

# Step 15: Pillow? Pillow!

Instructor: Eunil Park (eunilpark@skku.edu)



SUNG KYUN KWAN  
UNIVERSITY



## Simple Image Control

1. Image 편집 라이브러리 - Pillow

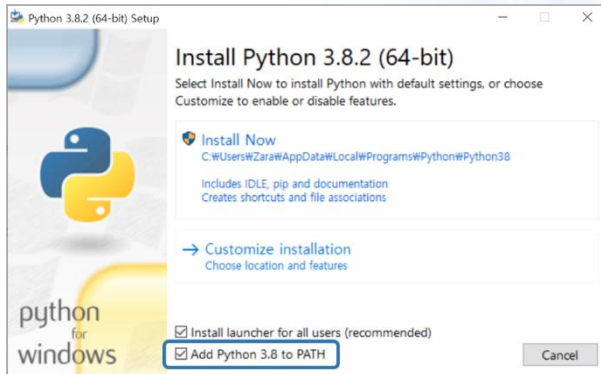
## Pillow

- 이미지 편집 라이브러리
  - 새 영역 경험해보기
  - 새 라이브러리 배워보기
  - 설치 라이브러리 경험하기

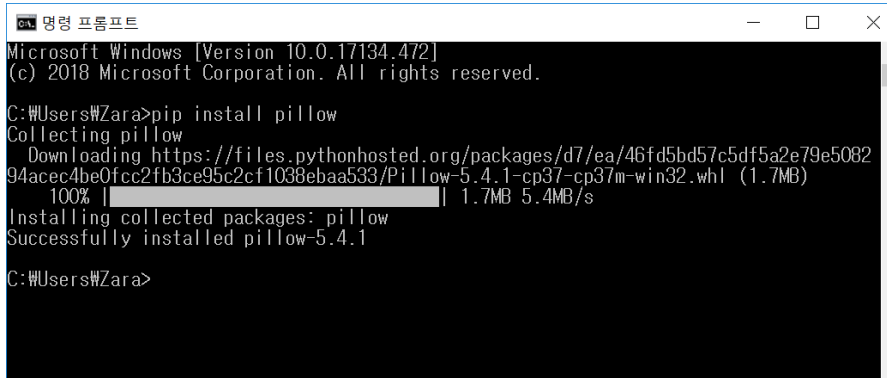


## Pillow

1. pip가 설치되어 있어야 새 라이브러리를 설치할 수 있음.



2. [시작] – [명령 프롬프트] 또는 Windows +R 키를 눌러 command 입력 후 실행
3. pip install pillow



```
Microsoft Windows [Version 10.0.17134.472]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Zara>pip install pillow
Collecting pillow
  Downloading https://files.pythonhosted.org/packages/d7/ea/46fd5bd57c5df5a2e79e5082
94acec4be0fcc2fb3ce95c2cf1038ebaa533/Pillow-5.4.1-cp37m-win32.whl (1.7MB)
    100% |#####| 1.7MB 5.4MB/s
Installing collected packages: pillow
Successfully installed pillow-5.4.1

C:\Users\Zara>
```

새 라이브러리 설치하는 법  
> pip install 라이브러리명

## 일반적인 방법으로 설치할 수 없을 때

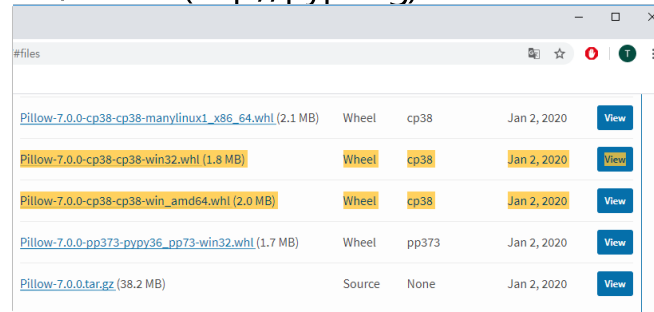
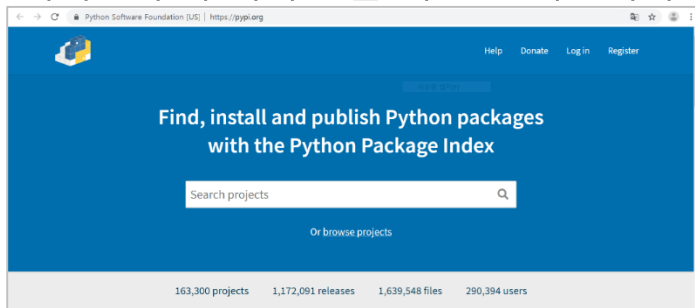
pip 버전 업그레이드 실시

```
> python -m pip install --upgrade pip
```

pip 명령을 찾을 수 없다는 메시지가 뜰 때

[제어판] - [시스템 및 보안] - [시스템]에서 왼쪽 [고급 시스템 설정] - [고급] 탭 - [환경 변수]에서 변수 Path를 더블클릭한 후 [새로 만들기]를 통해 [..\\파이썬설치폴더\\Scripts\\] 폴더를 지정

