



이수안 컴퓨터 연구소

suan computer laboratory

# 파이썬으로 포토샵 만들기



이수안

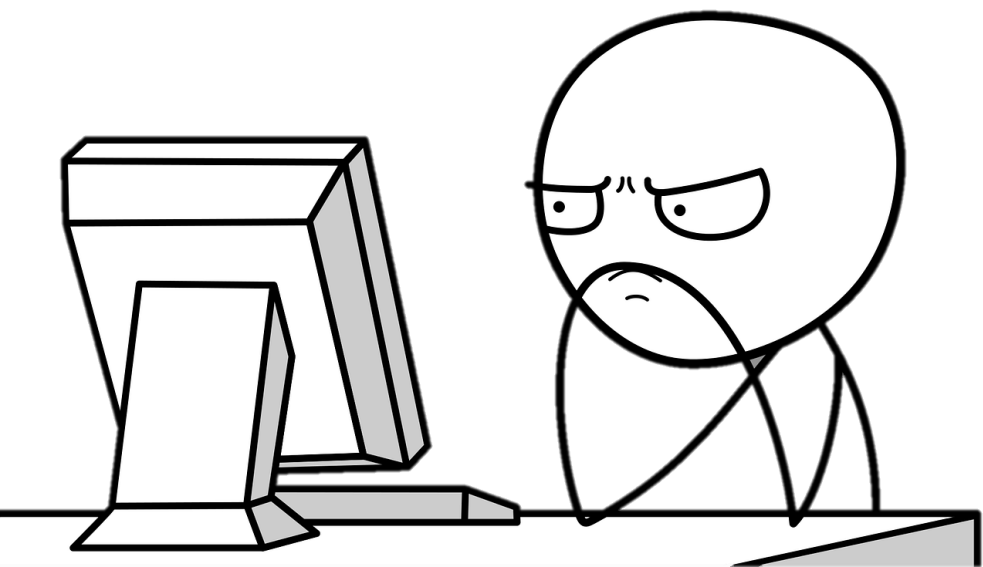


# 목차

---

1. 포토샵이 뭐지?
2. 이미지 매직 설치
3. 파이썬 포토샵 만들기
  1. GUI 구성
  2. 메뉴 구현과 함수 선언
  3. 사진 열기/저장
  4. 사진 확대/축소
  5. 사진 상하/좌우 반전
  6. 사진 회전
  7. 사진 명도/채도/색조
  8. 사진 흑백 변환

