**IS201 Fundamentals of Computing**

**HOP09 Manipulating Images and GUI Automation**

04/16/2020 Developed by Amrutha Vaidyanathan and Apiwat Chuaphan

05/28/2020 Reviewed by Kim Nguyen

Center for Information Assurance (CIAE) @City University of Seattle (CityU)

**Before You Start**

* The directory path shown in screenshots may be different from yours.
* Some steps are not explained in the tutorial**.** If you are not sure what to do:
  1. Consult the resources listed below.
  2. If you cannot solve the problem after a few tries, ask a TA for help.

**Learning Outcomes**

Students will be able to:

* Manipulate images with Python’s Pillow module
* Control your mouse and keyboard on computer

**Resources**

* [Automate the Boring Stuff with Python](https://login.proxy.cityu.edu/sso/skillport?context=89288)

**Preparation**

1. In Visual Studio Code, open the private repository generated when you accepted the HOP09 assignment (If you cannot find that repository in your machine, you might have not cloned the repo, if so, please do before proceeding).

A screenshot of a cell phone

Description automatically generated

**Manipulating Images**

Python’s **Pillow** module allows you to interact with image files such as cropping, resizing, and editing the content of the image.

First, install pillow module by typing this command. If using mac, you can do this in mac **terminal** or **cmd** for windows. You can do this in the VS code terminal too.

First, install pillow module by typing this command

>>> **pip3 install Pillow**

*Before using the module, this is the basic understanding of computer image. An RGBA value represents color in image (red, green, blue, alpha(transparency)). Each value is between 0-255, for example (255, 255, 255, 255) means maximum of white and fully opaque. In pillow, it makes thing easier for you, no need to memorize RGBA value. So, (255, 255, 255, 255) equals to ImageColor.getcolor(‘white’, ‘RGBA’)*

1. Let’s try to get some color on interactive shell with the following or try other colors.

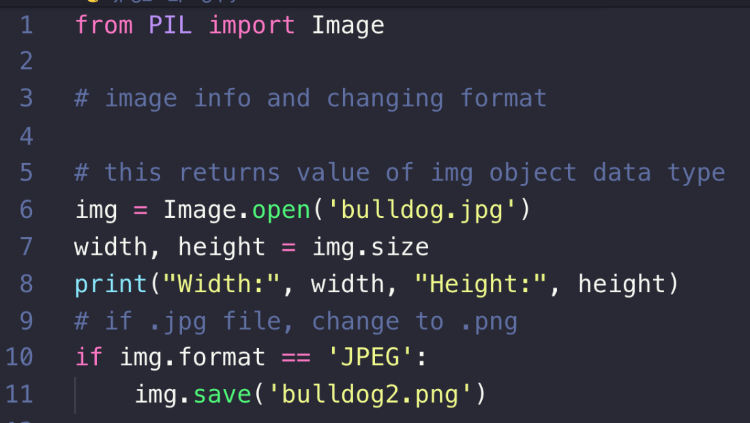
>>> Type **Python3** in terminal and then...

A screenshot of a cell phone

Description automatically generated

>>> Type **exit()** to exit from the above view

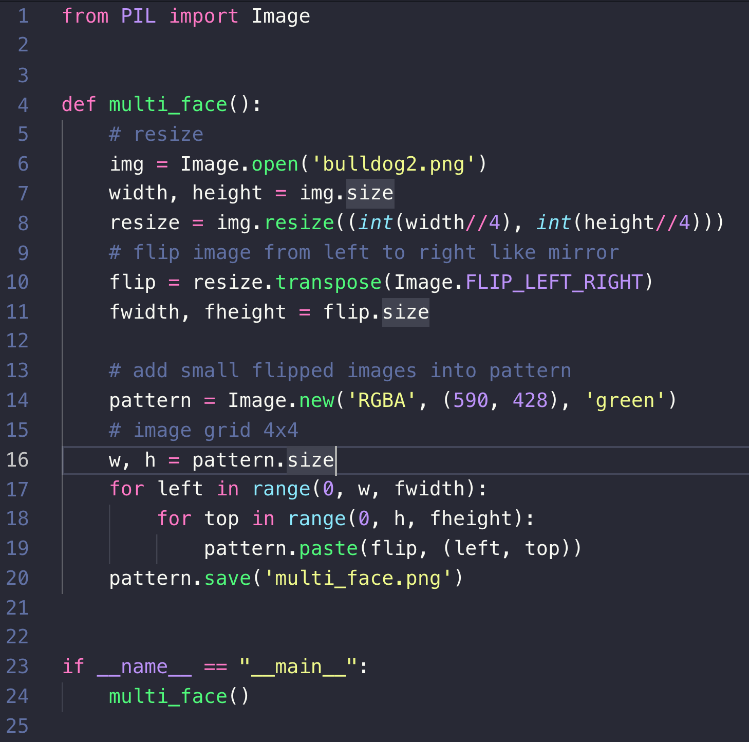
1. Download image [here](https://www.petmd.com/sites/default/files/Acute-Dog-Diarrhea-47066074.jpg) and we will use it for this module. Make sure it is in the current directory and name it as **bulldog.jpg**. Create a file **jpg\_to\_png.py** and type the following.



