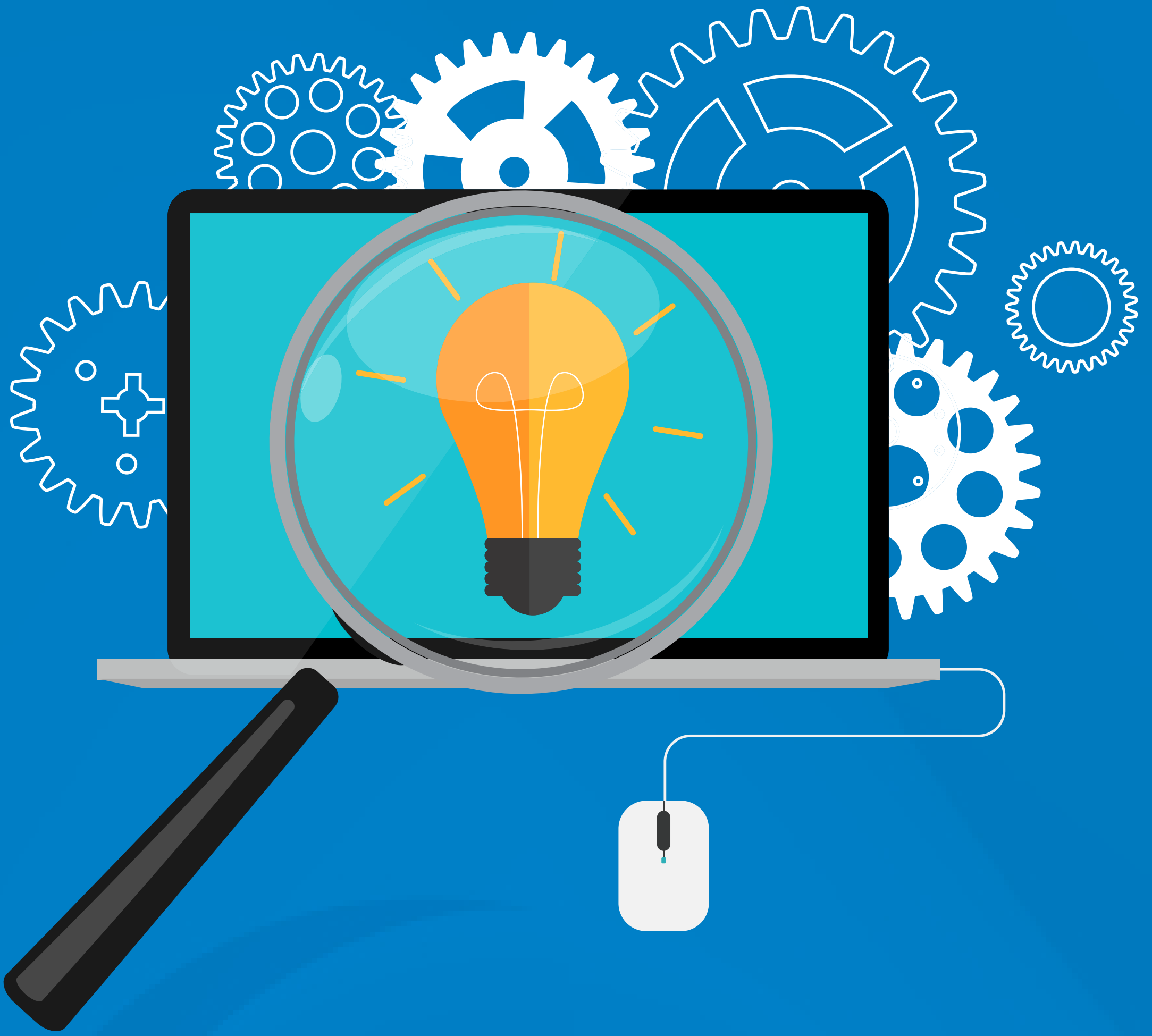


كتيب

# مقدمة لتطبيقات رؤية الحاسب

إعداد وتقديم مستشار الابتكار "منذر الرضي"



# المحتويات

4	عن مركز ذكاء وسياسة الاستخدام
5	عن الكتيب وماذا يقدم
6	مقدمة عن الكتيب
7	ما الرؤية الحاسوبية والذكاء الاصطناعي
8	أهمية الرؤية الحاسوبية والذكاء الاصطناعي في الحياة الحديثة
9	التعلم العميق مقابل رؤية الكمبيوتر التقليدية
10	تطبيقات الرؤية الحاسوبية والذكاء الاصطناعي
17	البرامج المستخدمة في الرؤية الآلية
23	العتاد المستخدم في الرؤية الآلية
31	لغة البرمجة بايثون (Python)
34	تنصيب البرمجيات
45	برنامج اليوم الأول
46	برنامج اليوم الثاني
47	قواعد كتابة الكود في بايثون (Python)
49	أمثلة
51	تمرين 1
53	تمرين 2
55	تمرين 3
57	Strings
60	Formatted text
61	Function vs method
64	Boolean expresion
65	Strings
66	العمليات الحسابية والمتغيرات
71	Mathematical module
75	If else Statement
83	Understand Pixel
84	Image of black and white
85	Gray scale image
86	Colored image
87	Colored VGA Image
89	Import image
90	To import video
91	Reading from a webcam

# المحتويات

96	.Basic Function (convert to Grayscale)
99	Basic Function (Blur)
100	Reference
101	Save to different format
102	shifting
103	blur
107	Image stack
108	Image perspective
109	Color Dtetection
115	Countor and shape
118	.Create function
119	.Copy img countour and draw countor
120	.Condition if to elemenate some countors + area and use length to count it
121	Git bounding box and draw it
123	.Ai

# مركز نكاء

**منشآت**  
monsha'at  
لهيئة العامة للمنشآت الصغيرة والمتوسطة  
Small & Medium Enterprises General Authority

## عن مركز نكاء

جاء إنشاء مركز "نكاء" - كأول مركز متخصص في التقنيات المتقدمة- لخدمة رواد الأعمال والمنشآت الصغيرة والمتوسطة في المملكة، حيث يهدف المركز إلى تمكين قطاع المنشآت الصغيرة والمتوسطة من توظيف التقنيات المتقدمة لتطوير هذه المنشآت، وزيادة تنافسيتها، ويكون حلقة ربط ما بين رواد الأعمال، وصناع القرار في مجالاته المتخصصة. يتخذ مركز "نكاء" لعلوم البيانات والذكاء الاصطناعي مدينة (الخُبر) مقراً له، ويقع مركز "نكاء" لإنترنت الأشياء والأمن السيبراني في مدينة (الرياض)، ويخدم المركز -بفرعيه- شتى أنحاء المملكة العربية السعودية.

بإمكانك النقر على الشعارات والروابط الموجودة في هذا الكتيب. للذهاب إلى المواقع الإلكترونية الخاصة بها.



## سياسة الاستخدام

إن المعلومات الواردة في هذا الكتيب تم تجميعها وتنسيقها بجهود موظفي مركز نكاء التابع لـ الهيئة العامة للمنشآت الصغيرة والمتوسطة "منشآت"، ولا ينبغي لقارئها أن يعمل بها دون مشورة مناسبة من المتخصصين.

للمزيد من المعلومات يرجى التواصل على البريد الإلكتروني [support@thakaa.sa](mailto:support@thakaa.sa)

جميع الحقوق محفوظة لمركز نكاء، أحد مراكز الابتكار التابعة للهيئة العامة للمنشآت الصغيرة والمتوسطة "منشآت".

## عن الكتيب

هذا الكتيب يتحدث عن تجربتنا في معسكر "نكاء" لتطبيقات رؤية الحاسب. يتناول المعسكر مفاهيم الرؤية الحاسوبية وتطبيقاتها، وكيفية استخدامها في حل المشاكل العملية والتطبيقات العملية. سنستكشف في هذا الكتيب التقنيات المستخدمة في مجال الرؤية الحاسوبية، والتطبيقات العملية في المجالات المختلفة، مثل: الطب والصناعة والأمن والترفيه وغيرها. يحتوي الكتيب أيضاً على تمارين عملية باستخدام لغة بايثون (Python)، ومكتبة أوبن سي قي (OpenCV).

## ماذا يقدم؟

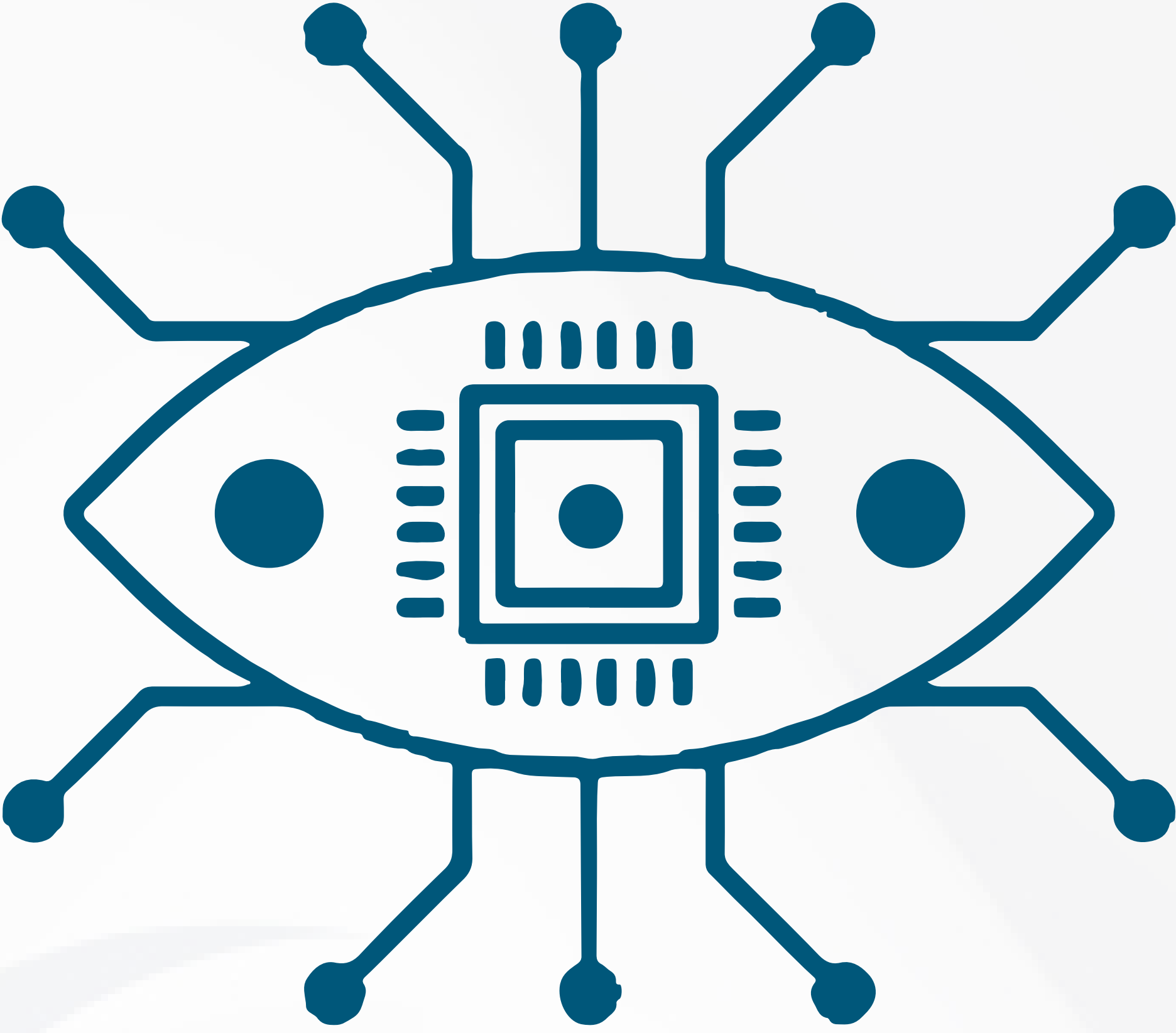
- عند الانتهاء من قراءة الكتيب، ستكون قد تعرفت على المفاهيم الأساسية للرؤية الحاسوبية، وستتمكن من استخدام لغة بايثون (Python) ومكتبة أوبن سي قي (OpenCV) لحل المشاكل العملية وتطبيق الرؤية الحاسوبية في معالجة الصور، وستعرف الألوان والأشكال الهندسية.
- نتمنى لك قراءةً ممتعةً، واكتساب معرفة قيمة مع هذا الكتيب.