# 1. 포토샵이 뭐지?

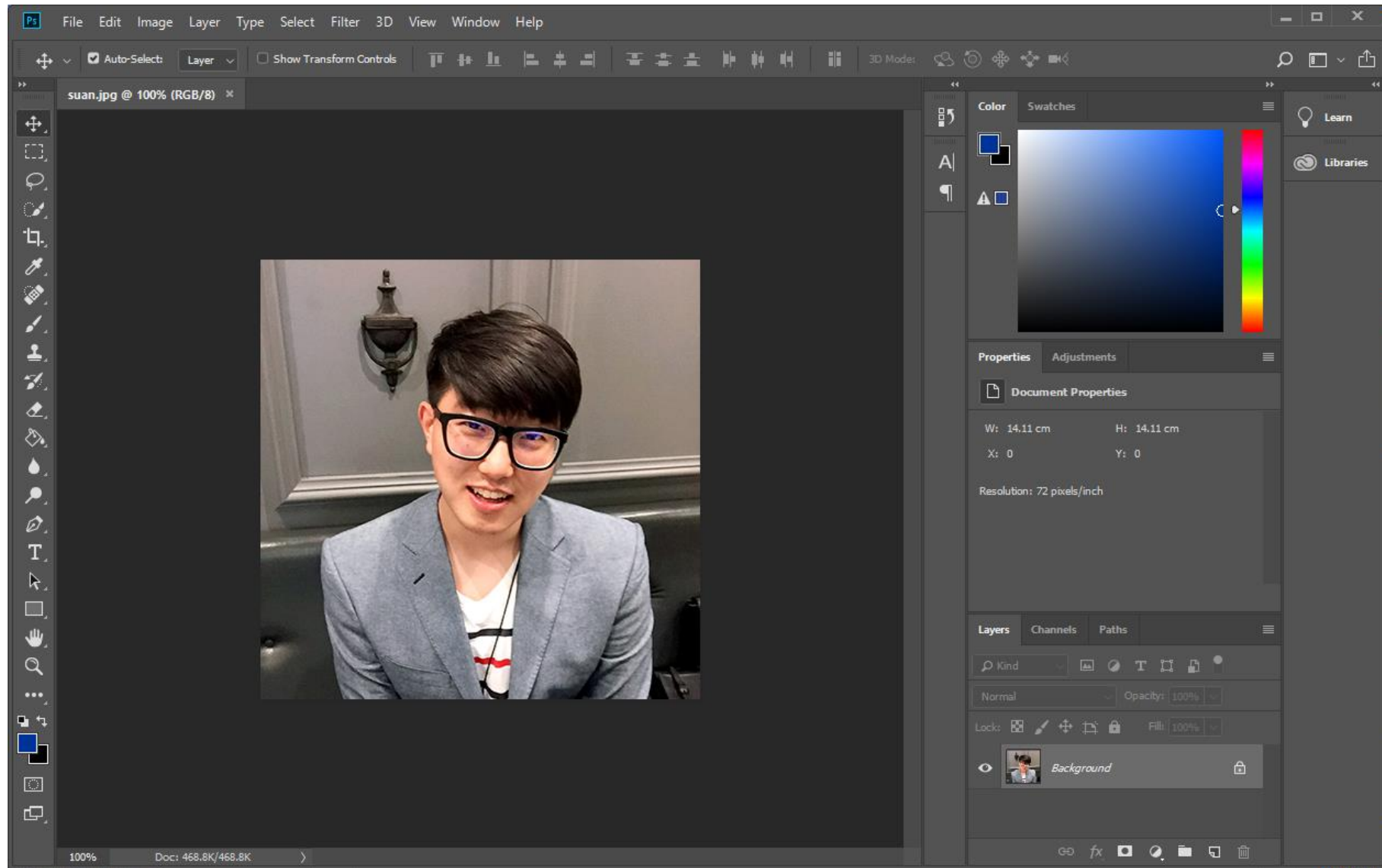


# 포토샵 소개

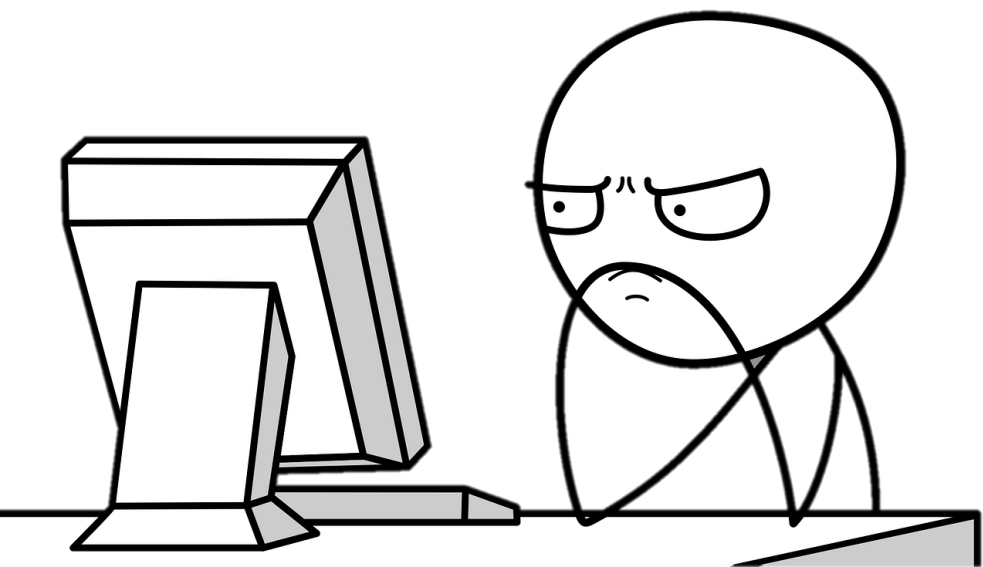
- 사진에 다양한 편집과 수정이 가능한 영상 처리 프로그램
- 사진 축소/확대, 상하/좌우 반전, 명도, 채도, 색조 변경 등의 사진 편집 가능



# 포토샵 화면



## 2. ImageMagick 설치























# ImageMagick 다운로드

## ■ 이미지매직 다운로드

- ImageMagick 6.9.10-14-Q16-x64-dll 버전으로 다운로드
- <http://www.imagemagick.org/download/binaries/ImageMagick-6.9.10-14-Q16-x64-dll.exe>

