

- Pillow یک کتابخانه پایتونی برای پردازش تصاویر است
- برای نصب pillow دستور زیر را وارد میکنیم:

```
pip install pillow
```

- برای استفاده از pillow دستور زیر را در برنامه خود اضافه میکنیم:

```
from PIL import Image
```

- معرفی متد ها و دستورات کتابخانه **pillow**
- برای دسترسی به یک تصویر از متد **open()** استفاده میکنیم، این متد آدرس تصویر مد نظر را به عنوان ورودی دریافت میکند

```
PIL open()

from PIL import Image

my_image = Image.open('a.jpg')

type(my_image)

my_image
my_image.show() # in IDEs
```

- چند ویژگی پر کاربرد در رابطه با تصاویر، **size**، نام فایل و فرمت آن هاست



PIL attr

```
# (width, height)
```

```
my_image.size
```

```
my_image.filename
```

```
my_image.format
```

- برای برش تصاویر می توانید از متد `crop()` روی شی تصویر استفاده کنید. متد `crop()` یک ناحیه مستطیل شکل را از این تصویر برمی گرداند. این متد یک کادر ۴ تایی است که مختصات پیکسل های چپ، بالا، راست و پایین را مشخص می کند

PIL crop() 1

```
pencils = Image.open("pencils.jpg")  
pencils.size
```



PIL crop() 2

```
# Start at top corner (0,0)
x = 0
y = 0

# Grab about 10% in y direction , and about 30% in x direction
w = 1950/3
h = 1300/10

pencils.crop((x,y,w,h))
```

PIL crop() 3

```
x = 0
y = 1100
w = 1950/3
h = 1300
pencils.crop((x,y,w,h))
```

- برای تغییر سایز یک تصویر از متد `resize()` استفاده میشود که طول و ارتفاع جدید تصویر را به عنوان آرگمان ورودی دریافت میکند

PIL resize()

```
w,h = my_image.size
new_h = int(h/10)
new_w = int(w/10)

my_new_img = my_image.resize((new_w,new_h))
```



- برای چرخش یک تصویر از متد `rotate()` استفاده میشود که زاویه چرخش به عنوان آرگومان ورودی به متد داده میشود

PIL rotate()

```
my_new_img.rotate(45)
```

```
my_new_img.rotate(45, expand=True)
```



- برای ذخیره یک تصویر از متد `save()` استفاده میشود که نام را به عنوان آرگمان ورودی دریافت میکند

PIL save()

```
my_new_img.save("b.jpg")
```

- یک فایل با پسوند فایل CSV یک فایل Comma Separated Values است
- همه فایل های CSV متون ساده ای حاوی کاراکترهای الفبایی هستند و داده های موجود در آنها را به شکل جدولی میباشند
- فایل های CSV بسیار شبیه به فایل های اکسل فرمت بندی می شوند، اما شامل انواع داده ای برای مقادیر خود نمیشوند و فقط شامل رشته میشوند
- فایل های با فرمت CSV معمولاً برای تبادل داده ها، زمانی که مقدار زیادی وجود دارد، بین برنامه های مختلف استفاده می شود. برنامه های پایگاه داده، نرم افزارهای تحلیلی و سایر برنامه هایی که حجم عظیمی از اطلاعات را ذخیره می کنند (مانند مخاطبین و داده های مشتری)، معمولاً از فرمت CSV پشتیبانی می کنند.

- کار با فایل های CSV در پایتون به راحتی کار با فایل های متنی نیست و نیاز به افزودن ماژول خاص این کار دارد
- مراحل دسترسی به یک فایل CSV :
  1. اضافه کردن ماژول CSV
  2. دسترسی به فایل با متد `open` و `encoding="utf-8"`
  3. استفاده از متد `reader` از ماژول `csv`
  4. نمایش مقادیر در قالب `list`



CSV

```
import csv

data = open('example.csv', encoding="utf-8")

csv_data = csv.reader(data)
csv_list = list(csv_data)
```



- تمرینات
- فایل example.csv را در نظر بگیرید
- تمرین ۱: ۳ آیتم ابتدایی لیست را نمایش دهید
- تمرین ۲: ۵ آیتم ابتدایی لیست را در خطوط مجزا چاپ کنید
- تمرین ۳: ایمیل ۱۵ نفر اول لیست را در قالب یک لیست مجزا نمایش دهید
- تمرین ۴: نام کامل ۱۵ نفر اول لیست را در قالب یک لیست مجزا نمایش دهید
- تمرین ۵: تعداد خانم و آقا در این فایل هر کدام چند نفر است



csv ex1

```
import csv

data = open('example.csv', encoding="utf-8")

csv_data = csv.reader(data)
csv_list = list(csv_data)

csv_list[:3]
```

csv ex2

```
for item in csv_list[:5]:  
    print(item)
```

csv ex3

```
emails = []  
for item in csv_list[1:15]:  
    emails.append(item[3])
```



csv ex4

```
full_names = []  
  
for line in csv_list[1:15]:  
    full_names.append(line[1]+' '+line[2])
```



csv ex5

```
gender = []  
  
for item in csv_list[1:]:  
    gender.append(item[4])  
  
gender.count("Male")  
gender.count("Female")
```