## المقدمة

تعد الرؤية الحاسوبية وتطبيقات الذكاء الاصطناعي، من أهم المجالات التقنية الحديثة التي تتطور بشكل متسارع في العصر الحالي. فقد أصبح من الممكن للحواسيب والأنظمة الحاسوبية الحديثة التعرف على الصور والفيديوهات والإشارات الرقمية بشكل مماثل؛ لتعرف الإنسان على العالم المحيط به. وتعد الرؤية الحاسوبية إحدى فروع الذكاء الاصطناعي الذي يهتم بتصميم الخوارزميات والنماذج التي تمكن الحواسيب من التعرف على الأشياء وفهمها.

تتنوع تطبيقات الرؤية الحاسوبية والذكاء الاصطناعي، فهي تستخدم في العديد من المجالات، مثل: التصوير الفوتوغرافي، والتصوير الطبي، وتقنيات الأمن والمراقبة، والتعرف على الوجوه والتعرف على النصوص والكتابة اليدوية، والتعرف على الأصوات، والتعرف على الحركات البشرية والتحكم في الروبوتات، وغيرها من التطبيقات الأخرى التي يمكن استخدامها في مجالات مختلفة.



### الرؤية الحاسوبية:

هي فرع من فروع علوم الحاسوب، والذكاء الاصطناعي، يتعلق بتطوير الأنظمة والبرامج التي تستطيع التعرف على الصور والفيديوهات ومعالجتها بشكل آلي، ويتضمن هذا المجال تقنيات لتحليل الصور واستخراج المعلومات الحاسوبية منها، مثل: التعرف على الأشكال والألوان والنماذج الهندسية، وكذلك تحليل الحركة والتتبع الحركي، والتعرف على الأشخاص والكائنات.

### الذكاء الاصطناعي:

يمكن تعريف الذكاء الاصطناعي بأنه: قدرة الأنظمة أو الأجهزة الحاسوبية على محاكاة الذكاء البشري، لإيجاد الحلول للمشاكل واتخاذ القرارات، وهو أحد فروع علوم الحاسوب، ويتعلق بتطوير الأنظمة والبرامج التي تتمتع بالقدرة على التعلّم والتكيّف واتخاذ القرارات بشكل مستقل، وذلك بمساعدة الخوارزميات والتقنيات المتقدّمة، ويُستخدم الذكاء الاصطناعي في العديد من التطبيقات مثل: الروبوتات، والأتمتة الصناعية، وتحليل البيانات، والتعلم الآلي، والترجمة الآلية، وغيرها.

تتداخل الرؤية الحاسوبية والذكاء الاصطناعي في العديد من التطبيقات، مثل: تطوير أنظمة التعرف على الصور، وتحليل البيانات الضخمة، والتعرف على الكائنات وغيرها. يمكن استخدام الذكاء الاصطناعي في تعزيز الرؤية الحاسوبية، بحيث يتم تحليل البيانات الحاسوبية بشكل أكثر تعقيداً، واتخاذ القرارات بشكل أكثر ذكاءً، وتفوقاً على البشر.

## أهمية الرؤية الحاسوبية والذكاء الاصطناعي في الحياة الحديثة

تشهد الحياة الحديثة اعتماداً متزايداً على التقنيات الحديثة التي تستخدم الرؤية الحاسوبية والذكاء الاصطناعي، والتي يمكنها إحداث تحول جذري في العديد من المجالات، كما يمكنها توفير حلول مبتكرة، وتحسين التقنيات الصناعية؛ حيث تمكّنت هذه التقنيات من تحويل العمليات البطيئة إلى عملياتٍ أسرع بكثير، وتخفيض التكاليف بشكلٍ كبير، وأتمتة العمليات الصعبة لتبدو أسهل مما كانت عليه، وتحسين المنتجات ذات الجودة القليلة، لتكون أكثر ملاءمةً وراحةً في الاستخدام وأكثر تطوراً، ومن هنا فإنّ استخدامات الرؤية الحاسوبية والذكاء الاصطناعي تزداد أهميةً في توفير حلول مبتكرة للعديد من المشاكل التي تواجهها المجتمعات الحديثة، فعلى سبيل المثال: تُستخدم تقنيات الرؤية الحاسوبية، والذكاء الاصطناعي في العديد من المجالات، مثل:

### الصناعة:

حيث يمكن استخدام الروبوتات الذكية؛ لتحسين الإنتاجية، وخفض التكاليف في الصناعة.



### الطب:

حيث يمكن استخدام تقنيات الرؤية الحاسوبية؛ لتحسين تشخيص الأمراض، وتحديد العلاجات الفعالة.



### التعليم:

حيث يمكن استخدام الذكاء الاصطناعي؛ لتحسين تجربة التعلم، وتوفير محتوى تعليمي مخصص لكل طالب.



### الأمن:

حيث يمكن استخدام تقنيات الرؤية الحاسوبية، والذكاء الاصطناعي؛ لتحسين الأمن في المطارات والمحطات، والمناطق العامة.



### النقل:

حيث يمكن استخدام الذكاء الاصطناعي؛ لتحسين السلامة والكفاءة في النقل العام.



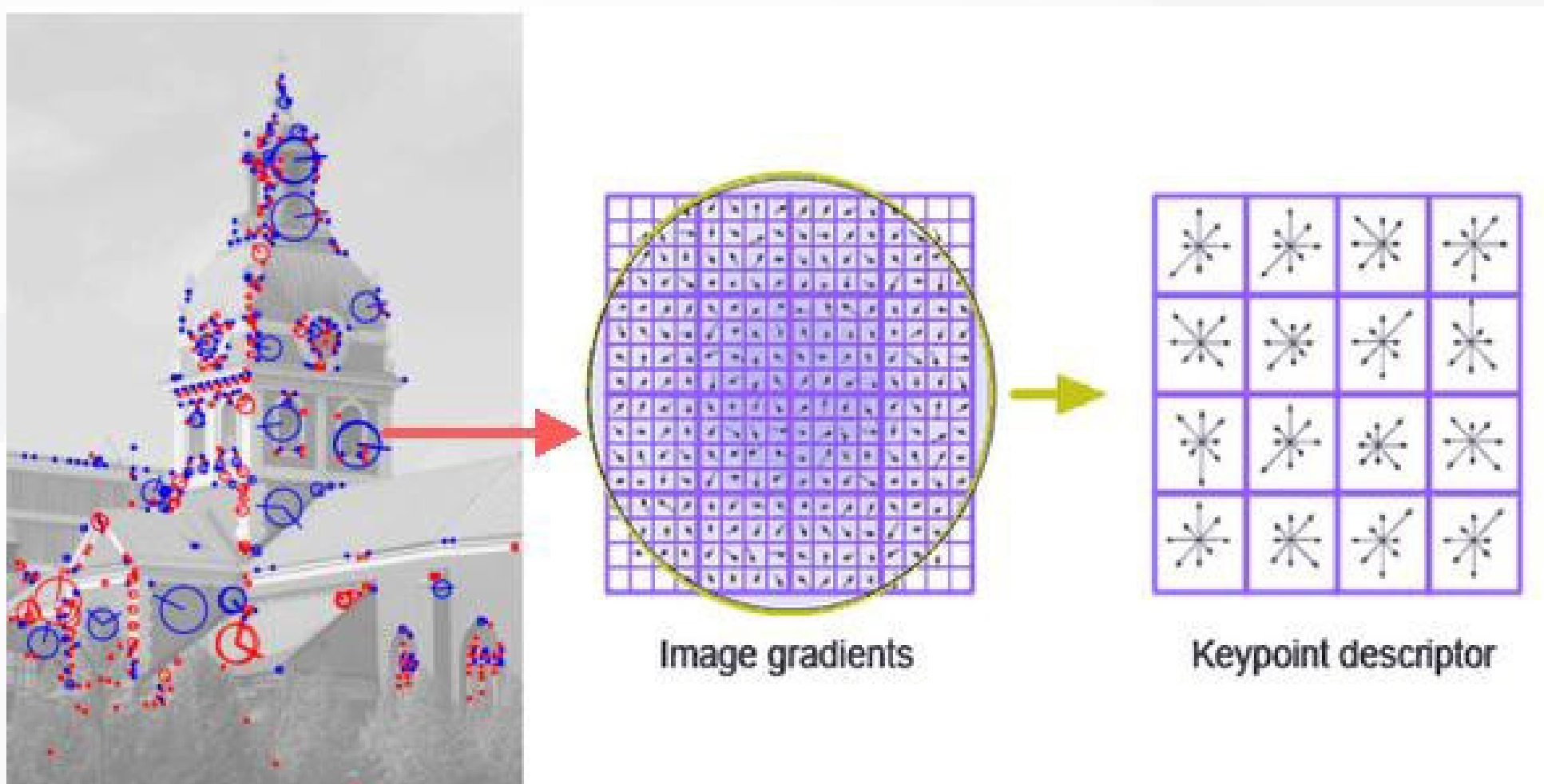


## التعلم العميق مقابل رؤية الكمبيوتر التقليدية

تتضمن تطبيقات الرؤية الحاسوبية والذكاء الاصطناعي تقنياتٍ مختلفة للتعرف على الصور والفيديو والمعلومات البصرية، ومن بين هذه التقنيات ما يسمى بـ «التعلم العميق»، يتميز التعلم العميق باعتماده على شبكات عصبونية اصطناعية، تقوم بمعالجة البيانات البصرية بطريقة شبه أوتوماتيكية، بينما تعتمد رؤية الكمبيوتر التقليدية على خوارزميات تُستخدم مجموعةً محددةً من المعالجات الرياضية، والمرشحات لاستخراج المعلومات البصرية.