위의 경우가 되지 않을 때 → 인터넷에서 모듈 다운로드(<http://pypi.org>)



다운로드한 파일을 [..\\파이썬 설치폴더\\Scripts\\]에 저장한 후 [pip3 install 파일명] 실행

## Pillow 시작하기

### 1. pillow 임포트하기

```
>>> from PIL import Image          # PIL 안의 Image 클래스 이용
```

### 2. 객체에 이미지 저장

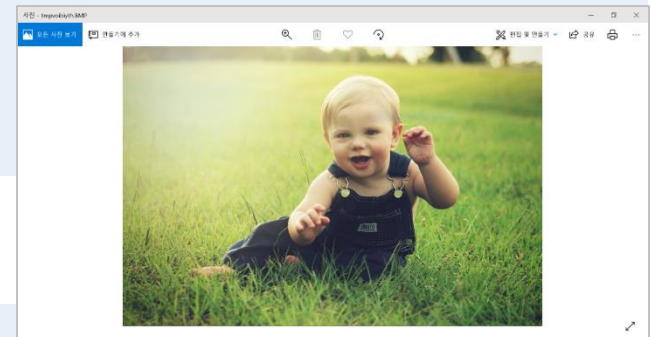
```
>>> img = Image.open('baby.jpg')
```

### 3. 이미지와 포맷정보(jpg, png 등) 알아보기

```
>>> print(img.size)
(1280, 853)
>>> print(img.format)
JPEG
```

### 4. 이미지 출력하기

```
>>> img.show()
```





## 좌표를 이용하여 이미지 잘라내기(crop)

### 1. 이미지 사이즈 확인

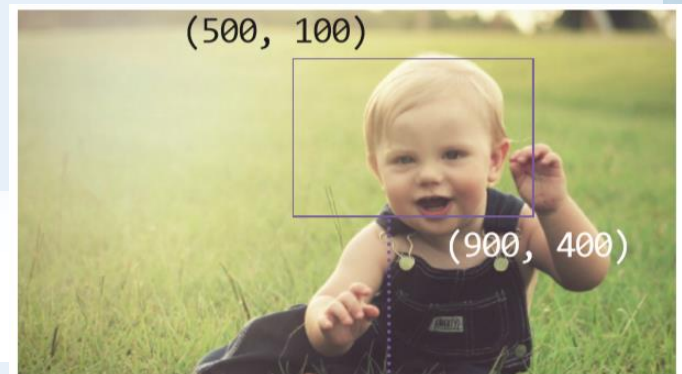
```
>>> from PIL import Image  
>>> img = Image.open('baby.jpg')  
>>> print(img.size)  
(1280, 853)
```

### 2. img.crop(좌표)로 잘라내기

```
>>> xy = (500, 100, 900, 400)  
>>> crop_img = img.crop(xy)  
>>> crop_img.show()
```

### 3. 잘라낸 이미지 다른 이름으로 저장하기

```
>>> crop_img.save('crop_baby.jpg')
```



## .paste를 이용하여 이미지 붙여넣기

### 1. 병합할 이미지 사이즈 알아내기

```
>>> from PIL import Image
>>> img = Image.open('baby.jpg')
>>> crop_img = Image.open('crop_baby.jpg')
>>> print(crop_img.size)
(400, 300)
```

#기존 사진  
#병합할 사진

### 2. 붙여넣을 곳 좌표 지정하여 붙이기 .paste(병합할 사진, 크기)

```
>>> xy = (0, 0, 400, 300)
>>> img.paste(crop_img, xy)
```

### 3. 이미지 확인하고 저장하기

```
>>> img.save('paste_img.jpg')
>>> img.show()
```

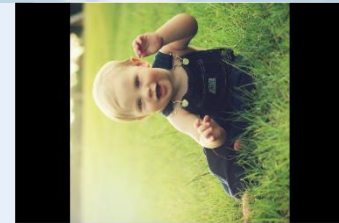




## 이미지 회전, 리사이즈, 뒤집기

### 1. 이미지 회전 .rotate(각도)

```
from PIL import Image
img = Image.open('baby.jpg')
img_r = img.rotate(90)      #img_r에 90도 회전한 것을 할당
img_r.show()
```



