

مروری بر کار با تصاویر

با استفاده از قدرت برخی از کتابخانه‌های متداول که می‌توانید نصب کنید، مانند PILLOW، پایتون قابلیت کار با تصاویر را برای وظایف ساده پیدا می‌کند. می‌توانید با اجرای دستور زیر کتابخانه Pillow را نصب کنید:

```
pip install pillow
```

در صورت وجود هرگونه مشکل، می‌توانید به <http://pillow.readthedocs.io/en/3.4.x/installation.html> آنها درباره نحوه نصب مراجعه کنید. اما برای اکثر کامپیوترها، روش نصب ساده با استفاده از pip باید کار کند.

توجه: هنگام کار با تصاویر در دفترچه یادداشت Jupyter ممکن است با خطای زیر مواجه شوید:

```
IOPub data rate exceeded.  
The notebook server will temporarily stop sending output  
to the client in order to avoid crashing it.  
To change this limit, set the config variable  
`--NotebookApp.iopub_data_rate_limit`.
```

اگر این هشدار را دریافت کردید، تلاش کنید دفترچه یادداشت را در خط فرمان متوقف کنید و سپس با دستور زیر آن را مجدداً راه‌اندازی کنید:

```
jupyter notebook --NotebookApp.iopub_data_rate_limit=1.0e10
```

در خط فرمان. در واقع این یک "پرچم" است که مقدار محدودیت باید در طول این نشست از دفترچه یادداشت Jupyter که در حال اجرا است، افزایش یابد.

کار با کتابخانه Pillow

باز کردن تصاویر

شما می‌توانید از کتابخانه Pillow برای باز کردن فایل‌های تصویر استفاده کنید. در یک دفترچه یادداشت Jupyter، برای نمایش فایل، کافیه نام متغیری که تصویر را نگهداری می‌کند را تایپ کنید. برای محیط‌های توسعه یکپارچه (IDE) دیگر، متغیر تصویر دارای یک متد به نام `show()` <https://stackoverflow.com/questions/28139637/how-can-i-display-an-image-using-pillow>.

In [1]:

```
1 from PIL import Image
```

In [2]:

```
1 mac = Image.open('example.jpg')
```

In [5]:

```
1 # Note this is a specialized file type from PIL (pillow)
2 type(mac)
```

Out[5]:

PIL.JpegImagePlugin.JpegImageFile

In [6]:

```
1 # Only for jupyter notebook , use mac.show() for other IDEs
2 mac
```

Out[6]:



اطلاعات تصاویر

In [7]:

```
1 # (width, height)
2 mac.size
```

Out[7]:

(1993, 1257)

In [8]:

```
1 mac.filename
```

Out[8]:

```
'example.jpg'
```

In [9]:

```
1 mac.format_description
```

Out[9]:

```
'JPEG (ISO 10918)'
```

برش تصاویر

برای برش تصاویر (یعنی گرفتن یک بخش فرعی) می‌توانید از متد `crop()` بر روی شیء تصویر استفاده کنید. متد `crop()` یک ناحیه مستطیلی را از این تصویر برمی‌گرداند. جعبه یک چهارتایی است که شامل مختصات پیکسل چپ، بالا، راست و پایین است.

توجه کنید! اگر به رشته مستندات نگاه کنید، مشخص می‌شود که تاپلی که در آن مقادیر را ارسال می‌کنید به عنوان (x, y, w, h) تعریف شده است. این متغیرها ممکن است یکم همراه‌کننده باشند. در واقع، این مقادیر نه ارتفاع و نه عرضی است که ارسال می‌شوند، بلکه مختصات پایانی عرض و ارتفاع شما هستند.

همه مختصات جعبه (x, y, w, h) از گوشه بالا و سمت چپ تصویر اندازه‌گیری می‌شوند. دوباره، همه این چهار مقدار مختصات هستند!

