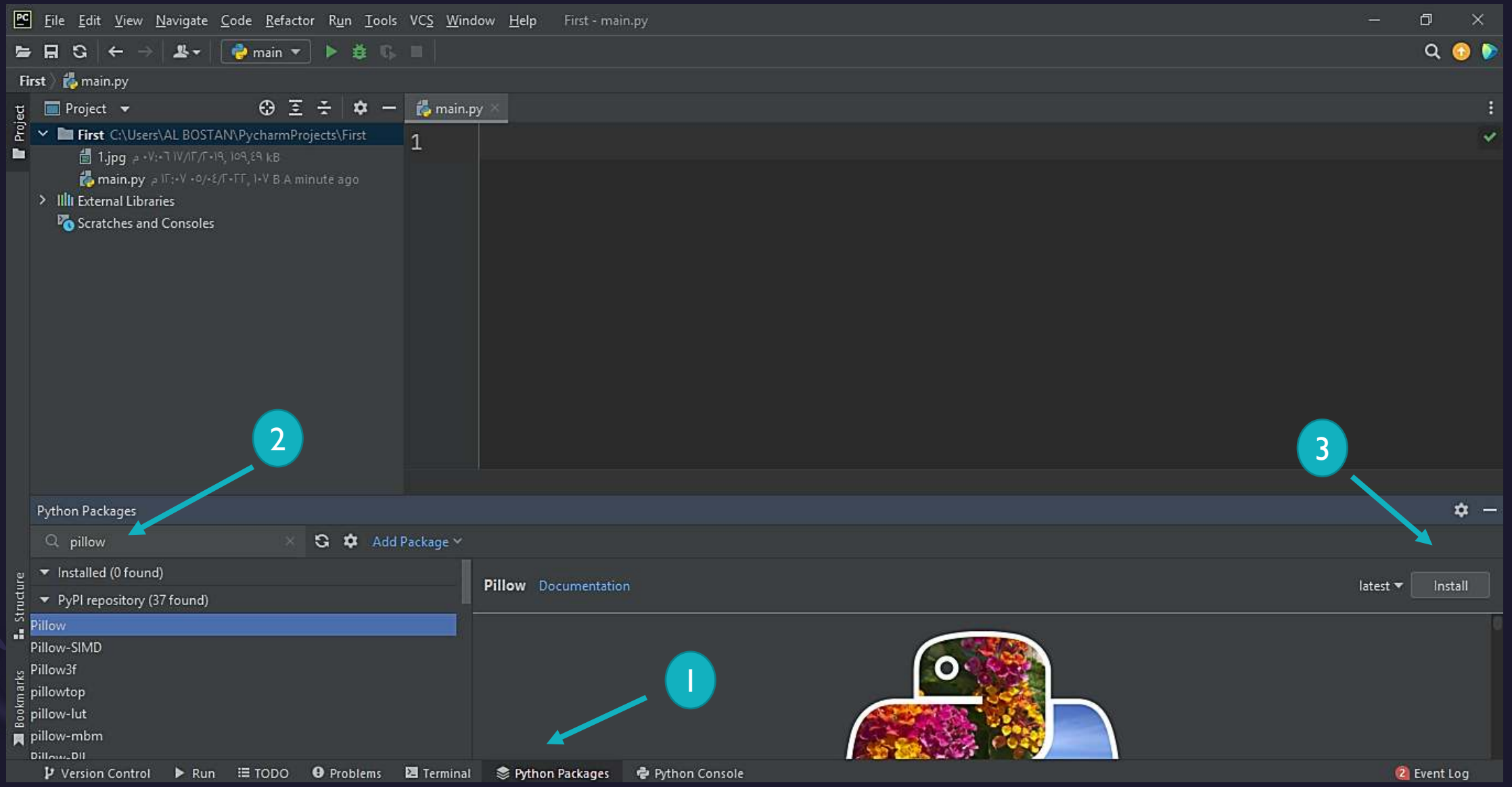




Type Project Name

Select Previous Interpreter (Ex: Python 3.9)

Install Pillow



Install Pillow in vs

Install Pillow in vs

```
python -m pip install pillow
```

Image: Open() and show()

```
#Import required library

from PIL import Image

#Open Image

im = Image.open(“1.jpg”)

#Image rotate & show

im.rotate(45).show()
```

Output

Image.size: It returns the tuple consist of height & weight of the image.
Image.format: It returns file format of the image file like ‘JPEG’, ‘BMP’, ‘PNG’, etc.

```
print(im.size)

print(im.format)
```

Output:

```
(1600, 1066)
JPEG
```

Image.width: It returns only the width of the image.

```
print(im.width)

    1600
```

Image.height: It returns only the height of the image.

```
print(im.height)

    1066
```

Image.info: It returns a dictionary holding data associated with the image

```
print(im.info)

{'jfif': 257, 'jfif_version': (1, 1), 'dpi': (96, 96),
'jfif_unit': 1, 'jfif_density': (96, 96)}
```

Image.filename: This function is used to get the file name or the path of the image.

```
print(im.filename)

    1.jpg
```

Image.mode: It is used to get the pixel format used by the image. Typical values are “1”, “L”, “RGB” or “CMYK”..