يتميز التعلم العميق -كذلك- بقدرته على تحليل البيانات البصرية وفهمها بطريقة أكثر دقةً وفاعليةً وهذا ما يجعلها تقنيةً مهمةً في مجالات مختلفة، مثل: التصوير الطبي، والأمن، والصناعة، والروبوتات، والتجارة الإلكترونية، والترفيه، وغيرها.

يتطلب التعلّم العميق مجموعةً كبيرةً من البيانات المدرّبة، والقوة الحاسوبية الهائلة، ل يتيح للحواسيب تعلّم الأنماط والخصائص الأساسية في الصور والفيديو بشكل مشابه للطريقة التي يتعلم بها الإنسان، ومن المتوقع -مع تطور التكنولوجيا- أن يستمر التعلّم العميق في تحسين قدرات الحواسيب في مجالات التصوير البصري والذكاء الاصطناعي.



## تطبيقات الرؤية الحاسوبية والذكاء الاصطناعي

تطورت تقنية الرؤية الحاسوبية والذكاء الاصطناعي بشكلٍ كبيرٍ خلال السنوات الأخيرة، وأصبحت موضوع اهتمام كبير في العديد من المجالات، حيث تتيح تطبيقات الرؤية الحاسوبية القدرة على تحليل الصور والفيديوهات، وفهمها بطريقة آلية، مما يمكنها من إنتاج معلومات دقيقة ومفيدة، بالإضافة إلى ذلك، يتيح الذكاء الاصطناعي إنتاج نماذج تعلّم آلي تُستخدم في الكثير من التطبيقات، وإليك بعض الأمثلة:

### 1. المركبات ذاتية القيادة

يستخدم العديد من مصنعي السيارات وشركات تقنية السيارات الرؤية الحاسوبية في تطوير تقنية السيارات ذاتية القيادة وتحسينها، تتمثل فكرة السيارات ذاتية القيادة في استخدام تقنيات الحوسبة والاتصالات؛ لتمكين السيارات من القيادة دون تدخل من السائق، وتتطلب هذه التقنية الاعتماد على الرؤية الحاسوبية لتحليل المعلومات التي تُجمع من مجموعة متنوعة من الحساسات المثبتة في المركبة.

تساعد الرؤية الحاسوبية في تحديد موقع السيارة بدقة، وتتبع الحركة بشكل دقيق، كما يمكن استخدامها لتحليل الصور والمعلومات التي تُجمع من الكاميرات المثبتة في المركبة، وتقنيات الرادار والليدار، والأشعة تحت الحمراء؛ لمساعدة السيارة على تحليل ما يحدث حولها، واتخاذ القرارات المناسبة بشأن الحركة، والتحكم في السرعة، والتوقف، وتغيير المسار.

وتُوظف تقنيات الذكاء الاصطناعي والتعلم العميق في هذا النوع من التطبيقات؛ حيث تُستخدم الشبكات العصبية الاصطناعية للكشف عن الأشياء والمعالم على الطريق، وتصنيفها، وتحليلها.

ويعد استخدام الرؤية الحاسوبية في المركبات ذاتية القيادة، من أكثر التطبيقات إثارةً للاهتمام والتحدي في مجال الذكاء الاصطناعي والروبوتيات.

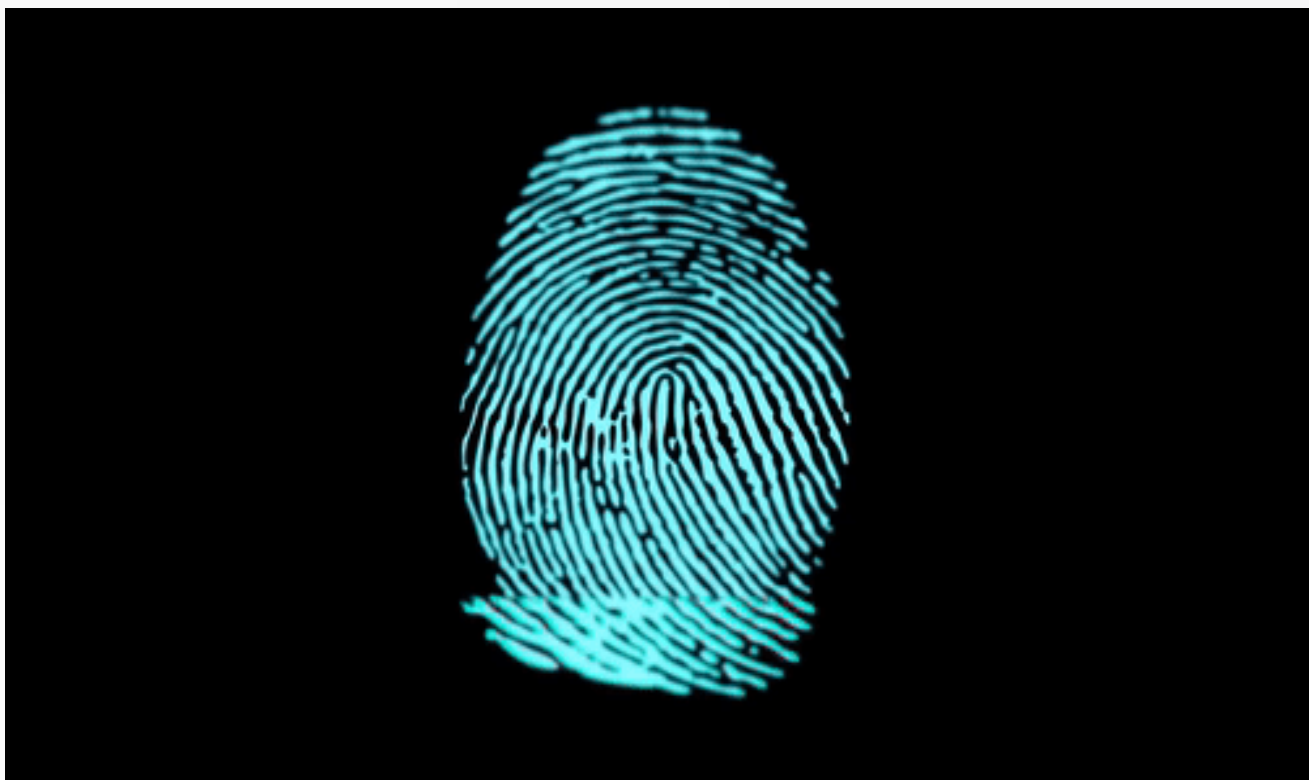




### 2. تقنية التعرف على الأشخاص

تعتبر تقنية التعرف على الأشخاص باستخدام الرؤية الحاسوبية من أهم التطبيقات وأكثرها شيوعاً في مجال الذكاء الاصطناعي؛ فهي تتيح للأنظمة الحاسوبية التعرف على معالم أوجه الأشخاص، والملامح الفريدة لكل شخص، وتمكّنها من تحديد هويتهم بدقة عالية، وتُستخدم هذه التقنية في العديد من المجالات كالأمن والسلامة، حيث يمكن استخدامها للتعرف على أشخاص محدّدين والتحقق من هويتهم، كما كما تحسّن تجربة المستخدم في التسوق الإلكتروني، حيث تتمكن الأنظمة الحاسوبية من التعرف على المستخدمين، وتوفير تجربة مخصصة لكل واحد منهم.

ويعتمد التعرف على الأشخاص بواسطة الرؤية الحاسوبية على تحليل البيانات المستخرجة من الصور والفيديوهات التي تتضمن وجوه الأشخاص، ويتم ذلك باستخدام تقنيات التعلم الآلي وتحليل الصور. ويمكن تحسين دقة التعرف على الأشخاص باستخدام الشبكات العصبية العميقة، والتي تمتلك القدرة على تعلم الملامح الفريدة لوجوه الأشخاص، وتحديد هويتهم بدقة عالية.





### 3. الطيار الآلي

يستخدم الطيار الآلي -أو ما يعرف بـ: «الطائرات بدون طيار»- تقنيات الرؤية الحاسوبية لأداء المهام بشكل آمن وفعال، ومن أهم استخدامات الرؤية الحاسوبية في الطيار الآلي: التحكم في مسار الطائرة وتجنب الاصطدامات، وذلك من خلال استخدام الكاميرات والحساسات الأخرى، للكشف عن العوائق المحتملة، وتحديد موقع الطائرة بدقة عالية.

تُستخدم الرؤية الحاسوبية في الطيار الآلي -كذلك- لمساعدة الطائرة على الهبوط بسلام؛ حيث تُستخدم كاميرات الرؤية الحاسوبية لتحديد مدى انحدار المدرج، وموقع الطائرة بالنسبة للمدرج، كما يمكن استخدام تقنيات الرؤية الحاسوبية لمراقبة الحالة الصحية للمحركات، وأجزاء الطائرة الأخرى، وكذلك للتحكم في نظام الطيار الآلي (Autopilot system) الذي يمكنه تشغيل الطائرة بشكل تلقائي دون تدخل الطيار البشري.

يعدُّ استخدام الرؤية الحاسوبية في الطيار الآلي أمراً حاسماً لنجاح تلك التقنية، وضمان سلامة الرحلات الجوية، ومن المتوقع أن يزداد الاعتماد عليها بشكل أكبر في المستقبل.