In [10]:

```
1 mac.crop((0,0, 100, 100))
```

Out[10]:



برای تصویر مک، این یک نمایش کاربردی نیست. به جای آن، از یک تصویر دیگر استفاده کنیم:

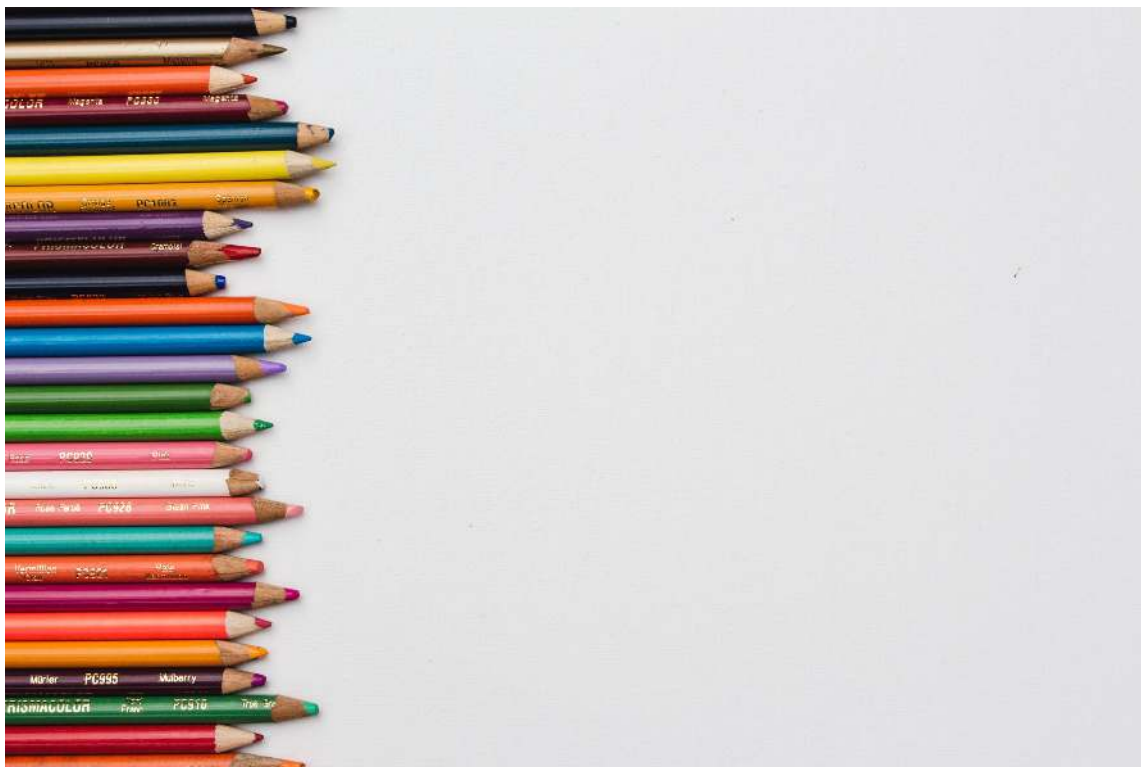
In [11]:

```
1 pencils = Image.open('pencils.jpg')
```

In [12]:

```
1 pencils
```

Out[12]:



حالا بیایید سعی کنیم برخی از مدادهای برتر را از گوشه بگیریم.

In [13]:

```
1 pencils.size
```

Out[13]:

(1950, 1300)

In [14]:

```
1 # Start at top corner (0,0)
2 x = 0
3 y = 0
4
5 # Grab about 10% in y direction , and about 30% in x direction
6
7 w = 1950 / 3
8 h = 1300 / 10
9
10 pencils.crop((x, y, w, h))
11
```