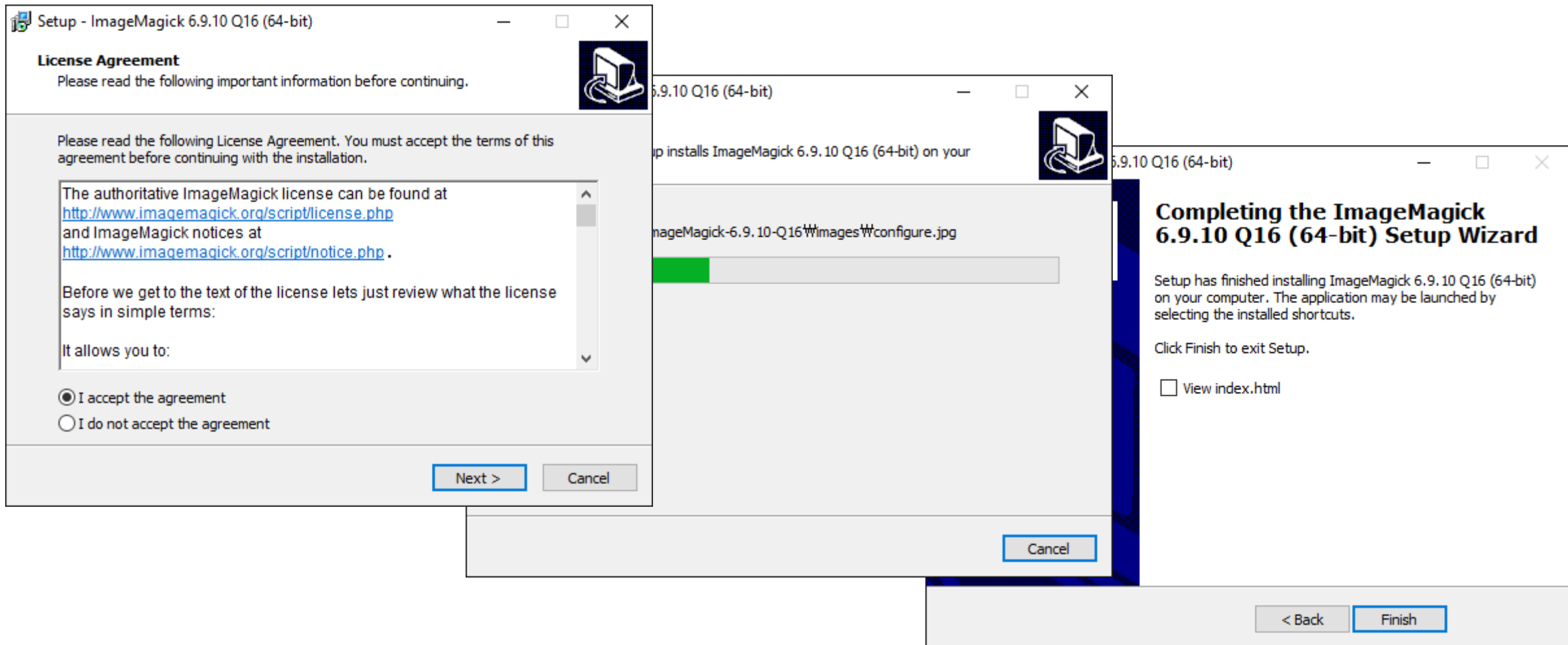
## Index of /download/binaries

	<a href="#">Name</a>	<a href="#">Last modified</a>	<a href="#">Size</a>	<a href="#">Description</a>
	<a href="#">Parent Directory</a>	-	-	-
	<a href="#">ImageMagick-6.9.10-1.&gt;</a>	2018-09-23 10:58	27M	
	<a href="#">ImageMagick-6.9.10-1.&gt;</a>	2018-09-23 10:59	801	
	<a href="#">ImageMagick-6.9.10-1.&gt;</a>	2018-09-23 10:58	48M	
	<a href="#">ImageMagick-6.9.10-1.&gt;</a>	2018-09-23 10:59	801	
	<a href="#">ImageMagick-6.9.10-1.&gt;</a>	2018-09-23 10:58	25M	
	<a href="#">ImageMagick-6.9.10-1.&gt;</a>	2018-09-23 10:59	801	
	<a href="#">ImageMagick-6.9.10-1.&gt;</a>	2018-09-23 10:58	57M	
	<a href="#">ImageMagick-6.9.10-1.&gt;</a>	2018-09-23 10:59	801	
	<a href="#">ImageMagick-6.9.10-1.&gt;</a>	2018-09-23 10:58	27M	
	<a href="#">ImageMagick-6.9.10-1.&gt;</a>	2018-09-23 10:58	801	
	<a href="#">ImageMagick-6.9.10-1.&gt;</a>	2018-09-23 10:58	65M	
	<a href="#">ImageMagick-6.9.10-1.&gt;</a>	2018-09-23 10:58	801	
	<a href="#">ImageMagick-6.9.10-1.&gt;</a>	2018-09-23 10:58	25M	
	<a href="#">ImageMagick-6.9.10-1.&gt;</a>	2018-09-23 10:58	801	
	<a href="#">ImageMagick-6.9.10-1.&gt;</a>	2018-09-23 10:58	57M	
	<a href="#">ImageMagick-6.9.10-1.&gt;</a>	2018-09-23 10:58	801	
	<a href="#">ImageMagick-6.9.10-1.&gt;</a>	2018-09-23 10:58	27M	
	<a href="#">ImageMagick-6.9.10-1.&gt;</a>	2018-09-23 10:58	801	
	<a href="#">ImageMagick-6.9.10-1.&gt;</a>	2018-09-23 10:58	65M	



# ImageMagick 설치

- [Next] 버튼을 계속 누르고, [Install] 버튼을 눌러 설치 진행



# 파이썬에 ImageMagick 라이브러리 추가

- Command Prompt 열기
  - [시작] - [실행] - cmd.exe
- ImageMagick 라이브러리 추가 명령어
  - pip install Wand



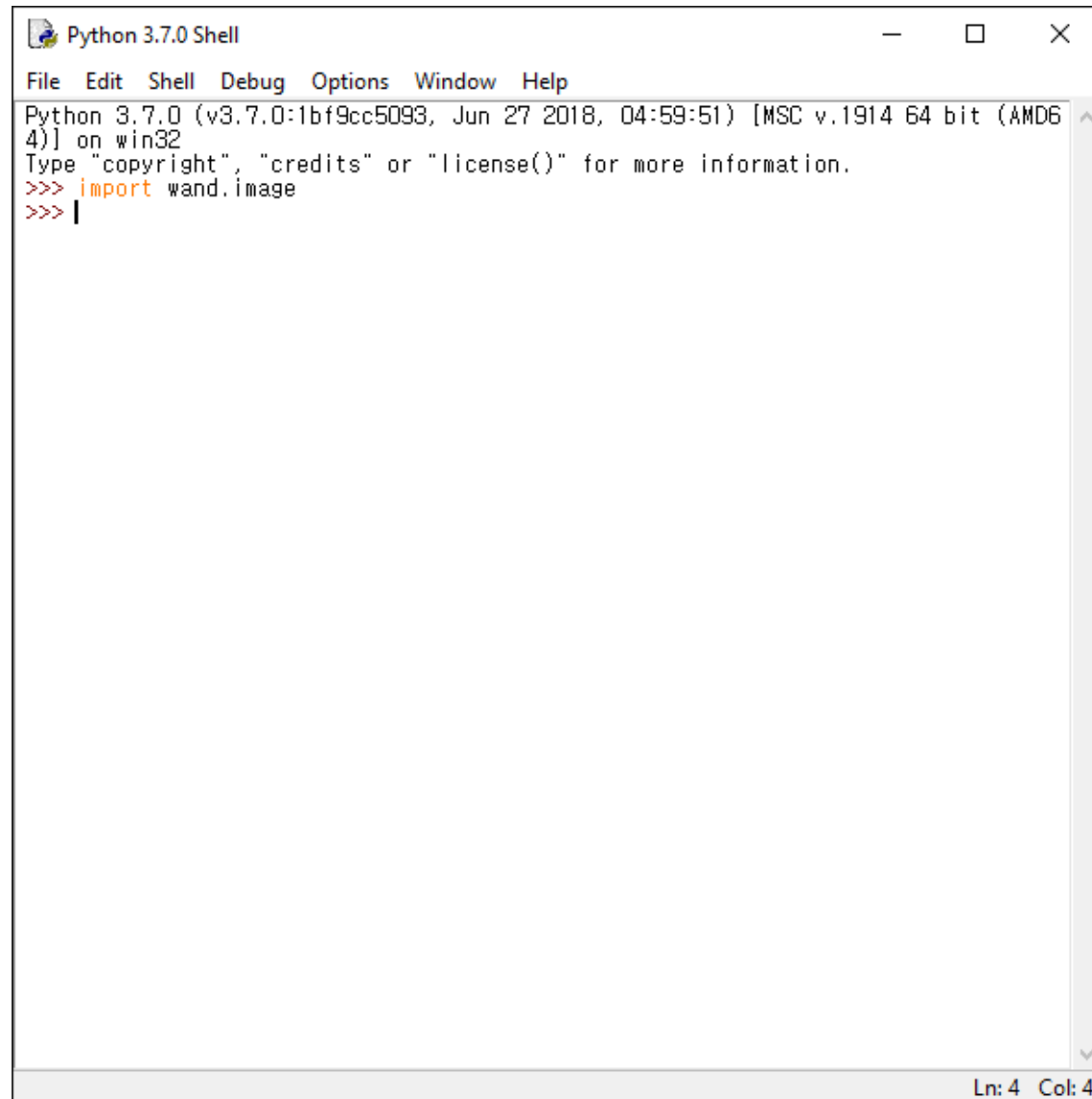
```
Select C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.17134.345]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Suan>pip install Wand
Collecting Wand
  Using cached https://files.pythonhosted.org/packages/c5/0e/4c7846ffac7a478578ff77c93d6aff3da2c181972d9447c74bfe1e87ac06/Wand-0.4.4.tar.gz
Installing collected packages: Wand
  Running setup.py install for Wand ... done
Successfully installed Wand-0.4.4
You are using pip version 10.0.1, however version 18.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

C:\Users\Suan>
```