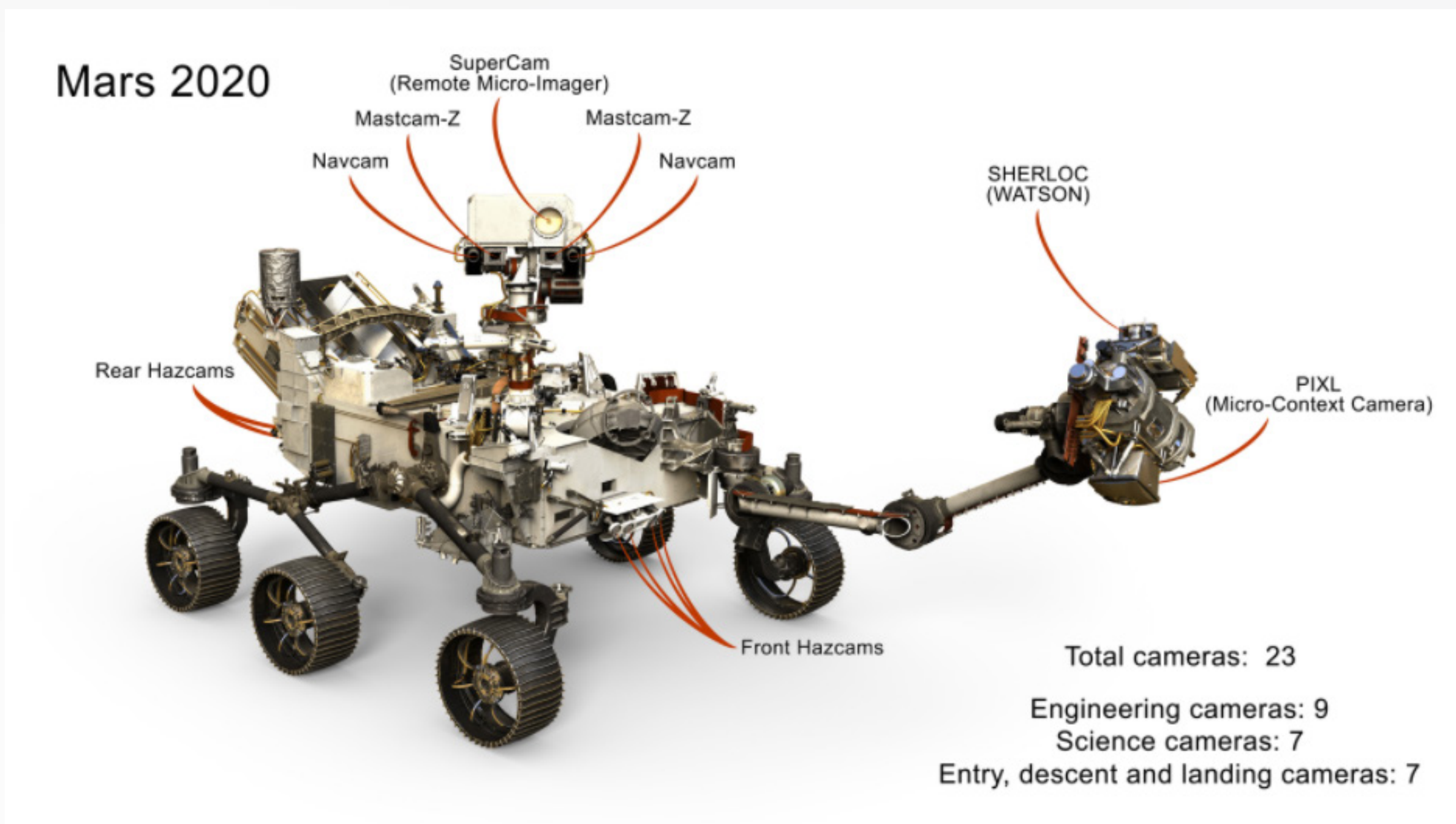




### 4. استكشاف الفضاء

تُستخدم التقنيات المتقدمة للرؤية الحاسوبية في الكشف والاستكشاف في الفضاء، حيث تلعب دوراً مهماً في تحسين الروبوتات الفضائية، وأدائها بشكل أفضل وأكثر كفاءة، ويتضمن ذلك استخدام الرؤية الحاسوبية في العديد من المهام المهمة، مثل: تحديد مواقع الأجسام السماوية والمناطق الفلكية، والتحكم في المركبات الفضائية الآلية، وتشخيص الأعطال وإصلاحها، وتصوير الكواكب والمجرات والكويكبات.

يستخدم الكمبيوتر المدمج في الروبوتات الفضائية والمركبات الفضائية؛ لتحليل الصور الملتقطة من الأجسام السماوية، باستخدام تقنيات الرؤية الحاسوبية المتطورة. على سبيل المثال: يمكن للروبوتات الفضائية استخدام الرؤية الحاسوبية لتحديد المناطق الخطرة وتجنبها، وتحديد مواقع الكويكبات وتحريكها بشكل آمن، وتصوير المناطق الفلكية، والكشف عن الأشياء التي لا يمكن تحديدها بالعين المجردة. بالإضافة إلى ذلك، يمكن إرسال هذه الروبوتات إلى الكواكب البعيدة التي يصعب على الإنسان الوصول إليها، لتقوم بعملية مسح للكوكب، وجمع بعض العينات، والتعرف عليها وتصنيفها.



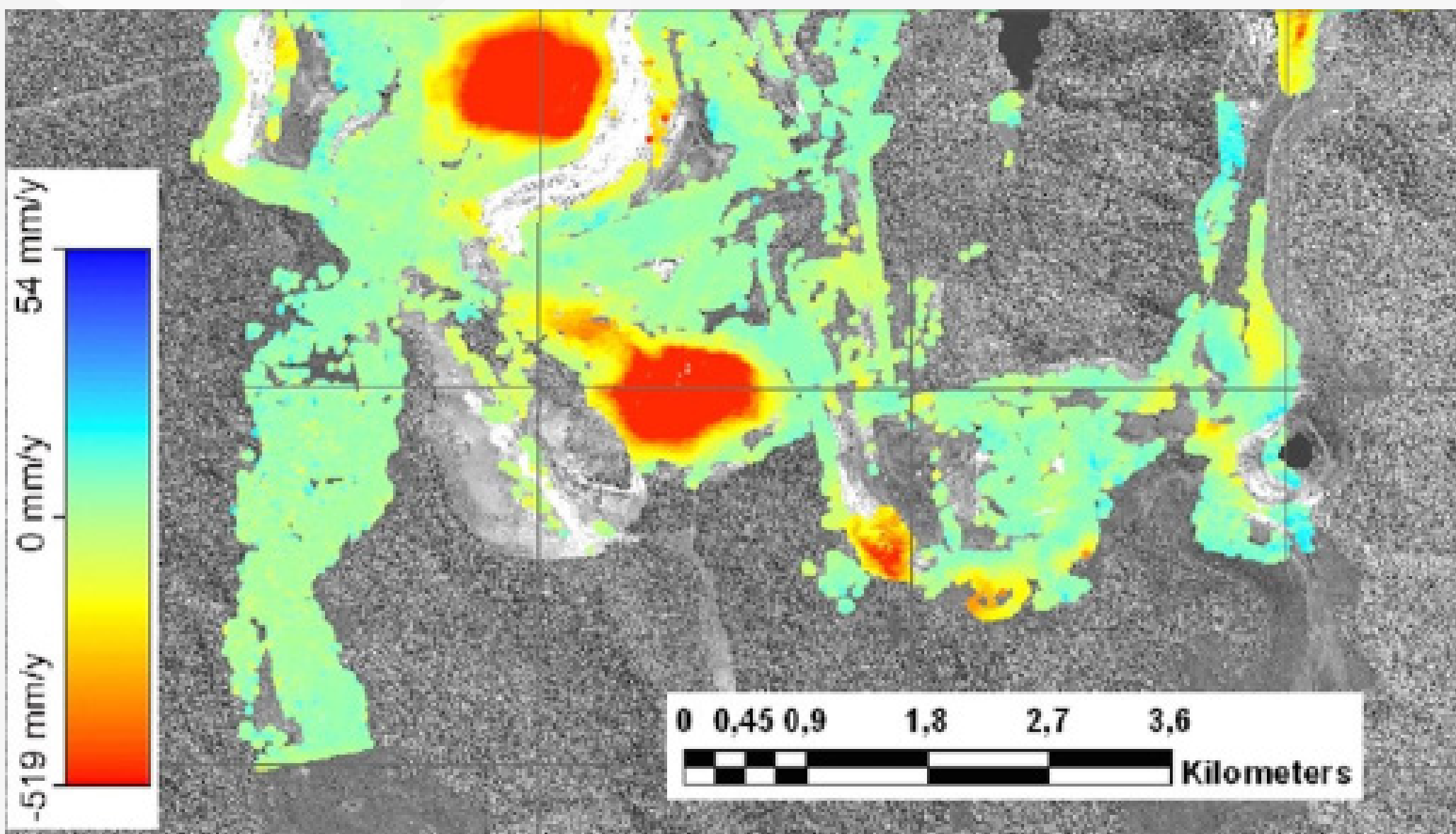


### 5. التنقيب عن النفط والغاز

تُستخدم تقنيات الرؤية الحاسوبية في صناعة التنقيب عن النفط والغاز بشكلٍ واسع، حيث تُستخدم للكشف عن مكامن النفط والغاز تحت سطح الأرض، وفي المحيطات، وتُعتمد هذه التقنيات على الصور الملتقطة من الأقمار الصناعية، والمروحيات والطائرات المسيّرة وغيرها، حيث يتم تحليل هذه الصور باستخدام خوارزميات معقدة لاستخراج المعلومات الضرورية.