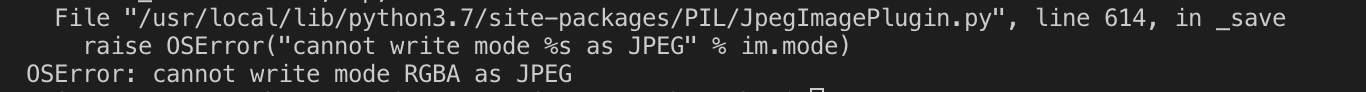
>>> In the VS code terminal type **python3 jpg\_to\_png.py**

The output will be displayed as W: 590 H:428. And you will see new image file name bulldog2.png created under Module9.

1. Let’s edit the image, cropping, resizing, and pasting into another image. Create **multi\_face.py**with the following code. We will use the **bulldog2.png** created from the above code.

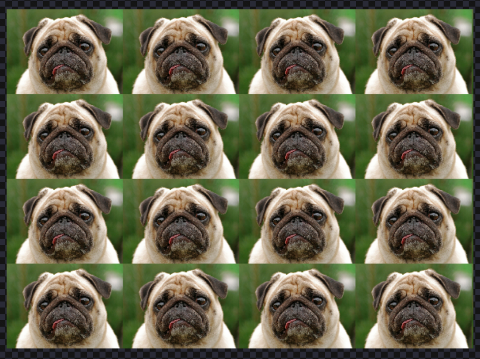


NOTE: As in the above code we are using RGBA we cannot write the image to jpg. If we do, we will get error like this.



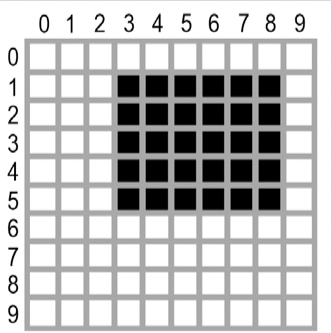
>>> In the terminal type **python3 multi\_face.py**

A new image file will be created **multi\_face.png**and this is the result.

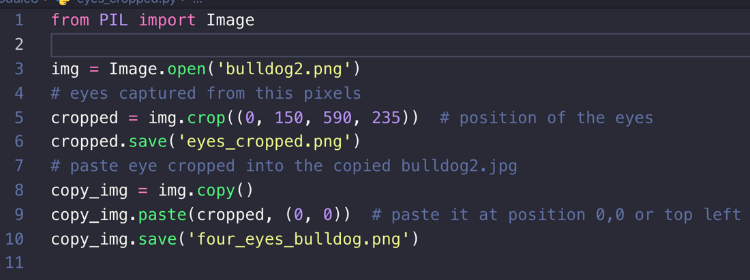


1. Create **eyes\_cropped.py** and type the following. We will use the **bulldog2.png**.

This is how we find a particular area.



The area represented by the box tuple (3, 1, 9, 6)

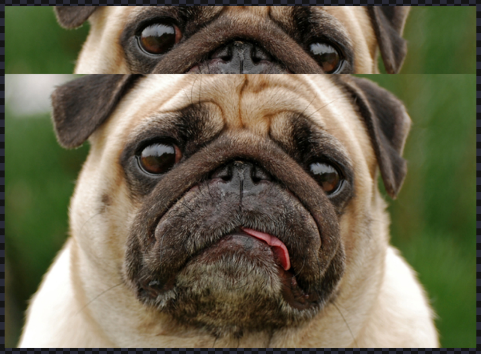


>>> In the terminal type **python3 eyes\_cropped.py**

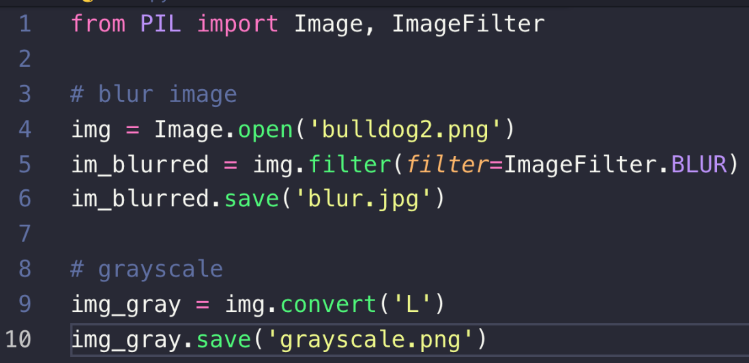
This is the how **eyes\_cropped.png** and **four\_eyes\_bulldog.png** look like.



and



1. This Python’s module can also filter your image. Create **filter.py** with the following.



>>> In the terminal type **python3 filter.py**

Two new files name **blur.jpg**and **grayscale.png**were created.

