

الجلسة الأولى

تثبيت المكتبات المطلوبة

سنستخدم في مقرر العملي لغة البايثون Python وسنحتاج خلال عملنا في المقرر للعديد من المكتبات التي يجب تنزيلها حتى تعمل الأكواد بشكل صحيح.

لتثبيت Python نذهب للموقع التالي ونقوم بتثبيت النسخة المناسبة لنظام التشغيل (فرضاً Windows64)

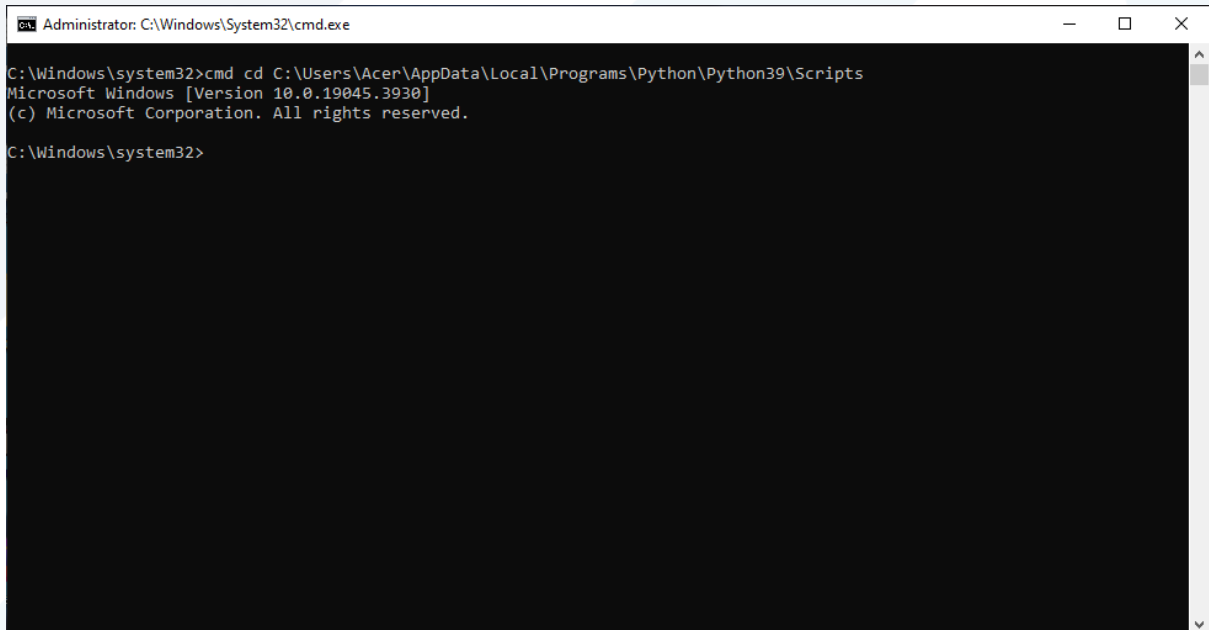
بعد التثبيت نقوم بإنشاء ملف نصي على سطح المكتب ثم نكتب فيه الأمر التالي:

```
cmd cd C:\Users\Acer\AppData\Local\Programs\Python\Python39\Scripts
```

يختلف هذا الأمر حسب مسار مجلد Scripts في الجهاز الذي تعمل عليه

بعد ذلك قم بحفظ الملف ثم حول امتداده إلى الامتداد bat

ثم قم بتشغيله بالنقر عليه بالزر اليمين ثم اختيار خيار run as administrator عليه فتظهر الشاشة الآتية:



نكتب الآن الأوامر التالية لتنصيب المكتبات اللازمة للعمل (قم بكتابة كل أمر على حدى ليتم تنصيب مكتبة تلو الأخرى)