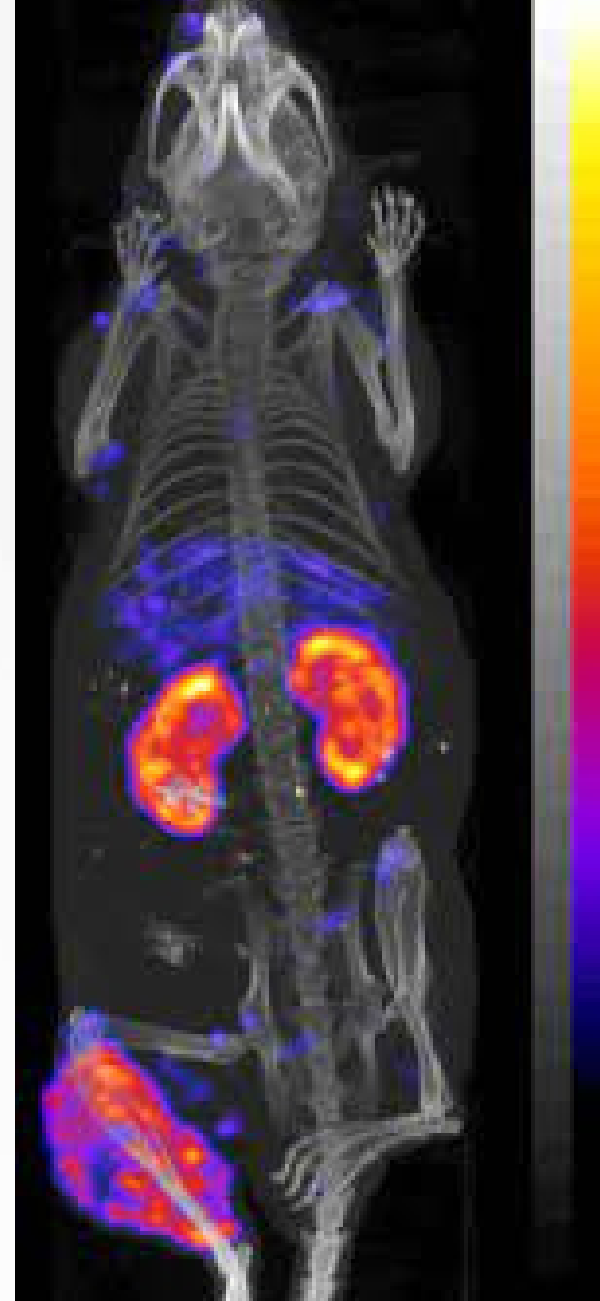
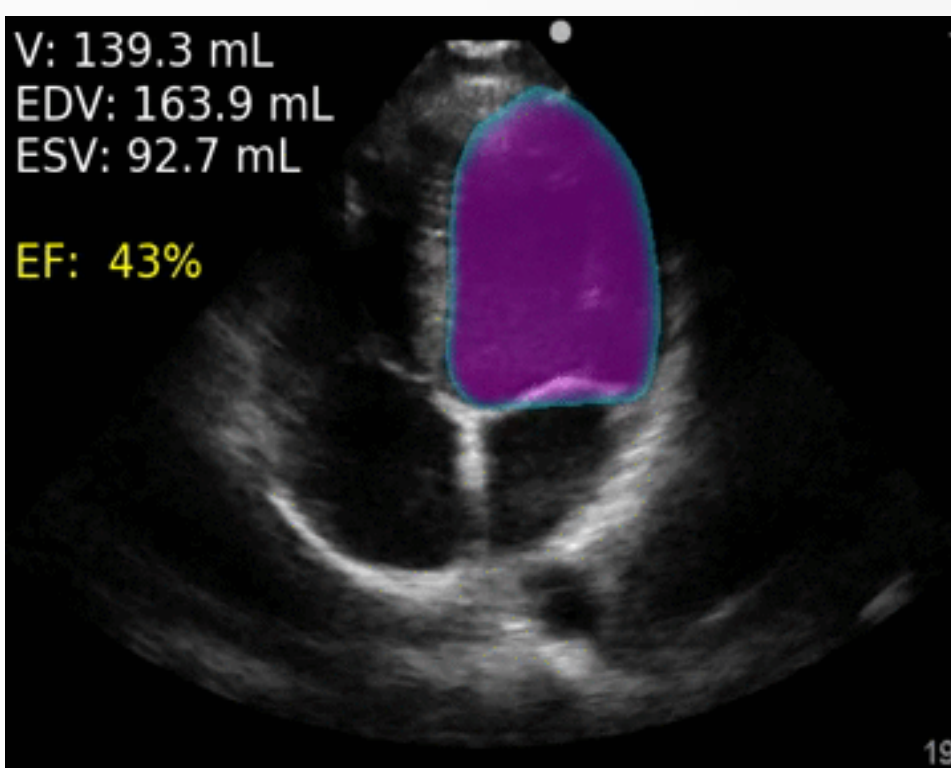
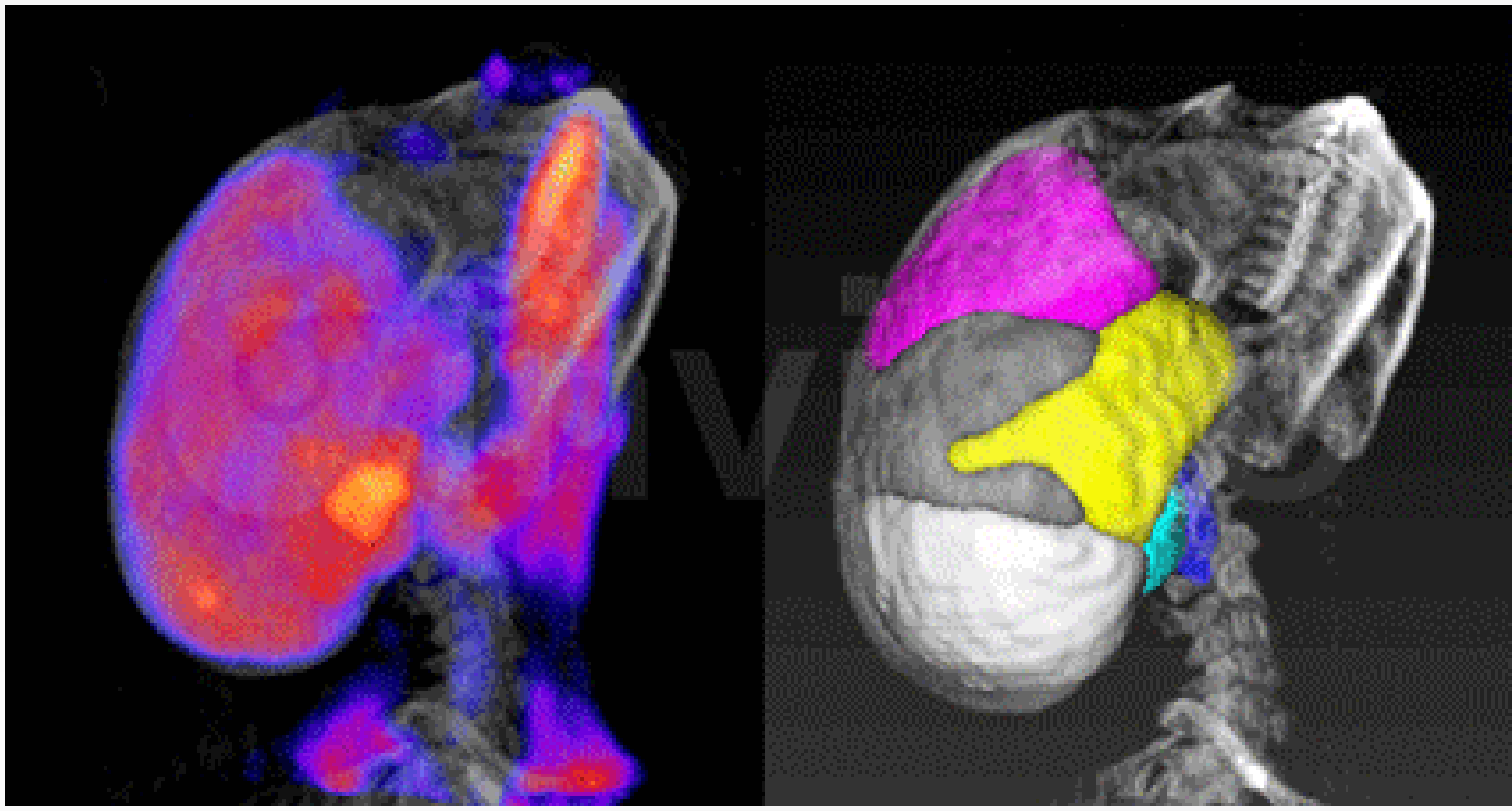
تُستخدم التقنيات الحديثة في الرؤية الحاسوبية لتحديد أماكن الحفر الأكثر احتمالاً، والأماكن التي تحتوي على النفط والغاز بشكلٍ أسرع وأكثر دقةً، وذلك بفضل استخدام تقنيات التعلم الآلي، والشبكات العصبية العميقة، كما يتم استخدام تقنيات الرؤية الحاسوبية لمراقبة عمليات الحفر والإنتاج والتخزين، حيث يتم استخدام الكاميرات، وأجهزة الاستشعار المختلفة؛ لرصد العمليات بشكلٍ دقيق ومستمر.

ويعدّ استخدام تقنيات الرؤية الحاسوبية في صناعة التنقيب عن النفط والغاز من العوامل المهمة لتحقيق أعلى مستويات الإنتاجية والربحية، وذلك بتوفير المعلومات الدقيقة والموثوقة حول أماكن النفط والغاز، ومراقبة عمليات الإنتاج بشكلٍ دقيقٍ وفعال.



### 6. المجالات الطبية

تُستخدم الرؤية الحاسوبية في المجالات الطبية لتحسين التشخيص وتحديد العلاجات المناسبة؛ حيث يمكنها استخدام التصوير الطبي لتوليد صور ثلاثية الأبعاد للأعضاء والأنسجة داخل الجسم، ويمكن للرؤية الحاسوبية تحليل هذه الصور للكشف عن الأمراض والإصابات، وتحديد العلاجات المناسبة، كما يمكن استخدام الرؤية الحاسوبية لمراقبة عمليات الجراحة، وتحديد الأجزاء المراد استئصالها أو علاجها بدقة عالية، ودون إحداث ضرر غير مقصود، ويمكن -كذلك- استخدامها في تحليل الصور السريرية، وتحديد العلامات المرضية، والتغيرات التشريحية التي يصعب كشفها بالعين المجردة، ويمكن استخدامها في مجال الأبحاث الطبية؛ لتحليل الصور الطبية الكبيرة، وتحديد العلاقات بين العوامل المختلفة المؤثرة في الأمراض والعلاجات.



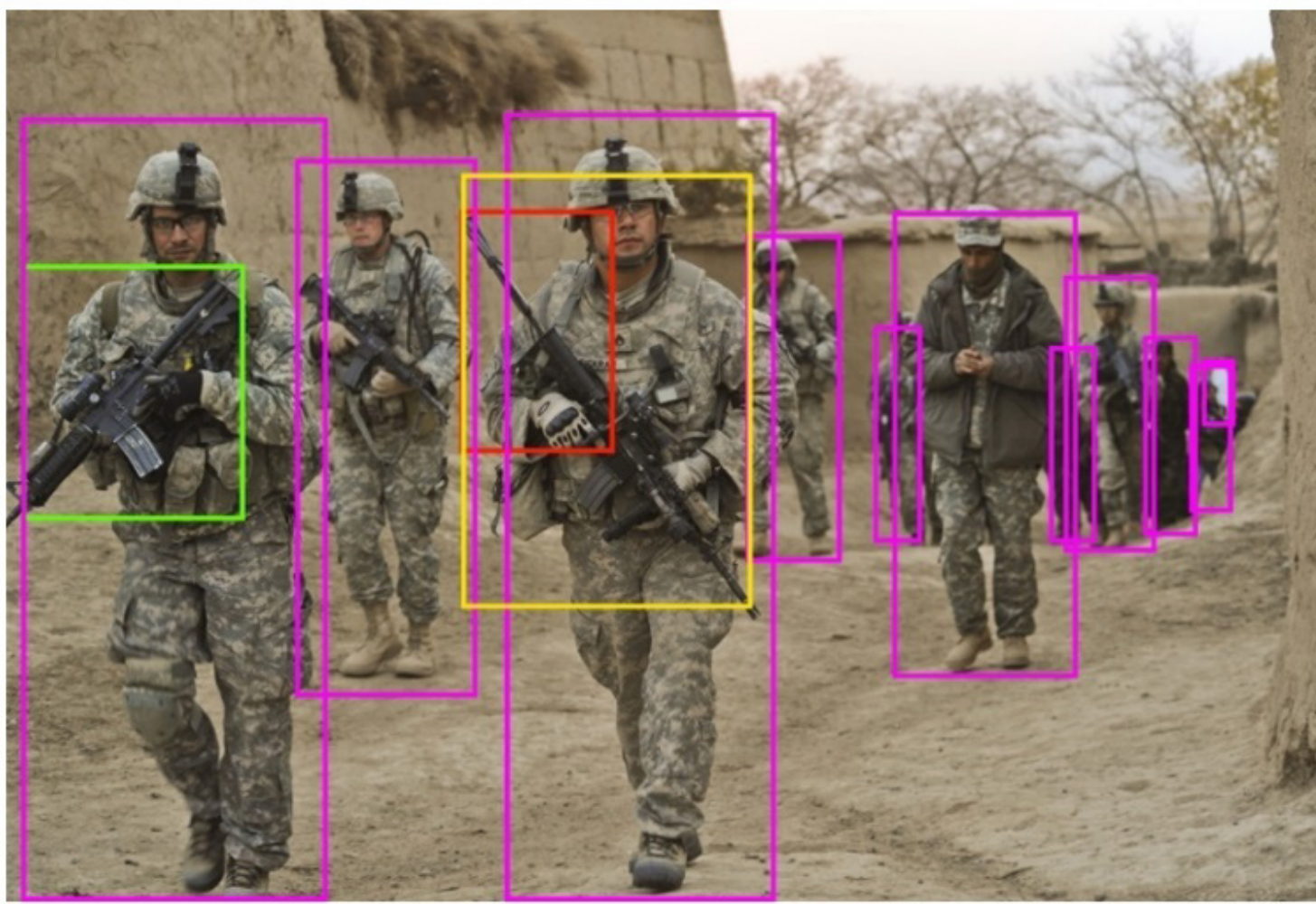


### 7. المجالات العسكرية

تُستخدم الرؤية الحاسوبية في المجالات العسكرية لتحسين التشخيص، والتعرّف على المعلومات بسرعة ودقة أفضل. وتشمل التطبيقات العسكرية للرؤية الحاسوبية: تحديد الأهداف ومتابعتها، والتعرّف على الوجوه والمعالم، والتصوير الجوي، والملاحة، والتحكم في الطائرات بدون طيار.