# 파이썬에서 ImageMagick 설치 확인

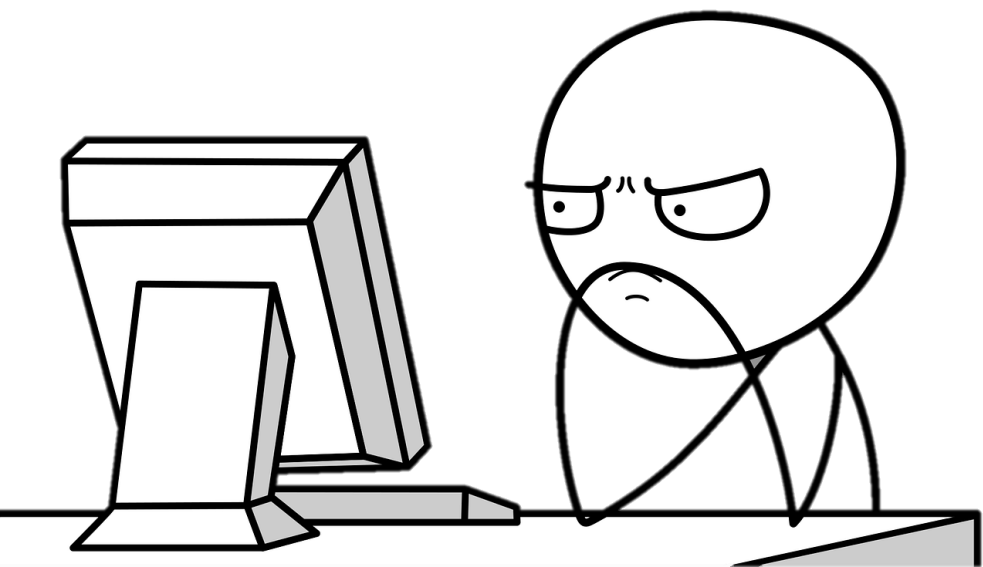
- Python Shell에서 명령어를 통해 ImageMagick 설치 확인
  - `import wand.image`



```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> import wand.image
>>> |
```

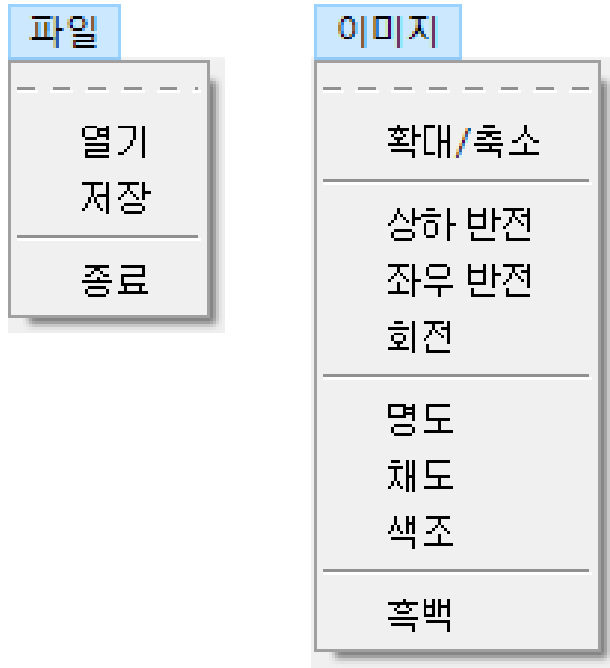
Ln: 4 Col: 4

### 3. 파이썬 포토샵 만들기



# GUI 구성

## ■ 메뉴 구성



## ■ 파일

- 열기
- 저장
- 종료

## ■ 이미지

- 확대/축소
- 상하 반전
- 좌우 반전
- 회전
- 명도
- 채도
- 색조
- 흑백

# 라이브러리

```
from tkinter import *  
from tkinter.filedialog import *  
from tkinter.simpledialog import *  
from wand.image import *
```

- tkinter: 파이썬에서 GUI 관련 모듈을 제공해주는 표준 윈도우 라이브러리
- filedialog: 파일 다이얼로그 박스
- simpledialog: 간단한 다이얼로그 박스
- wand.image: ImageMagick 라이브러리

# 함수 선언

<code>def displayImage() :</code> <code>pass</code>	<code>def funcFlop() :</code> <code>pass</code>
<code>def funcOpen() :</code> <code>pass</code>	<code>def funcRotate() :</code> <code>pass</code>
<code>def funcSave() :</code> <code>pass</code>	<code>def funcBrightness() :</code> <code>pass</code>
<code>def funcExit() :</code> <code>pass</code>	<code>def funcSaturation() :</code> <code>pass</code>
<code>def funcZoomInOut() :</code> <code>pass</code>	<code>def funcHue() :</code> <code>pass</code>
<code>def funcFlip() :</code> <code>pass</code>	<code>def funcBW() :</code> <code>pass</code>

## ■ 파일

- |      |                         |
|------|-------------------------|
| ■ 열기 | <code>funcOpen()</code> |
| ■ 저장 | <code>funcSave()</code> |
| ■ 종료 | <code>funcExit()</code> |

## ■ 이미지

- |         |                               |
|---------|-------------------------------|
| ■ 확대/축소 | <code>funcZoomInOut()</code>  |
| ■ 상하 반전 | <code>funcFlip()</code>       |
| ■ 좌우 반전 | <code>funcFlop()</code>       |
| ■ 회전    | <code>funcRotate()</code>     |
| ■ 명도    | <code>funcBrightness()</code> |
| ■ 채도    | <code>funcSaturation()</code> |
| ■ 색조    | <code>funcHue()</code>        |
| ■ 흑백    | <code>funcBW()</code>         |