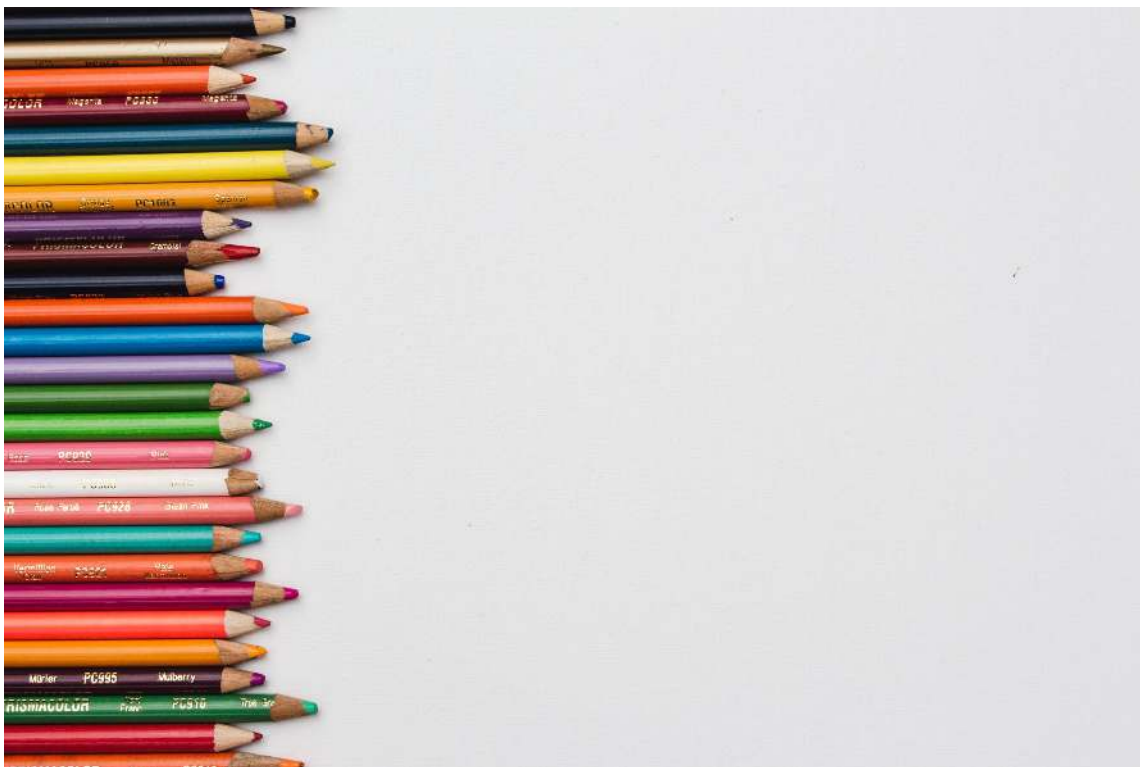
Out[14]:



In [15]:

```
1 pencils
```

Out[15]:



حالا بیا یید مدادها را از پایین بگیریم.

In [16]:

```
1 pencils.size
```

Out[16]:

(1950, 1300)

In [17]:

```
1 x = 0
2 y = 1100
3 w = 1950/3
4 h = 1300
```

In [18]:

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

Out[18]:



حالا به عکس مک برگردیم و ببینیم آیا می‌توانیم فقط کامپیوتر را بگیریم:

In [19]:

```
1 mac
```

Out[19]:



In [20]:

```
1 mac.size
```

Out[20]:

```
(1993, 1257)
```

In [21]:

```
1 halfway = 1993 / 2
```

In [22]:

```
1 x = halfway - 200  
2 w = halfway + 200
```

In [25]:

```
1 y = 800  
2 h = 1257
```

In [26]:

```
1 mac.crop((x, y, w, h))
```

Out[26]:



کپی کردن و چسباندن تصاویر

ما می‌توانیم با استفاده از متد `copy()` کپی‌هایی از تصاویر ایجاد کنیم و با استفاده از متد `paste()` تصاویر را بر روی تصاویر دیگر چسبانیم.

In [27]:

```
1 computer = mac.crop((x, y, w, h))
```

In [28]:

```
1 mac.paste(im = computer, box = (0, 0))
```


In [29]:

```
1 mac
```

Out[29]:



In [30]:

```
1 mac.paste(im = computer, box = (796, 0))
```

In [31]:

```
1 mac
```

Out[31]:



تغییر اندازه

شما می‌توانید از متد `resize()` برای تغییر اندازه یک تصویر استفاده کنید.

In [32]:

```
1 mac.size
```

Out[32]:

```
(1993, 1257)
```

In [33]:

```
1 h, w = mac.size
```

In [34]:

```
1 new_h = int(h / 3)
2 new_w = int(w / 3)
```

In [35]:

```
1 # Note this is not permanent change
2 # for permanent change, do a reassignment
3 # e.g. mac = mac.resize((100,100))
4 mac.resize((new_h, new_w))
```

Out[35]:



همچنین می‌توانید آن را کشیده و فشرده کنید.