في العمليات العسكرية: تُستخدم الرؤية الحاسوبية لتحليل الصور من الأقمار الصناعية، والتعرّف على المواقع الجغرافية والأهداف المحتملة، ويمكن استخدام هذه التقنية لتصفية الصور للتعرف على المعالم المفيدة وتحليلها بدقة أكبر، كما يمكن استخدامها للكشف عن الأهداف العدائية، مثل: الدبابات، والمركبات المسلحة، وتحديد مساراتها، والتحكم فيها.

وفيما يتعلق بالطائرات بدون طيار: تُستخدم الرؤية الحاسوبية لتمكين الطائرات من الطيران بشكل آمن وفعال، بما في ذلك: التّحكم في ارتفاع الطائرة، والتعرّف على الأهداف والمعالم الأرضية، وتحديد المواقع، والتّحكم في التوجيه، كما يمكن استخدام التقنية لتحليل البيانات الاستخباراتية، والمعلومات المتعلقة بالمواقع العسكرية.

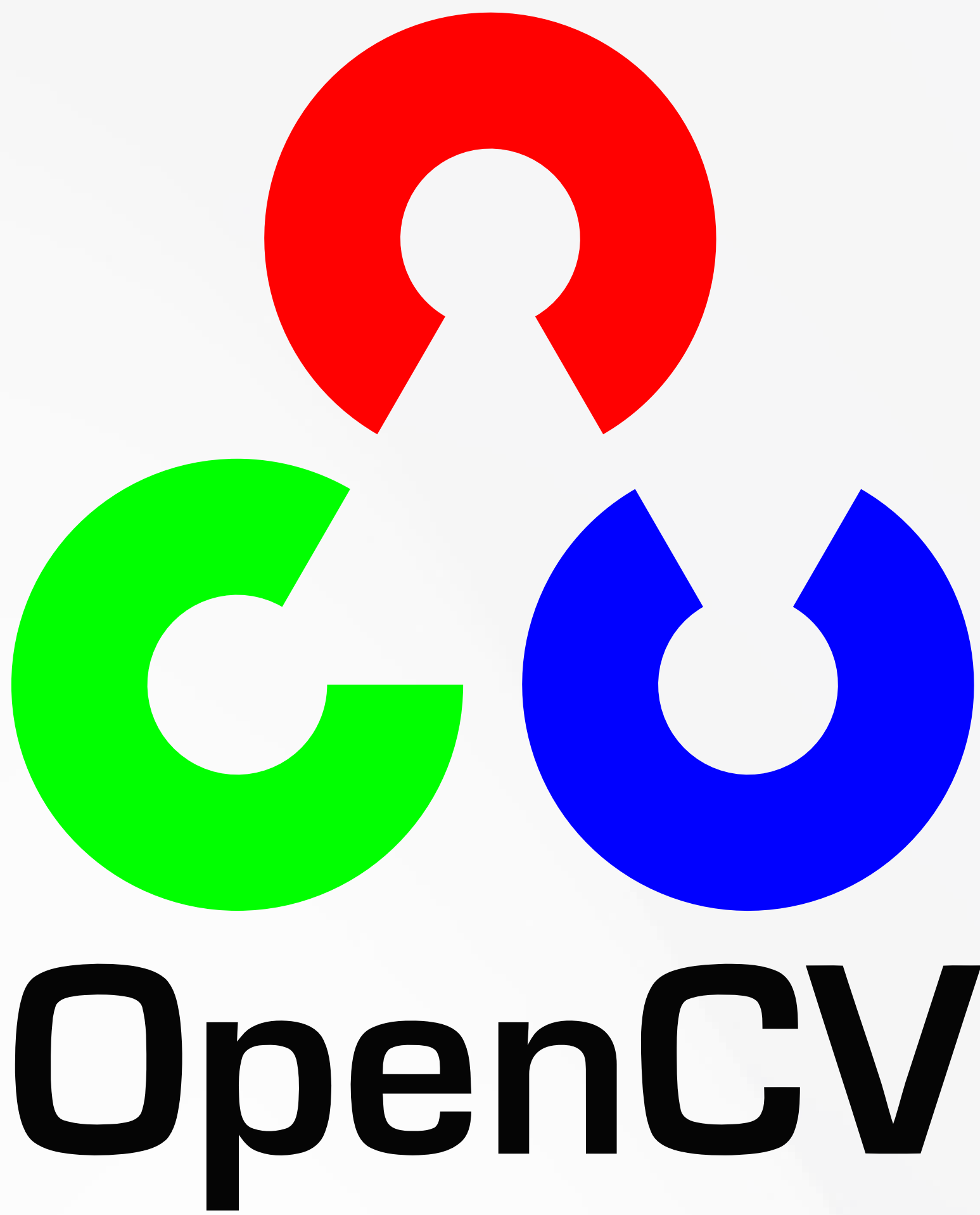




### 1. مكتبة أوبن سي في (OpenCV)

مكتبة أوبن سي في (OpenCV): هي مكتبة برمجية مفتوحة المصدر، تُستخدم لتطوير تطبيقات رؤية الحاسوب ومعالجة الصور والفيديو، تم تطويرها بلغة البرمجة (C++) وتدعم العديد من اللغات الأخرى مثل: (Python) و (Java) وغيرها، تحتوي (OpenCV) على مجموعة واسعة من الخوارزميات المتخصصة في مجالات التعرف على الأشكال، والوجوه، والحركة، والتتبع، والتصوير الطبي، والروبوتيات، وغيرها، كما أنها تدعم العديد من أنظمة التشغيل، مثل: (Linux) و (Windows) و (Android) و (macOS) و (OS)، وتعدُّ (OpenCV) من أشهر مكتبات رؤية الحاسوب وأكثرها شيوعاً، وتُستخدم على نطاق واسع في العديد من التطبيقات المختلفة.

تعدُّ (OpenCV) أداةً سهلة الاستخدام، ومتعددة المنصات؛ مما يسمح للمطورين بإنشاء تطبيقاتٍ لعددٍ كبيرٍ من الأنظمة الأساسية والهواتف المحمولة، ويمكن أن تُستخدم بنجاح في العديد من التطبيقات المختلفة، مثل: تطبيقات الروبوتات، والمركبات ذاتية القيادة، والتطبيقات الطبية، والعسكرية، وغيرها الكثير.



### 2. تنسرفلو (TensorFlow)

تنسرفلو (TensorFlow): هو إطار عمل (Framework) مفتوح المصدر، مخصص للذكاء الاصطناعي (AI) وتعلم الآلة (Machine Learning)، تم تطويره من قبل فريق (Google Brain) عام 2015.

يستخدم تنسرفلو (TensorFlow) الرياضيات والإحصاء، لتمثيل النماذج الرياضية، والتعلم الآلي، والعمليات الحسابية الأخرى بشكل رمزي.

يتضمن تنسرفلو (TensorFlow) تحميل البيانات وتحليلها، وتصميم النماذج الحاسوبية وتدريبها وتقييمها، ويعتمد على الرسوم البيانية التي تعرض التدفقات الحاسوبية بين مكونات النموذج الحاسوبي، ويحتوي أيضاً على واجهة برمجة تطبيقات قوية؛ تتيح للمطورين العمل بسهولة مع النماذج الحاسوبية المختلفة.

يوفر تنسرفلو (TensorFlow) أيضاً واجهة برمجة تطبيقات (API) لتطوير تطبيقات الذكاء الاصطناعي بلغات البرمجة المختلفة، بما في ذلك (Python) و (C++) و (Java) و (Go) وغيرها، ويتم استخدام تنسرفلو (TensorFlow) على نطاق واسع من قبل العديد من الشركات والمؤسسات؛ لتطوير تطبيقات الذكاء الاصطناعي والتعلم الآلي.

يمكن استخدام تنسرفلو (TensorFlow) في تطبيقات الرؤية الحاسوبية لتصميم وتدريب نماذج تحليل الصور، والتعرف على الأشكال والموجهات والكائنات، وتحليل الفيديو، وتعلم الإشارات، ويمكن أيضاً استخدامه في تصنيف الصور، والكشف عن الوجوه والتعرف عليها، والتعرف على الكلمات والنصوص في الصور.



# TensorFlow

### 3. ماتلاب (Matlab)

ماتلاب (Matlab): هي بيئة برمجية ولغة برمجة عالية المستوى، تُستخدم في العديد من التطبيقات الهندسية والعلمية، ويمكن استخدامها في مجال الرؤية الحاسوبية لتحليل الصور ومعالجتها، باستخدام العديد من الأدوات المتاحة في البرنامج.

يتضمن برنامج ماتلاب (Matlab) العديد من الدّوالّ والأدوات المخصصة للعمليات الحسابية المتعلقة بالصور، مثل: عمليات تحميل الصور وتحويلها إلى أشكال أخرى، وتطبيق الفلاتر، وتحديد الحواف، واستخراج الميزات، والتعرف على الأشياء في الصور، يمكن كذلك استخدامه لتحليل الصور، وتعديلها، وتحسينها، وتصحيحها، وتصميم الخوارزميات الجديدة.

يمكن لـ ماتلاب (Matlab) أن يكون مفيداً للغاية في العديد من تطبيقات الرؤية الحاسوبية، مثل: التعرف على الأشياء، والتحقق من الأمان والمراقبة، والتصوير الطبي، وتحليل الصور الفلكية والإشعاعية والبيولوجية والجيولوجية، يمكن أيضاً استخدام ماتلاب (Matlab) في تطوير تطبيقات التعرف على الوجوه، والتعرف على الأحرف والكلمات، وتحليل الحركات والاهتزازات والصوتيات.

يتميز ماتلاب (Matlab) بسهولة الاستخدام والتعلم، وهو يتضمن واجهة برمجة التطبيقات التي تجعل من السهل إجراء التطبيقات الخاصة بالرؤية الحاسوبية.





### 4. كودا (CUDA)

كودا (CUDA): هي منصة للحوسبة المتوازية، اخترعتها شركة إنفيديا (NVIDIA)، وهي تقنية تُستخدم في الحوسبة الموازية لمعالجة البيانات باستخدام وحدة معالجة الرسومات (GPU)، التي تتيح معالجة مجموعات بيانات كبيرة بشكل فعال وسريع، تم تصميم إنفيديا كودا (NVIDIA CUDA) بشكل خاص لتحسين الأداء عند تشغيل تطبيقات الذكاء الاصطناعي ورؤية الحاسوب.