# 전역 변수 선언

```
window, canvas, paper = None, None, None  
photo, newPhoto = None, None  
oriX, oriY = 0, 0
```

- 공통으로 접근할 전역 변수 선언
- photo: 원본 사진
- newPhoto: 이미지 처리된 결과 사진
- oriX, oriY: 원본 사진 폭과 높이



# 윈도우와 메뉴 구성

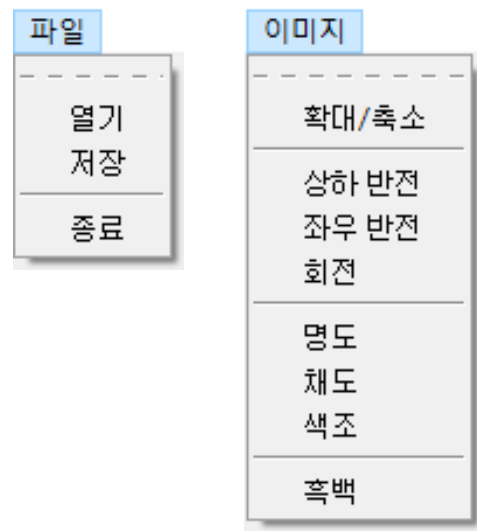
```
window = Tk()
window.geometry("400x400")
window.title("파이썬 포토샵")

mainMenu = Menu(window)
window.config(menu = mainMenu)
photo = PhotoImage()
pLabel = Label(window, image = photo)
pLabel.pack(expand = 1, anchor = CENTER)

fileMenu = Menu(mainMenu)
mainMenu.add_cascade(label = "파일", menu = fileMenu)
fileMenu.add_command(label = "열기", command = funcOpen)
fileMenu.add_command(label = "저장", command = funcSave)
fileMenu.add_separator()
fileMenu.add_command(label = "종료", command = funcExit)

imageMenu = Menu(mainMenu)
mainMenu.add_cascade(label = "이미지", menu = imageMenu)
imageMenu.add_command(label = "확대/축소", command = funcZoomInOut)
imageMenu.add_separator()
imageMenu.add_command(label = "상하 반전", command = funcFlip)
imageMenu.add_command(label = "좌우 반전", command = funcFlop)
imageMenu.add_command(label = "회전", command = funcRotate)
imageMenu.add_separator()
imageMenu.add_command(label = "명도", command = funcBrightness)
imageMenu.add_command(label = "채도", command = funcSaturation)
imageMenu.add_command(label = "색조", command = funcHue)
imageMenu.add_separator()
imageMenu.add_command(label = "흑백", command = funcBW)
window.mainloop()
```

- window.geometry: 윈도우 크기
- window.title: 윈도우 타이틀
- mainMenu: 메인 메뉴
  - fileMenu: 파일 메뉴
  - imageMenu: 이미지 메뉴



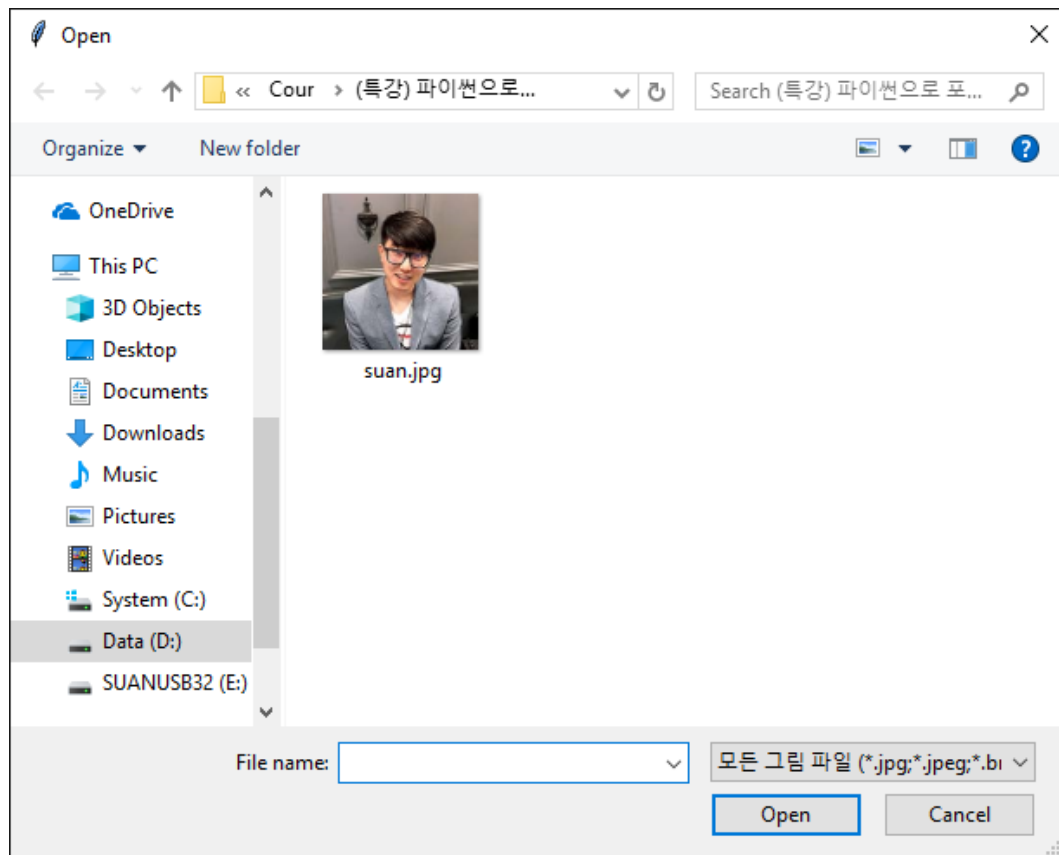
# 이미지 출력

```
def displayImage(img, width, height) :  
    global window, canvas, paper, photo, newPhoto, oriX, oriY  
  
    window.geometry(str(width) + "x" + str(height))  
    if canvas != None :  
        canvas.destroy()  
  
    canvas = Canvas(window, width = width, height = height)  
    paper = PhotoImage(width = width, height = height)  
    canvas.create_image((width/2, height/2), image = paper, state = "normal")  
  
    blob = img.make_blob(format = 'RGB')  
    for i in range(0, width) :  
        for k in range(0, height) :  
            r = blob[(i*3*width) + (k*3) + 0]  
            g = blob[(i*3*width) + (k*3) + 1]  
            b = blob[(i*3*width) + (k*3) + 2]  
            paper.put("#%02x%02x%02x" %(r, g, b), (k, i))  
  
    canvas.pack()
```