In [36]:

```
1 mac.resize((3000,500))
```

Out[36]:



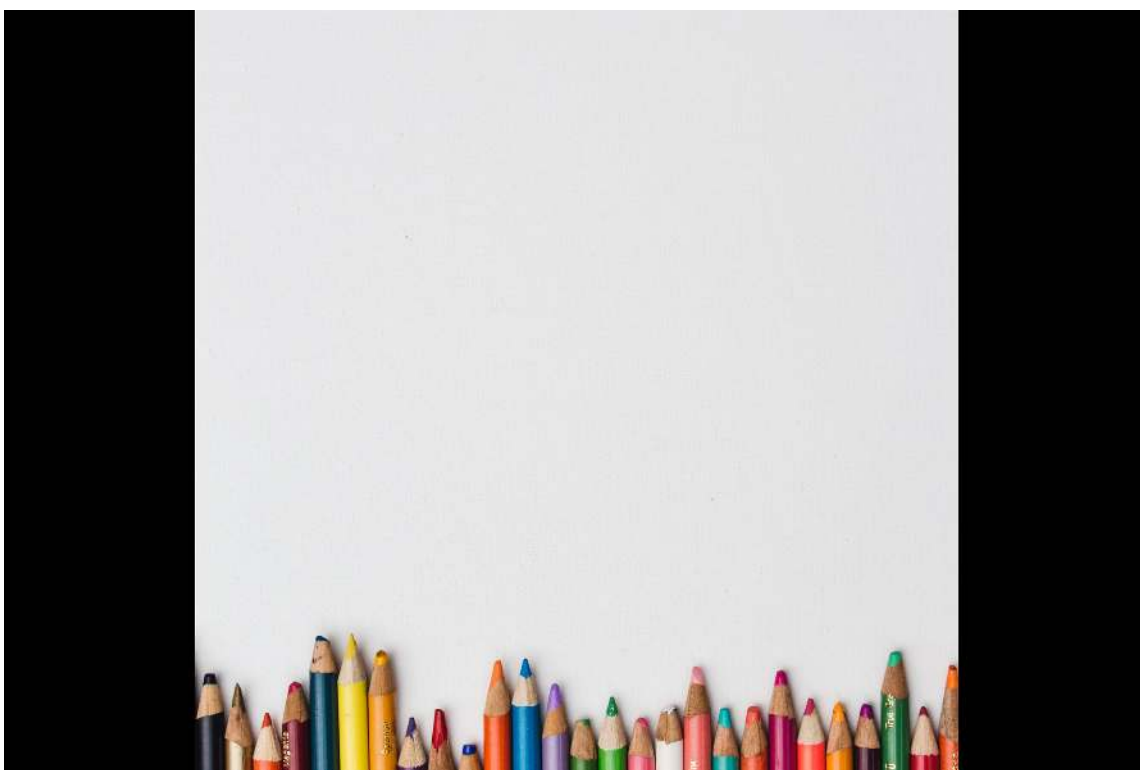
چرخش تصاویر

شما می‌توانید با تعیین مقدار درجه‌ای که باید چرخانده شود در متد `rotate()` تصاویر را چرخانید. ابعاد اصلی حفظ می‌شود و با رنگ سیاه "پر می‌شود". می‌توانید به صورت اختیاری پارامتر `expand` را به آن ارسال کنید تا تصویر چرخانده جدید را با ابعاد قدیمی پر کنید.