يتم استخدام إنفيديا كودا (NVIDIA CUDA) في العديد من تطبيقات رؤية الحاسوب؛ لتسريع عمليات التعلم العميق، والشبكات العصبية الاصطناعية، كما تتيح للمطورين والمهندسين إمكانية الوصول إلى العتاد الحاسوبي عالي الأداء، لمعالجة الصور والفيديو بطريقة أكثر فاعلية من حيث الوقت والتكلفة.

توفر إنفيديا كودا (NVIDIA CUDA) مجموعة واسعة من الأدوات والمكتبات التي تساعد على تسريع وتحسين عمليات الرؤية الحاسوبية؛ فيمكن استخدام مكتبة (CUDA Deep Neural Network) (cuDNN)؛ لتسريع عمليات التعلم العميق، وتحسين الأداء بشكل عام.



# NVIDIA CUDA



### 5. كيراس (Keras)

كيراس (Keras): هي مكتبة بايثون (Python) للتعلم العميق، تجمع بين عناصر مكتبات مختلفة مثل: (Tensorflow) و (Theano) و (CNTK)، تتمتع كيراس (Keras) بمكانة أفضل من المنافسين مثل: (Scikit-Learn) و (PyTorch)؛ حيث تعمل على قمة (Tensorflow).

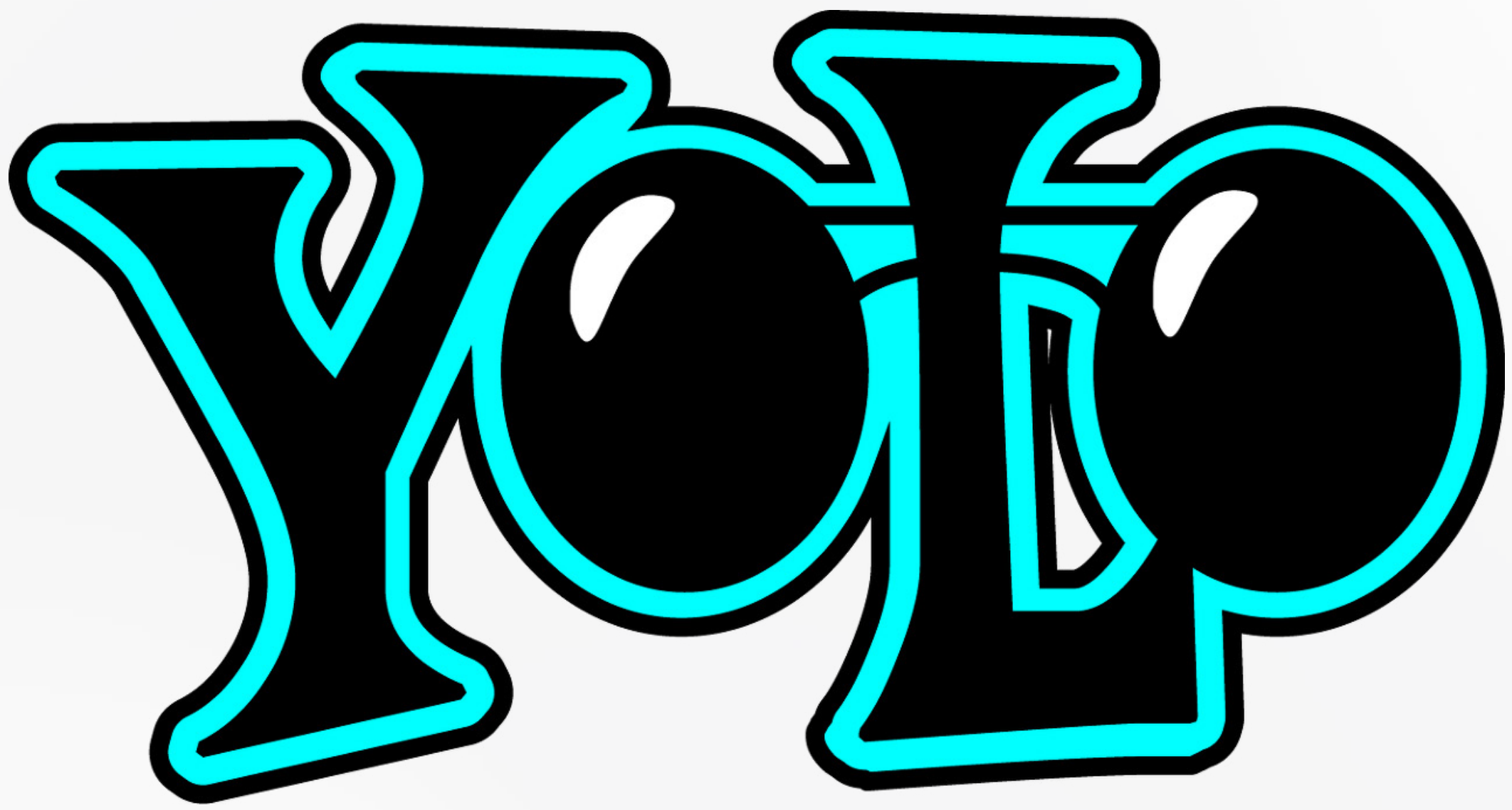
يمكن تشغيل كيراس (Keras) على (TensorFlow) أو (Microsoft Cognitive Toolkit) أو Theano أو PlaidML، وتهدف إلى إجراء تجارب سريعة مع الشبكات العصبية العميقة، تتمحور حول الراحة، والجودة المُقاسة، والقابلية للتوسع، تتبع كيراس (Keras) أفضل الممارسات لتقليل الحمل المعرفي فهي توفر واجهات برمجة تطبيقات ثابتة وأساسية، وتقلل عدد إجراءات المستخدم المطلوبة لحالات الاستخدام المنتظمة.



# Keras

## 6. يولو (YOLO)

يولو (YOLO): هو نظام اكتشاف كائن مخطط خصيصاً للمعالجة في الوقت الفعلي، تم إنشاؤه بواسطة «جوزيف ريدمون» ، و«علي فرهادي» من جامعة واشنطن، تُطبّق خوارزميتهم شبكةً عصبيةً على صورةٍ كاملةٍ، وتقوم الشبكة العصبية بتقسيم الصورة إلى شبكة، وتطبع المناطق بالعناصر المكتشفة.





## Nvidia Jetson TK1

(Nvidia Jetson TK1): سلف (Jetson TX1).

يبلغ سعره: 192 دولاراً، ويمكنك إضافة وحدة معالجة الرسومات (Kepler GK20a) ذات 192 نواة (بسعر: 1 دولار لكل نواة CUDA) للوصول لأداء يبلغ: 300 جيجا فلوب. لا يحتوي (Tk1) على شبكة (WiFi) أو بلوتوث، ولكن يمكن إضافتها عبر منفذ (USB) أو منفذ (mini-PCIe).



### Technical Specifications

<b>Tegra K1 SOC</b>	<ul style="list-style-type: none"><li>Kepler GPU with 192 CUDA cores</li><li>4-Plus-1 quad-core ARM Cortex A15 CPU</li></ul>	1 USB 2.0 port, micro AB	<i>The following signals are available through an expansion port:</i>
2 GB x16 memory with 64 bit width	1 RS232 serial port	1 USB 3.0 port, A	
16 GB 4.51 eMMC memory	1 ALC5639 Realtek Audio codec with Mic in and Line out	DP/LVDS	
1 Half mini-PCIE slot	1 RTL8111GS Realtek GigE LAN	Touch SPI 1x4 + 1x1 CSI-2	
1 Full size SD/MMC connector	1 SATA data port	GPIOs	
1 Full-size HDMI port	SPI 4MByte boot flash	UART	
		HSIC	
		i2c	

### Nvidia Jetson TX1

(Nvidia Jetson TX1) المزود بوحدة المعالجة المركزية (ARM Cortex- A57) رباعية النواة 64 بت، مع وحدة معالجة رسومات (Maxwell 256) نواة.

هو أحد أقوى الأجهزة في السوق لرؤية الكمبيوتر المدمجة؛ ويتميز بتوفير استهلاك الطاقة، حيث يستهلك 10 واط فقط لتقديم أداء (1Teraflop 16FP).

يبلغ سعره 495 دولاراً، ورغم ارتفاع ثمنه إلا أنه يمثل نظاماً بيئياً متكاملًا، وضعت إنفديا (Nvidia) كل خبرتها في صنعه، ليكون خيارك الأمثل لتطبيقات رؤية الكمبيوتر المتطورة.



### Technical Specifications

<b>GPU</b>	256-core NVIDIA Maxwell™ GPU
<b>CPU</b>	Quad-Core ARM® Cortex®-A57 MPCore
<b>Memory</b>	4GB 64-bit LPDDR4 Memory
<b>Storage</b>	16GB eMMC 5.1
<b>Power</b>	Under 10W
<b>PCIE</b>	Gen 2   1x4 + 1x1
<b>CSI</b>	12x CSI2 D-PHY 1.1 lanes (1.5 Gbps/Lane)
<b>Wi-Fi</b>	Yes



### Jetson Nano Developer Kit

(NVIDIA® Jetson Nano™ Developer Kit) عبارة عن كمبيوتر صغير وقوي يتيح لك تشغيل شبكات عصبية متعددة بالتوازي مع تطبيقات، مثل: تصنيف الصور، واكتشاف الكائنات، والتجزئة ومعالجة الكلام. كل ذلك في منصة سهلة الاستخدام تعمل بأقل من 5 واط.

+ \$59.00



#### [VIEW TECHNICAL SPECIFICATIONS >](#)

<b>GPU</b>	128-core Maxwell
<b>CPU</b>	Quad-core ARM A57 @ 1.43 GHz
<b>Memory</b>	4 GB 64-bit LPDDR4 25.6 GB/s
<b>Storage</b>	microSD (not included)
<b>Video Encode</b>	4K @ 30   4x 1080p @ 30   9x 720p @ 30 (H.264/H.265)
<b>Video Decode</b>	4K @ 60   2x 4K @ 30   8x 1080p @ 30   18x 720p @ 30 (H.264/H.265)
<b>Camera</b>	2x MIPI CSI-2 DPHY lanes
<b>Connectivity</b>	Gigabit Ethernet, M.2 Key E
<b>Display</b>	HDMI and display port
<b>USB</b>	4x USB 3.0, USB 2.0 Micro-B
<b>Others</b>	GPIO, I <sup>2</sup> C, I <sup>2</sup> S, SPI, UART
<b>Mechanical</b>	69 mm x 45 mm, 260-pin edge connector

Please refer to NVIDIA documentation for what is currently supported, and the [Jetson Hardware](#) page for a comparison of all Jetson modules.

## Jetson Xavier NX Developer Kit

(Jetson Xavier NX NVIDIA®) تزيد أداء الكمبيوتر الفائق إلى أعلى مستوى، وهي تشتمل على وحدة (Jetson Xavier NX) لتطوير تطبيقات الذكاء الاصطناعي متعددة الوسائط، باستخدام مكدس برامج (NVIDIA) بأقل من 10 واط.

كما تتيح لك الاستفادة من الدعم السحابي؛ لتطوير ونشر برامج الذكاء الاصطناعي بسهولة أكبر على الأجهزة المتطورة.