- 이미지 파일을 화면에 출력하는 함수
- 이미 canvas에 출력된 적이 있으면 제거
- paper.put()을 통해 점 단위로 화면에 출력

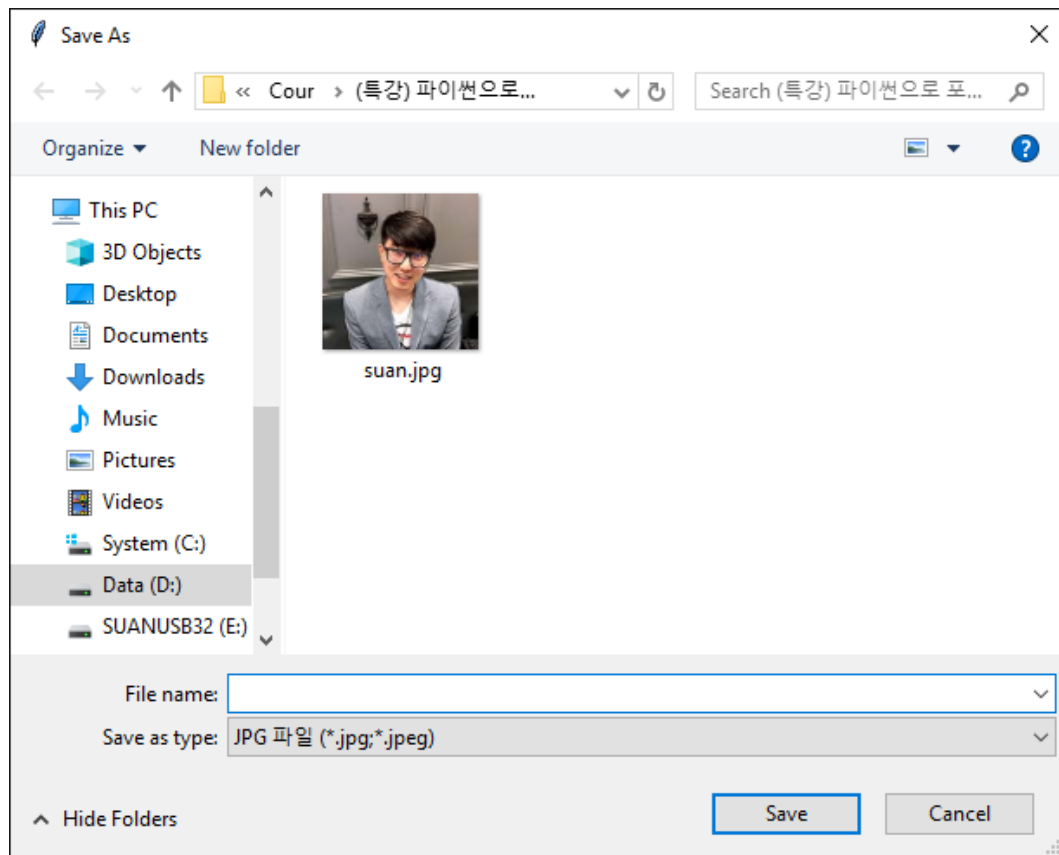
# 파일 열기

```
def funcOpen() :  
    global window, canvas, paper, photo, newPhoto, oriX, oriY  
    readFp = askopenfilename(parent = window, filetypes = (("모든 그림 파일", "*.jpg;*.jpeg;*.bmp;*.png;*.tif;*.gif"), ("모든 파일", "*.*")))  
    photo = Image(filename = readFp)  
    oriX = photo.width  
    oriY = photo.height  
    newPhoto = photo.clone()  
    newX = newPhoto.width  
    newY = newPhoto.height  
    displayImage(newPhoto, newX, newY)
```



# 파일 저장

```
def funcSave() :  
    global window, canvas, paper, photo, newPhoto, oriX, oriY  
  
    if newPhoto == None :  
        return  
    saveFp = asksaveasfile(parent = window, mode = "w", defaultextension = ".jpg", filetypes=(("JPG 파일", "*.jpg;*.jpeg"), ("모든 파일", "*.*")))  
    savePhoto = newPhoto.convert("jpg")  
    savePhoto.save(filename = saveFp.name)
```

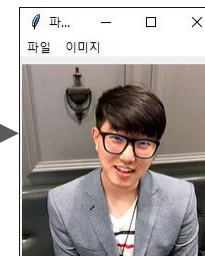
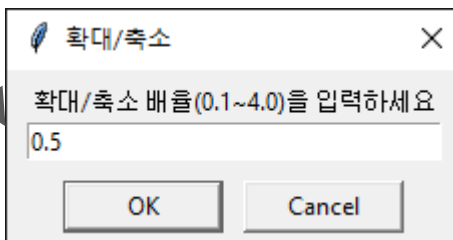
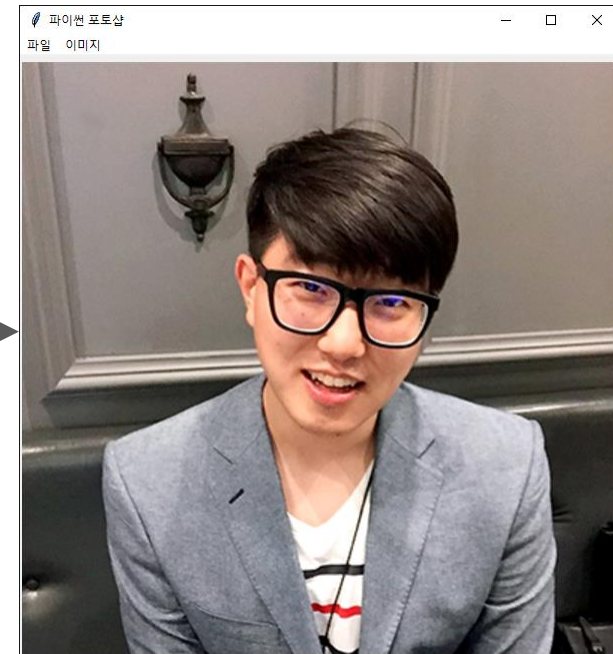
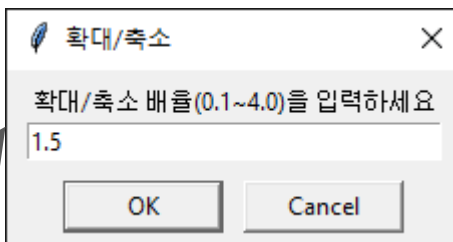


```
def funcExit() :  
    global window  
    window.destroy()
```