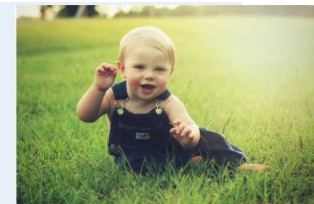
### 2. 사이즈 조절 .resize((x, y))

```
from PIL import Image
img = Image.open('baby.jpg')
img_s = img.resize((500, 500)) #사이즈를 (x, y) 형태로 지정
img_s.show()
```

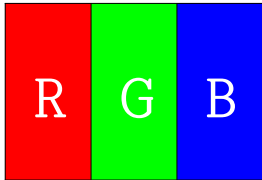


### 3. 이미지 확인하고 저장하기 .transpose(Image.FLIP\_LEFT\_RIGHT) TOP BOTTOM LEFT RIGHT

```
from PIL import Image
img = Image.open('baby.jpg')
img_opp = img.transpose(Image.FLIP_LEFT_RIGHT) #좌우반전 FLIP_LEFT_RIGHT
img_opp.show()
```



## RGB란? / RGB 분리하기



빛의 삼원색을 이용하여 색을 표현하는 방식  
각 색마다 0-255의 색이 있고 각각을 조절하여 우리 눈에 다양한 색으로 보임  
현재까지 거의 모든 패널은 이와 같은 패러다임 유지

.mode 확인 후 색상값 분리하여 담기

```
>>> from PIL import Image
>>> img = Image.open('baby.jpg')
>>> print(img.mode)
```

RGB

```
>>> r, g, b = img.split()
>>> r.show()
>>> g.show()
>>> b.show()
```

#R,G,B로 색상값을 split() 즉, 잘라낸다.



## 분리된 RGB 합치기

.merge(모드, 순서) 이용하여 RGB 합치기

```
>>> RGB_img = Image.merge("RGB", (r,g,b))    #RGB모드 r,g,b로 merge
>>> RGB_img.show()
>>> BRG_img = Image.merge("RGB", (b,r,g))    #RGB모드 b,r,g로 merge
>>> BRG_img.show()
>>>
```



Image.merge("RGB", (r,g,b))



Image.merge("RGB", (b,r,g))

RGB로 다른 방식의 조합도 가능함.

## ImageFilter 클래스에서 이미지 필터 사용하기

### 1. PIL 내의 ImageFilter 클래스 사용하기

```
>>> from PIL import Image          # 이미지 정보와 편집을 위한 모듈 불러오기
>>> from PIL import ImageFilter    # 이미지 필터 처리 모듈 불러오기
>>> img = Image.open('baby.jpg')
>>>
```

### 2. BLUR, EMBOSS, CONTOUR 필터 적용해보기

```
>>> img1 = img.filter(ImageFilter.BLUR)
>>> img2 = img.filter(ImageFilter.EMBOSS)
>>> img3 = img.filter(ImageFilter.CONTOUR)

>>> img1.show()
>>> img2.show()
>>> img3.show()
```



## tkinter와 pillow 함께 사용하기

### 1. PIL 내의 imageTK 모듈 이용하기

```
>>> from PIL import Image, ImageTk  
>>> import tkinter      #[from tkinter import *]는 imageTK와 겹치는 메서드로 충돌이 일어남
```

### 2. tkinter에 있는 캔버스 기능 이용하기 .Canvas

```
>>> root = tkinter.Tk()  
>>> canvas = tkinter.Canvas(root, width=1200, height=867)  
>>> canvas.pack()  
>>> img = Image.open('baby.jpg')
```

### 3. 이미지를 tkinter에서 사용할 수 있는 형식으로 변환하기 imageTk.PhotoImage(img객체)

```
>>> img_tk = ImageTk.PhotoImage(img)
```

### 4. 변환한 이미지를 캔버스에 띄우기 / 캔버스명.create\_image(x, y, image= [ ])

```
>>> canvas.create_image(600, 430, image=img_tk)
```



## tkinter와 pillow 함께 사용하기

```
from PIL import Image, ImageTk
import tkinter
root = tkinter.Tk()
canvas = tkinter.Canvas(root, width=1200, height=867)
canvas.pack()
img = Image.open('baby.jpg')
img_tk = ImageTk.PhotoImage(img)
canvas.create_image(600, 430, image=img_tk)
```