In [37]:

```
1 pencils.rotate(90)
```

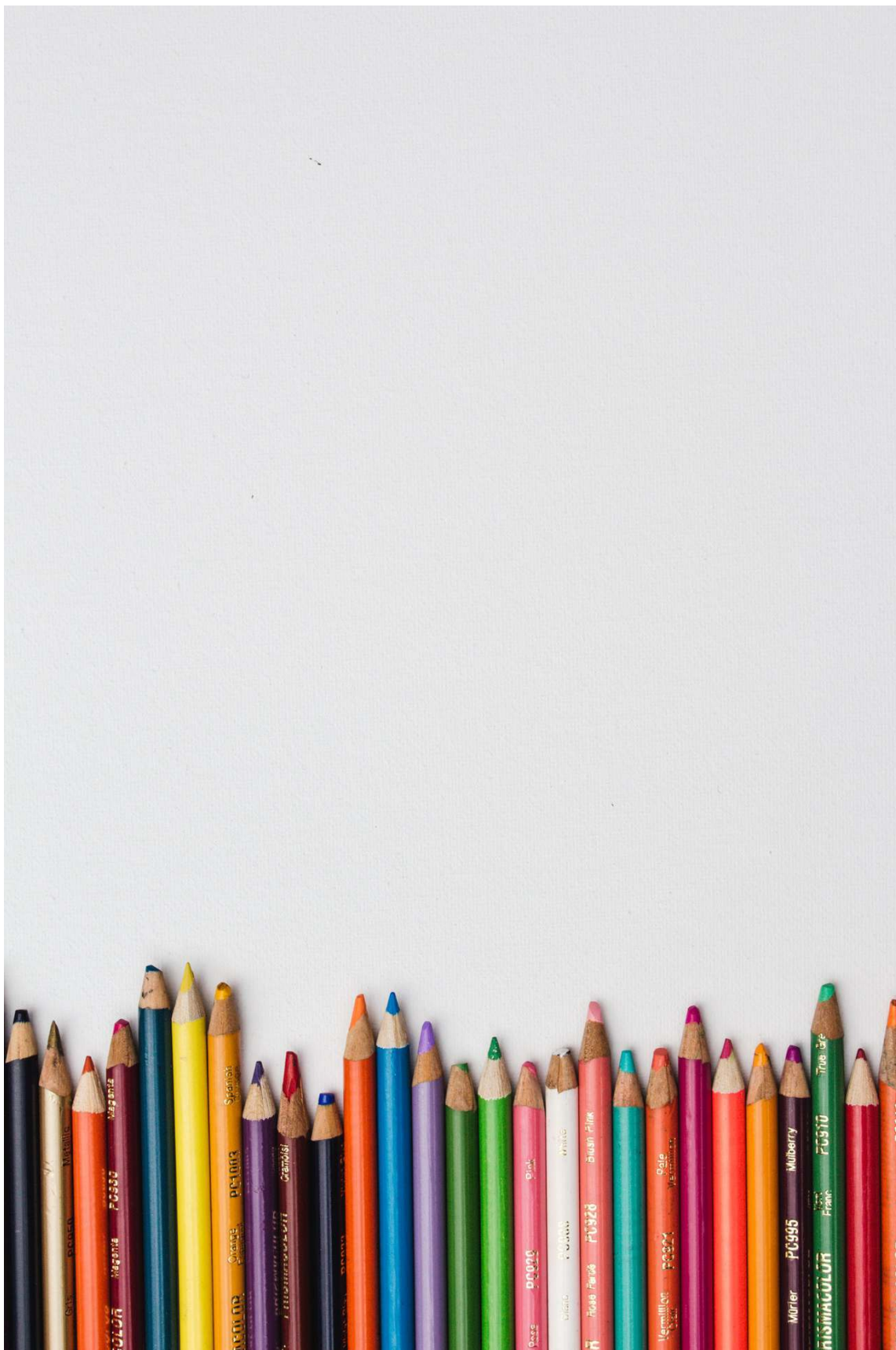
Out[37]:



In [38]:

```
1 pencils.rotate(90, expand=True)
```

Out[38]:

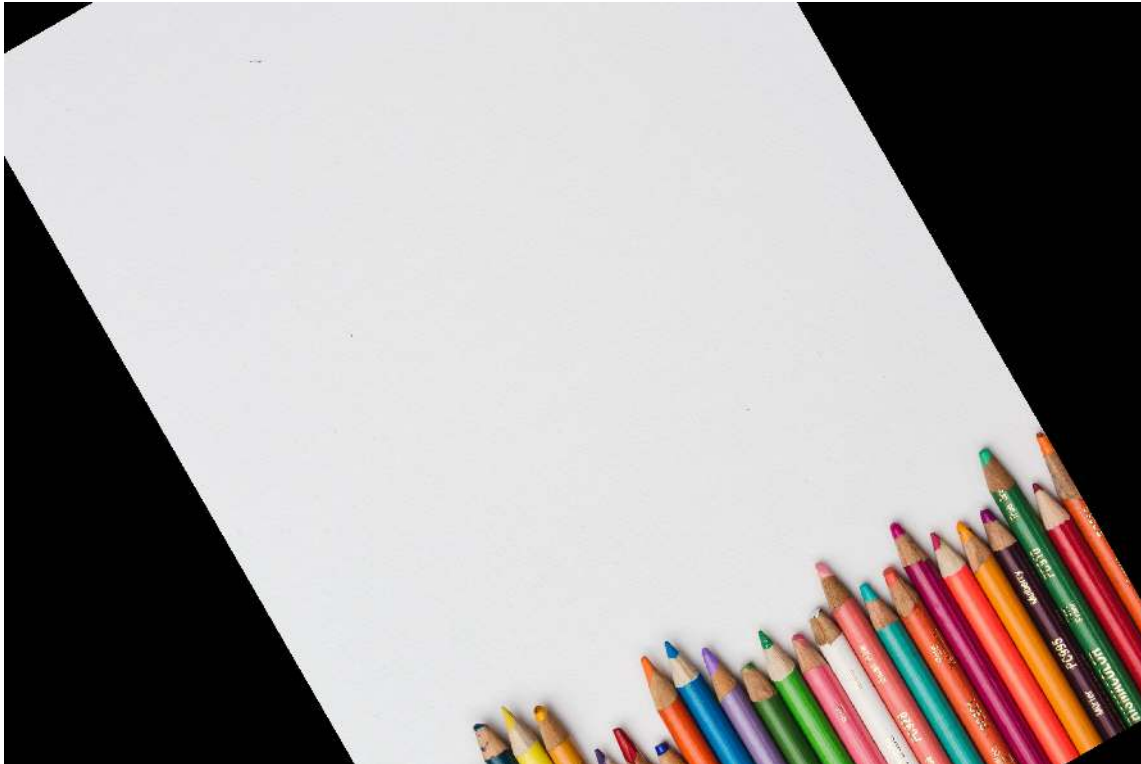


توجه کنید که در چرخش 120 درجه چه اتفاقی می افتد.

In [39]:

```
1 # The image is cut off  
2 pencils.rotate(120)
```

Out[39]:



In [40]:

```
1 pencils.rotate(120,expand=True)
```

Out[40]:



شفافیت

ما می‌توانیم یک مقدار آلفا (RED، Green، Blue، Alpha برای RGBA) اضافه کنیم که مقادیر آن می‌تواند از 0 تا 255 باشد. اگر آلفا 0 باشد، تصویر کاملاً شفاف است و اگر 255 باشد، کاملاً مات است.

شما می‌توانید رنگ خود را اینجا بسازید تا مقادیر ممکن را بررسی کنید: https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Colors/Color_picker_tool (https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Colors/Color_picker_tool)

ما می‌توانیم با استفاده از متد `putalpha()` مقادیر آلفا تصویر را تنظیم کنیم:

In [64]:

```
1 red = Image.open('red_color.jpg')
```

In [65]:

```
1 red
```

Out[65]:



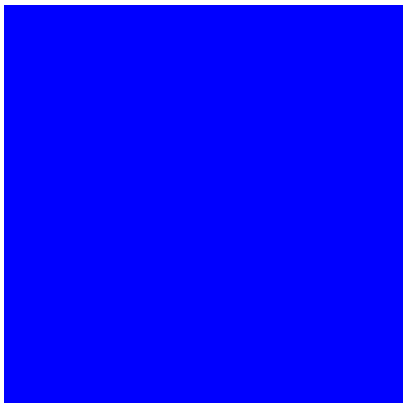
In [72]:

```
1 blue = Image.open('blue_color.png')
```

In [73]:

```
1 blue
```

Out[73]:



In [68]:

```
1 red.putalpha(150)
```

In [69]:

```
1 red
```

Out[69]:



In [70]:

```
1 blue.putalpha(128)
```

In []:

```
1 blue
```

In [76]:

```
1 blue.paste(red,box=(0,0),mask=red)
```

In [77]:

```
1 # Get back an image that is more purple.  
2 blue
```

Out[77]:



شفافیت و ماسک‌گذاری می‌تواند پیچیده‌تر از آنچه که در اینجا نشان داده شده است باشد، اگر به چیزی بیشتر نیاز داشته باشید، مستندات را بررسی کنید: <https://pillow.readthedocs.io/en/stable/>

ذخیره تصاویر

In [79]:

```
1 blue.save("purple.png")
```

بیا باید بررسی کنیم که آیا این کار کرد یا خیر:

In [80]:

```
1 purple = Image.open("purple.png")
```

In [81]:

```
1 purple
```

Out[81]:



عالی!