- 윈도우 종료

# 사진 확대/축소

```
def funcZoomInOut() :  
    global window, canvas, paper, photo, newPhoto, oriX, oriY  
    scale = askfloat("확대/축소", "확대/축소 배율(0.1~4.0)을 입력하세요", minvalue = 0.1, maxvalue = 4.0)  
    newPhoto = photo.clone()  
    newPhoto.resize(int(oriX * scale), int(oriY * scale))  
    newX = newPhoto.width  
    newY = newPhoto.height  
    displayImage(newPhoto, newX, newY)
```



# 사진 상하/좌우 반전

```
def funcFlip() :  
    global window, canvas, paper, photo, newPhoto, oriX, oriY  
    newPhoto = photo.clone()  
    newPhoto.flip()  
    newX = newPhoto.width  
    newY = newPhoto.height  
    displayImage(newPhoto, newX, newY)
```

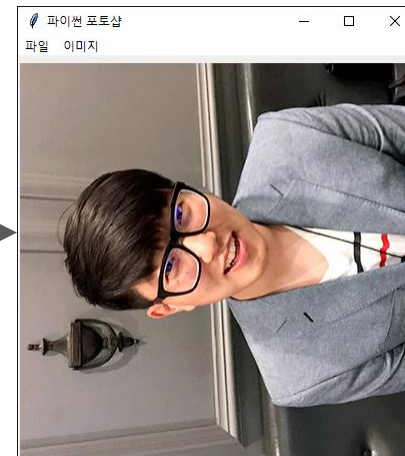
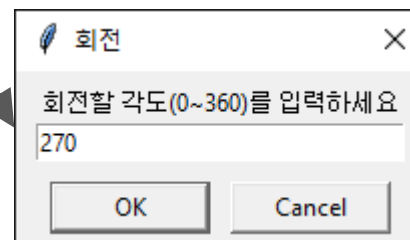
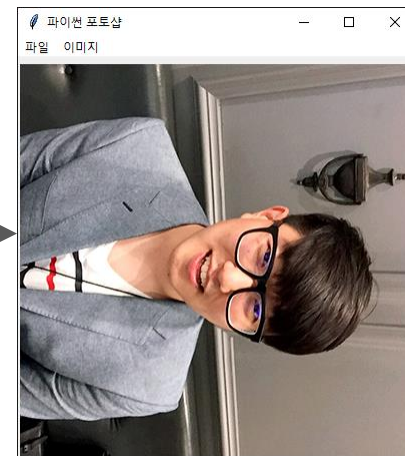
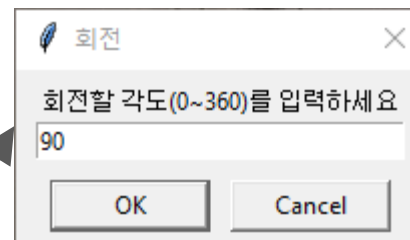
```
def funcFlop() :  
    global window, canvas, paper, photo, newPhoto, oriX, oriY  
    newPhoto = photo.clone()  
    newPhoto.flop()  
    newX = newPhoto.width  
    newY = newPhoto.height  
    displayImage(newPhoto, newX, newY)
```





# 사진 회전

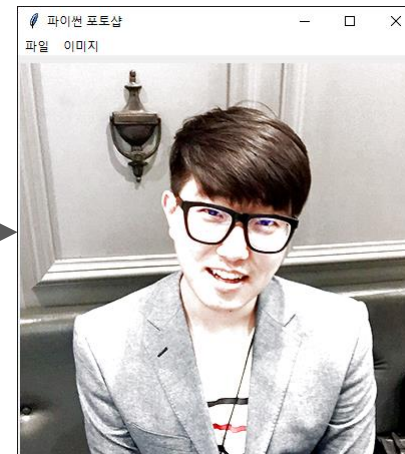
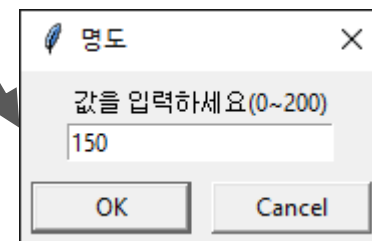
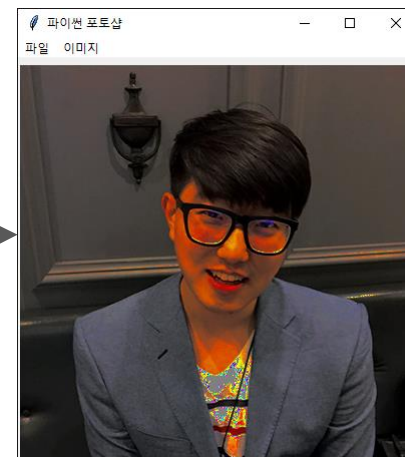
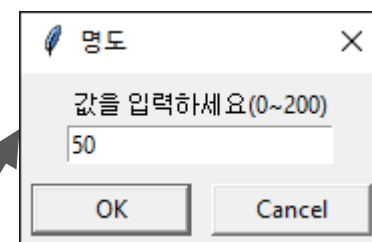
```
def funcRotate() :  
    global window, canvas, paper, photo, newPhoto, oriX, oriY  
    degree = askinteger("회전", "회전할 각도(0~360)를 입력하세요", minvalue = 0, maxvalue = 360)  
    newPhoto = photo.clone()  
    newPhoto.rotate(degree)  
    newX = newPhoto.width  
    newY = newPhoto.height  
    displayImage(newPhoto, newX, newY)
```