#캔버스 지정  
#캔버스 생성

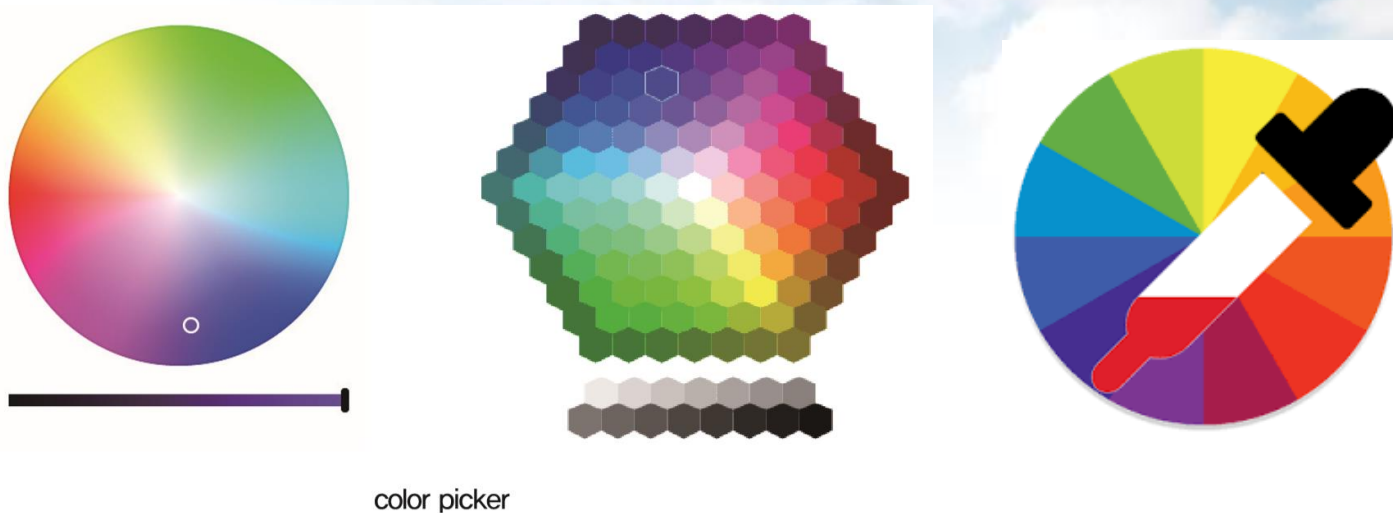
#이미지 tkinter 형식으로  
#캔버스에 이미지 생성





## 이벤트 처리를 이용하여 RGB picker 만들기

그림의 한 곳을 클릭하면 해당 RGB 값이 출력되는 프로그램



## 1. 이벤트 바인딩을 이용하여 클릭하는 곳의 좌표 알기

```
import tkinter

root = tkinter.Tk()
cv = tkinter.Canvas(root, width=500, height=300)
def click(event):
    print("클릭 위치 :", event.x, event.y)
cv.bind("<Button-1>", click)
cv.pack()

root.mainloop()
```

# 클릭 이벤트  
# event.x는 클릭이벤트시의 x좌표  
#<Button-1>은 마우스 왼쪽클릭

```
클릭 위치 : 235 129
클릭 위치 : 232 128
클릭 위치 : 74 122
클릭 위치 : 297 129
...
```

## 2. 캔버스에 그림 불러오기

```
from PIL import Image, ImageTk  
  
img = Image.open('baby.jpg')  
img_tk = ImageTk.PhotoImage(img)  
cv.create_image(250,250, image=img_tk)
```

3. `.getpixel((x, y))`을 이용하여 해당 부분의 픽셀 구하기

```
r, g, b = img.getpixel((event.x, event.y))  
print("R:%d G:%d B:%d"%(r,g,b))
```

#해당 좌표의 pixel값 구하기



```
클릭 위치 : 316 171  
R:230 G:237 B:196  
클릭 위치 : 204 170  
R:221 G:231 B:196  
...
```

## 전체 소스

```
import tkinter
from PIL import Image, ImageTk

root = tkinter.Tk()
cv = tkinter.Canvas(root, width=500, height=300)
def click(event):
    print("클릭 위치 :", event.x, event.y)
    r, g, b = img.getpixel((event.x, event.y))
    print("R:%d G:%d B:%d"%(r,g,b))

cv.bind("<Button-1>", click)
cv.pack()

img = Image.open('baby.jpg')
img_tk = ImageTk.PhotoImage(img)
cv.create_image(250,250, image=img_tk)

root.mainloop()
```

#클릭이벤트 (매개변수 : 이벤트 정보)  
#이벤트 내 x,y좌표  
#해당좌표 픽셀 정보

#<Button-1> 이벤트 바인딩

#이미지 형식 tkinter  
#캔버스에 이미지 출력

## 이미지, 영상 편집을 더 하고 싶다면...

1. Pillow의 20여가지가 넘는 라이브러리들 이용
2. 그래픽을 통한 얼굴인식, 데이터 처리등 → OpenCV

<https://docs.python.org/3>

<https://stackoverflow.com/>

## Summary

1. Pillow!



# Thanks

Step 15: Pillow? Pillow!

Instructor: Eunil Park ([eunilpark@skku.edu](mailto:eunilpark@skku.edu))