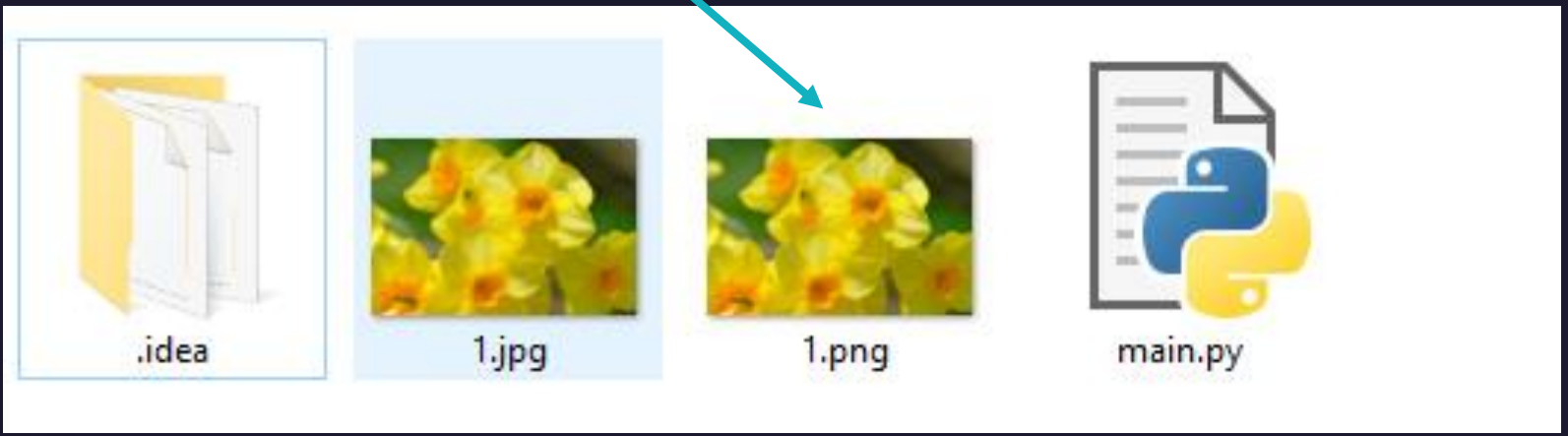
```
print(im.mode)

    RGB
```

Convert and Save() Image

We can change the format of image from one form to another:

```
im.save("1.png")
```



Resize the image

The resize() function doesn't modify the used image. Instead, it returns another Image with the new dimensions. The Image.resize() method returns a resized copy of the source image.

```
resized_img = im.resize((300,300))

print(resized_img.size)  #(300,300)

resized_img.show()
```

Resize-thumbnails()

We can change the size of image using thumbnail() method of pillow –

```
im.thumbnail((300,300))

im.save('image_thumbnail.jpg')

im2 = Image.open('image_thumbnail.jpg')

im2.show()

print(im.size)    #(300,200)
```

Difference between resize() vs. thumbnail()

- The resize() method returns the image whose width and height exactly match the passed in value.

This could be what you want, but at times you may find that the images returned by the resize() function aren't ideal. This is mostly because the method doesn't account for the image's Aspect Ratio, so you might end up with the image that either looks stretched or squished.

- If you want to resize the image and keep their aspect ratios as it is, then you should use a thumbnail() function to resize them

Converting to grayscale image – convert()

We can make the grayscale image from our original colored image.

```
im_gray = im.convert('L')

im_gray.show()
```

Where "L" stands for 'luminous'.

Merging two images

In the same way, to merge two different images, you need to:

- Create image object for the required images using the open() function.
- While merging two images, you need to make sure that both images are of same size. Therefore, get each sizes of both images and if required, resize them accordingly.
- Create an empty image using the Image.new() function.
- Paste the images using the paste() function.
- Save and display the resultant image using the save() and show() functions

```
from PIL import Image

image1 = Image.open("elephant.png")

image1.show()

print("Size of image 1: ", image1.size)

image2 = Image.open("mountain.png")

image2.show()

print("Size of image 2: ", image2.size)

# Resize first image

image1 = image1.resize((image2.width, image2.height))

image1_size = image1.size

image2_size = image2.size
```

```
new_image =
Image.new('RGB',(2*image1_size[0],image1_size[1]),(250,250,250))

new_image.paste(image1,(0,0))

new_image.paste(image2,(image1_size[0],0))

new_image.save("Merged_image.PNG")

new_image.show()
```

Input image 1:

Input image 2:

Merged Image

Task

Write python program that convert RGB image to Binary and save it as ‘png’.

Thank You