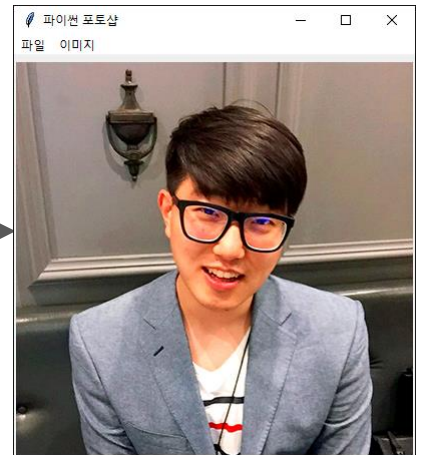
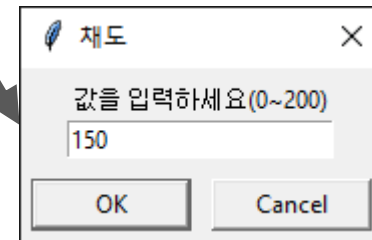
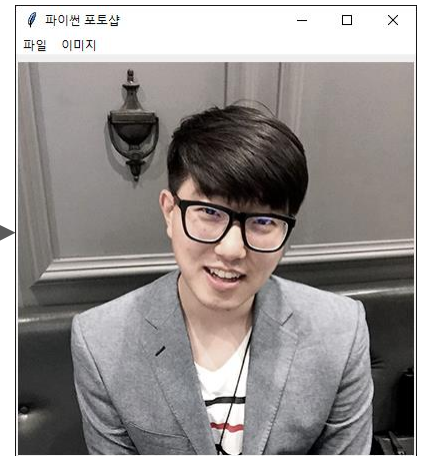
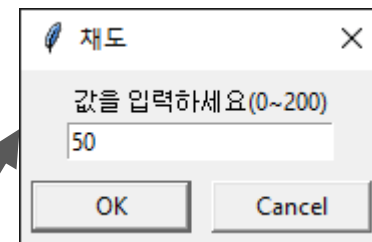
# 사진 명도

```
def funcBrightness() :  
    global window, canvas, paper, photo, newPhoto, oriX, oriY  
    value = askinteger("명도", "값을 입력하세요(0~200)", minvalue = 0, maxvalue = 200)  
    newPhoto = photo.clone()  
    newPhoto.modulate(value, 100, 100)  
    newX = newPhoto.width  
    newY = newPhoto.height  
    displayImage(newPhoto, newX, newY)
```



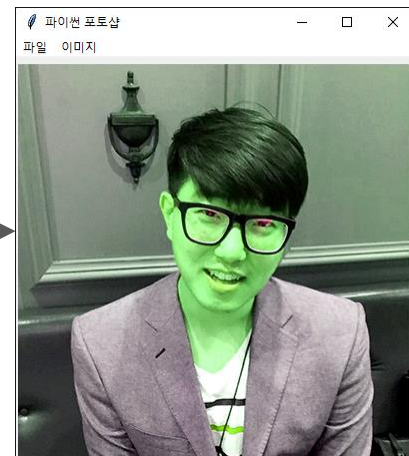
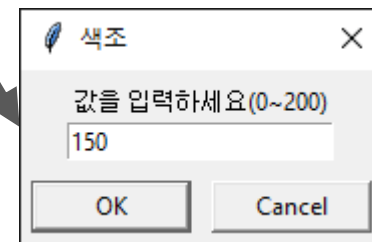
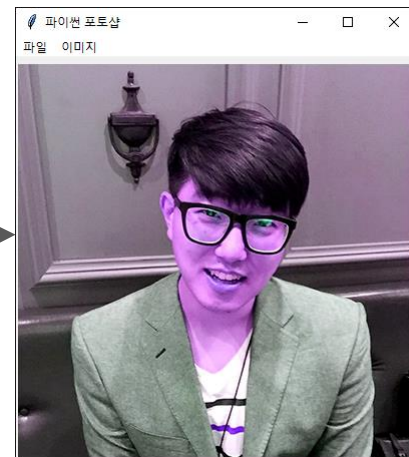
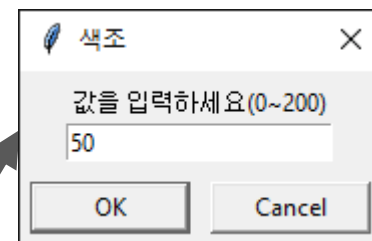
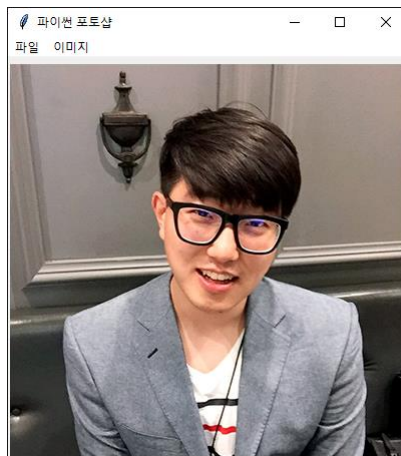
# 사진 채도

```
def funcSaturation() :  
    global window, canvas, paper, photo, photo2, oriX, oriY  
    value = askinteger("채도", "값을 입력하세요(0~200)", minvalue = 0, maxvalue = 200)  
    photo2 = photo.clone()  
    photo2.modulate(100, value, 100)  
    newX = photo2.width  
    newY = photo2.height  
    displayImage(photo2, newX, newY)
```



# 사진 색조

```
def funcHue() :  
    global window, canvas, paper, photo, newPhoto, oriX, oriY  
    value = askinteger("색조", "값을 입력하세요(0~200)", minvalue = 0, maxvalue = 200)  
    newPhoto = photo.clone()  
    newPhoto.modulate(100, 100, value)  
    newX = newPhoto.width  
    newY = newPhoto.height  
    displayImage(newPhoto, newX, newY)
```



# 사진 흑백 변환

```
def funcBW() :  
    global window, canvas, paper, photo, newPhoto, oriX, oriY  
    newPhoto = photo.clone()  
    newPhoto.type = "grayscale"  
    newX = newPhoto.width  
    newY = newPhoto.height  
    displayImage(newPhoto, newX, newY)
```