**blur.jpg**

A close up of a dog looking at the camera

Description automatically generated

**grayscale.png**

A close up of a dog looking at the camera

Description automatically generated

**GUI Automation**

Python provides some modules that can do anything like you sit in front of computer. Pyautogui has functions to simulating mouse and keyboards.

To install:

>>> **pip3 install pyautogui**

MacOS may need to install other dependencies.

>>> **pyau**

**>>> pip3 install pyobjc-core**

**>>> pip3 install pyobjc**

0,0 X increases -->

+---------------------------+

| | Y increases

| | |

| 1920 x 1080 screen | |

| | V

| |

| |

+---------------------------+ 1919, 1079

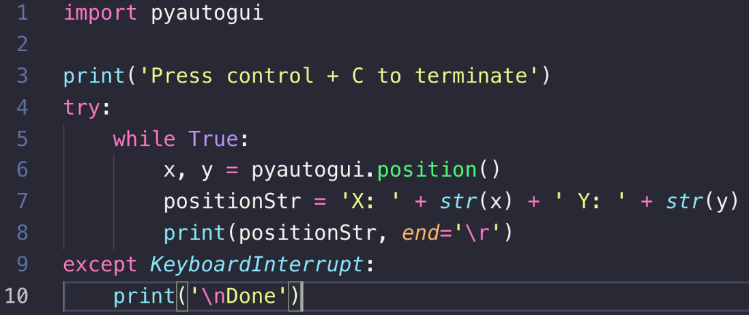
1. First, find your screen’s width and height in pixels by entering this in the terminal

A close up of a logo

Description automatically generated

>>> Type **exit()** to exit from the above view

1. What if you can get your position in real time. Create **find\_position.py**with the following code to do that.



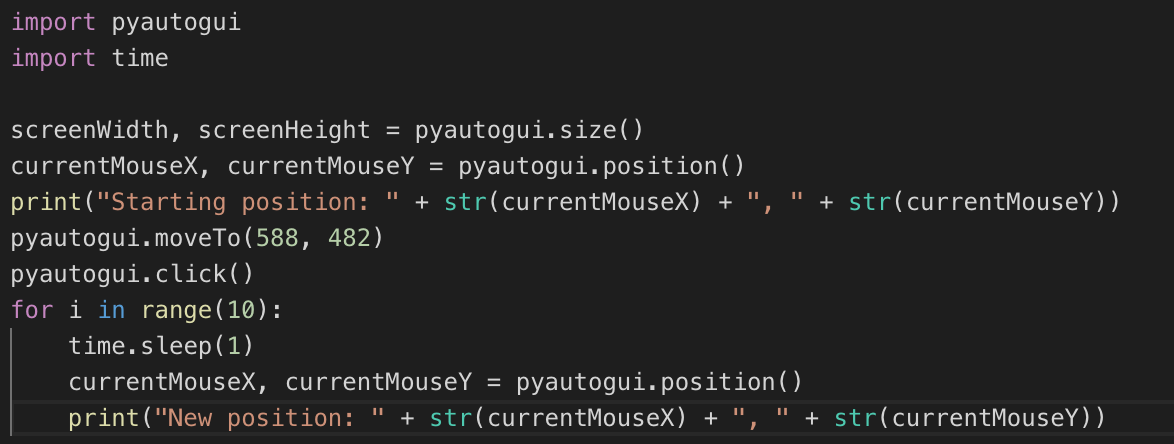
>>> In the terminal type **python3 find\_position.py**

A picture containing drawing

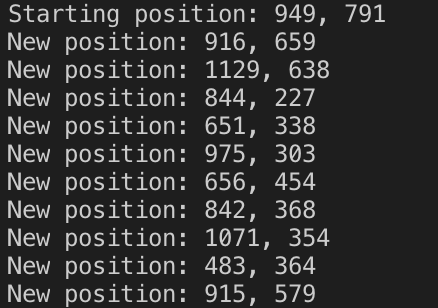
Description automatically generated

Each time you move your cursor, the X and Y value changes. Press control + c to terminate. Done message will be printed for terminating.

1. Let’s create a program called **find\_position1.py**and type the following.



>>> In the terminal type **python3** **find\_position1.py**. Move the cursor to print the new position every time.



**Push your work to GitHub**

Run the following commands to push your work to the GitHub repository:

Open the terminal from the VSCode by hitting the “control” + “~” key and type the following command:

>>> git add .

>>> git commit -m “Submission for Module 9”

>>> git push origin master