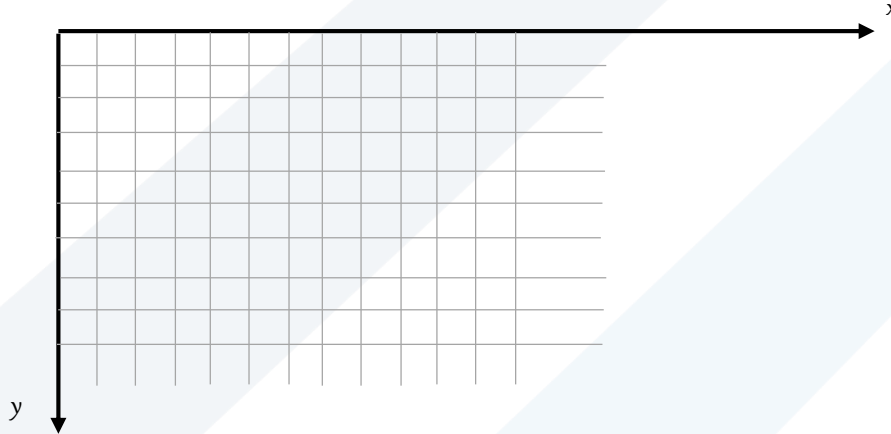
```
pip install numpy
```

```
pip install matplotlib
```

```
pip install opencv-python
```

مفاهيم أساسية عن الصورة الرقمية 2D Digital Image

الصورة هي عبارة عن مجموعة من القيم الرقمية ثنائية البعد لها إحداثيات أفقية وعمودية. العنصر الرئيسي هو Pixel. وتمثل الصورة بالشكل التالي:



يمثل تقاطع السطر مع العمود البكسل pixel، وترمز البكسلات وفقاً لإحداثياتها العمودية والأفقية، فمثلاً البكسل الأول هو (0,0) والبكسل الأخير هو (m-1,n-1) حيث أن m هو عدد الأسطر و n هو عدد الأعمدة. قد تمتلك الصورة الواحدة ثلاث مركبات وتسمى المركبات اللونية وتكون الصورة بهذه الحالة ملونة R,G,B.

1.1 حساب الحجم التخزيني للصورة ضمن الهارد

الحجم = عدد الأسطر * عدد الأعمدة * الحجم اللازم لتخزين البكسل الواحد.
مثال: لدينا صورة ملونة أبعادها 100*100 ذات دقة 4 بت لكل بكسل يكون حجمها التخزيني:
 $100 * 100 * 4 * 3 = 120000$ بت = 15000 بايت = 14.6 ميغا بايت.

دقة السويات الرمادية وأنواع الصور

دقة السويات الرمادية هي عدد البتات اللازمة لتمثيل الصورة ضمن الحاسب. وتعطى وفق العلاقة $L=2^k$ حيث k هي عدد البتات اللازمة لتخزين البكسل ضمن الذاكرة في حين أن L هي عدد السويات الرمادية والتي تبدأ بالصففر وتنتهي بـ $L-1$.

نوع الصورة	مجال السويات	عدد البتات
ثنائية	0 أو 1	بت واحد

رمادية	2 بت	[0-3] تحسب وفق القانون $[0-L] = [0-2^k - 1] = [0-2^2 - 1] = [0-3]$
رمادية	3 بت	[0-7]
رمادية	8 بت uint8	[0-255]
ملونة	8 بت ثلاث مركبات	[0-255]

التعامل مع الصورة في لغة Python باستخدام مكتبة opencv

1.2 قراءة الصورة

يتم قراءة الصورة باستخدام التابع imread. والذي يأخذ الشكل التالي:

`cv2.imread(path, flag)`

flag: تمثل الطريقة التي نريد قراءة الصورة بها، ولها ثلاثة أنماط:

1- `cv.IMREAD_COLOR`: تستخدم من أجل قراءة صورة ملونة، يمكن الاستعاضة عنها عن طريق كتابة الرقم 1، وهي القيمة الافتراضية.

2- `cv2.IMREAD_GRAYSCALE`: تستخدم من أجل قراءة صورة في وضع التدرج الرمادي. يمكن استخدام الرقم 0 بدلاً من كتابة العبارة.

3- `cv2.IMREAD_UNCHANGED`: تحميل الصورة مع معامل α . يمكن استخدام الرقم -1.

path: عبارة عن سلسلة محرفية تمثل مسار الصورة التي نريد قراءتها. وهنا يمكن أن نميز حالتين:

(a) في حال كانت الصورة ضمن المجلد الذي يحتوي على ملف python الذي نعمل عليه يكفي فقط أن نمرر اسم الصورة مع امتدادها كبارامتر للتابع (إضافة إلى البارامتر الثاني الذي يمثل Flag).

(b) في حال كانت الصورة ضمن مجلد آخر مختلف عن المجلد الذي يحتوي على ملف python الذي نعمل عليه عندها نقوم بتمرير المسار Directory الذي يحتوي على الصورة ونهي المسار باسم الصورة مع امتدادها.

مثال:

```
import cv2 as cv
```

مدرس المقرر: د. علي محمود ميا

```
I=cv.imread("rose.jpg")
```

في هذا المثال سوف يتم تحميل الصورة وقراءتها على أنها صورة ملونة لأن القيمة الافتراضية لـ Flag هي 1.

```
import cv2 as cv
```

```
I=cv.imread("rose.jpg",0)
```

يتم تحميل الصورة وقراءتها على أنها صورة رمادية.

1.3 عرض الصورة

يتم عرض الصورة باستخدام التابع imshow الذي يقوم بإظهار الصورة في نافذة، يأخذ هذا التابع بارامترين، البارامتر الأول عبارة عن سلسلة محرفية تمثل اسم النافذة التي سيتم عرض الصورة من خلالها أما البارامتر الثاني يمثل الصورة المراد عرضها.

```
import cv2 as cv
```

```
I=cv.imread("rose.jpg")
```

```
cv.imshow("imag",I)
```

```
cv2.waitKey()
```

في هذا المثال سوف يتم عرض الصورة rose.jpg ضمن نافذة اسمها image. التعليمة cv2.waitKey() تقوم بإيقاف تنفيذ الكود لمدة زمنية معين تقدر بالملي ثانية وتوضع بين قوسين بالشكل (). في حال كانت القيمة الممررة كبارامتر أصغر أو تساوي 0 عندها سوف يتم عرض الصورة وبعد عرضها سوف يتم إيقاف تنفيذ الكود إلى ما لا نهاية، ويتم استكمال تنفيذ الكود عند قيام المستخدم بالضغط على زر معين.

أما في حال كانت القيمة الممررة كبارامتر قيمة موجبة عندها سوف يقوم النظام باستكمال تنفيذ الكود عند انقضاء الفترة الزمنية المحددة كبارامتر (يجب تحديد القيمة بالملي ثانية مثلاً لو أدركنا الانتظار 5 ثواني نمرر وسيط 5000).

ملاحظة: إذا لم يتم استخدام التعليمة cv2.waitKey() لن يتم إظهار الصورة.

1.4 تخزين الصورة وحفظها باسم أخرى

يتم تخزين الصورة باستخدام التعليمة imwrite() حيث تأخذ بارامترين: البارامتر الأول يمثل اسم الصورة مع امتدادها أما البارامتر الثاني يمثل الصورة التي نريد حفظها. مثال:

```
Import cv2 as cv
```

```
I=cv.imread("rose.jpg")
```

```
cv.imwrite(I, R.png)
```

1.5 عرض أبعاد الصورة

لعرض أبعاد الصورة كما في المثال التالي:

```
import cv2 as cv
```

```
I=cv.imread("rose.jpg")
```

```
print(I.shape)
```

1.6 عرض نوع بيانات الصورة وأصغر قيمة وأعلى قيمة

لعرض نوع الصورة وأصغر قيمة فيها وأعلى قيمة فيها نستخدم التعليمة `np.iinfo()`، مثال:

```
import cv2
```

```
import numpy as np
```

```
I=cv2.imread(r"D:\rose.jpg")
```

```
np.iinfo(I.dtype)
```

1.7 إظهار الصورة كمصفوفة بنوع بيانات محدد

لإظهار الصورة على شكل مصفوفة نستخدم التعليمة `astype()` مع تحديد نوع البيانات الذي نريد أن يظهر به (بعد أن يتم قراءة الصورة) بالشكل التالي:

```
import cv2
```

```
I=cv2.imread(r"D:\rose.jpg")
```

```
I.astype(float)
```

ملاحظات:

لتحديد عدد البكسلات في الصورة (بعد أن يتم قراءة الصورة) نستخدم:

```
Image_name.size
```

لتحديد نوع بيانات الصورة (بعد أن يتم قراءة الصورة) نستخدم:

```
Image_name.dtype
```