**\$399.00**



<b>GPU</b>	NVIDIA Volta architecture with 384 NVIDIA CUDA® cores and 48 Tensor cores
<b>CPU</b>	6-core NVIDIA Carmel ARM®v8.2 64-bit CPU 6 MB L2 + 4 MB L3
<b>DL Accelerator</b>	2x NVDLA Engines
<b>Vision Accelerator</b>	7-Way VLIW Vision Processor
<b>Memory</b>	8 GB 128-bit LPDDR4x @ 51.2GB/s
<b>Storage</b>	microSD (not included)
<b>Video Encode</b>	2x 4K @ 30   6x 1080p @ 60   14x 1080p @ 30 (H.265/H.264)
<b>Video Decode</b>	2x 4K @ 60   4x 4K @ 30   12x 1080p @ 60   32x 1080p @ 30 (H.265) 2x 4K @ 30   6x 1080p @ 60   16x 1080p @ 30 (H.264)
<b>Camera</b>	2x MIPI CSI-2 DPHY lanes
<b>Connectivity</b>	Gigabit Ethernet, M.2 Key E (WiFi/BT included), M.2 Key M (NVMe)
<b>Display</b>	HDMI and display port
<b>USB</b>	4x USB 3.1, USB 2.0 Micro-B
<b>Others</b>	GPIO, I <sup>2</sup> C, I <sup>2</sup> S, SPI, UART
<b>Mechanical</b>	103 mm x 90.5 mm x 34.66 mm



### Beagleboard: BeagleBone Black

يشتهر (BeagleBone Black) باستخدامه في تطبيقات إنترنت الأشياء، ويتم مقارنته مع (Raspberry Pi) الذي يحتوي على 26 منفذاً، ويمكن استخدامه للتحكم بـ 8 مدخلات ومخرجات (GPIO) الرقمية، أو كناقل تسلسلي، يحتوي (Beaglebone Black) على رأسين من 48 مقبساً، يمكن استخدامها كمنافذ لأجهزة الإدخال / الإخراج غير المحدودة تقريباً، يتضمن أيضاً عدداً من منافذ الإدخال / الإخراج التناظرية التي تسمح له بالاتصال بمجموعة متنوعة من أجهزة الاستشعار، التي لا يمكن استخدامها مع (Raspberry Pi). يعتبر (BeagleBone Black) خياراً مناسباً لاختبار برمجيات رؤية الحاسب ولكنه لا يعد الخيار الأمثل، نظراً لضعف السعر والأداء المنخفض؛ حيث أن فك تشفير الفيديو، والعرض ثلاثي الأبعاد، وأداء واجهة المستخدم الرسومية العامة أفضل بكثير على (Raspberry Pi 3).



### BeagleBone® Blue

(BeagleBone® Blue) هو كمبيوتر الكل في واحد قائم على (Linux) للروبوتات ، يتكامل في لوحة واحدة صغيرة (2.15 × 3.5 بوصة) معالجات (Octavo OSD3358) جنباً إلى جنب مع (bluetooth) / (wifi و barometer) / (IMU) ، وتنظيم الطاقة وحالة -شحن مصابيح (LED) لبطاريات (LiPo) و(H-Bridges) ذات خليتين وموصلات منفصلة لـ 4 محركات DC + مشغرات و 8 أجهزة مؤازرة، وجميع الحافلات المطلوبة بشكل شائع للأجهزة الطرفية الإضافية في التطبيقات المضمنة.

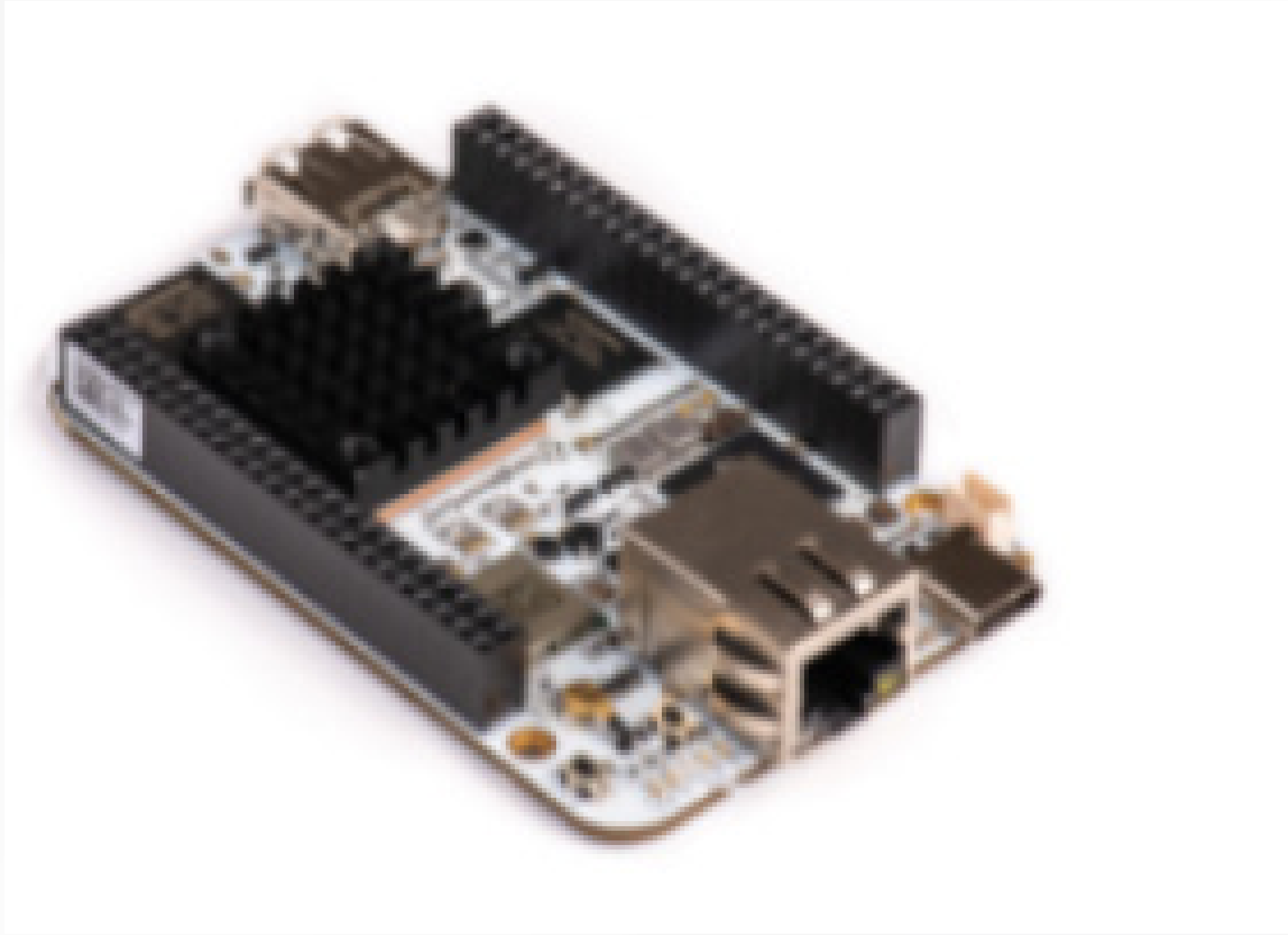




®BeagleBoardorg

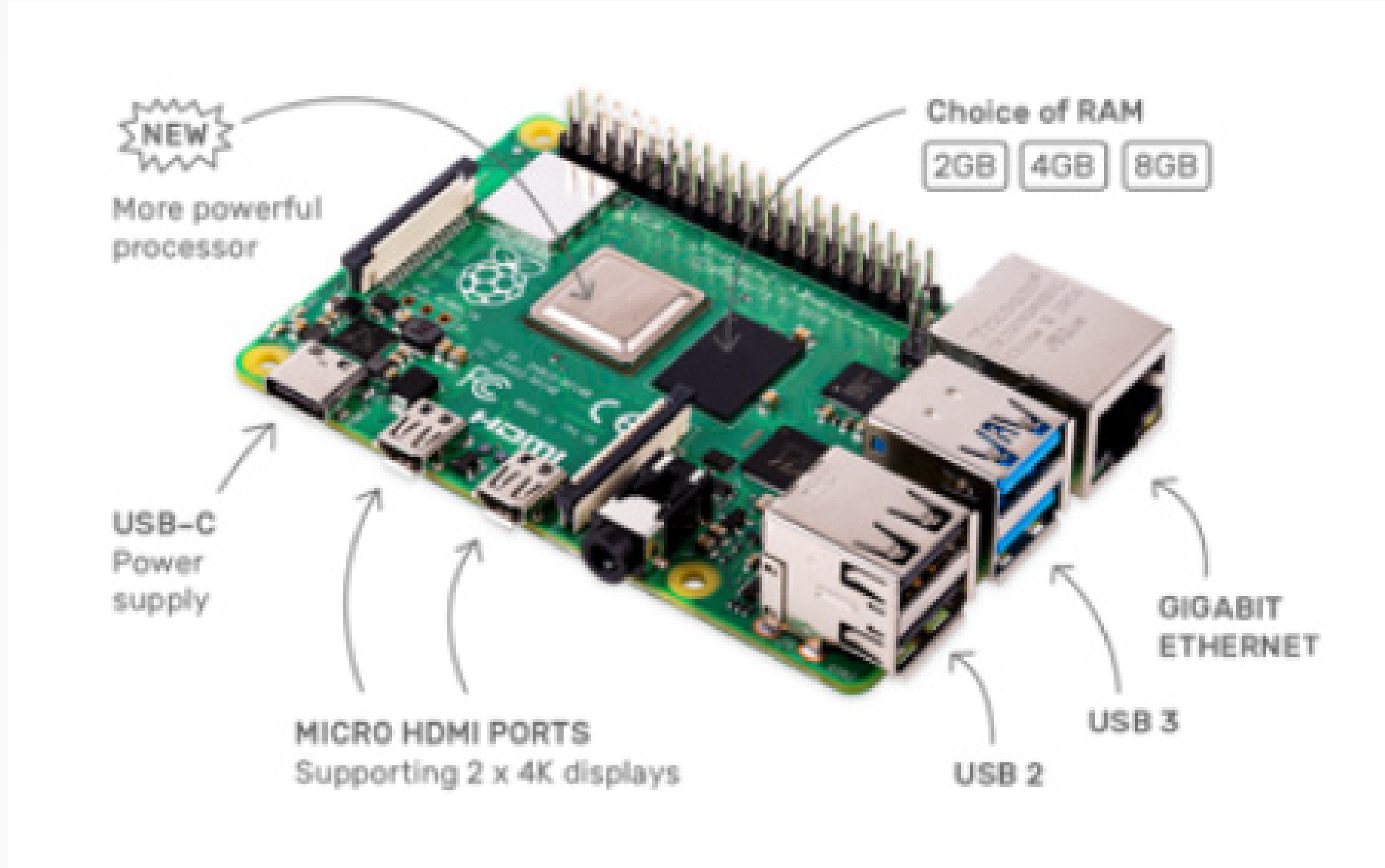
### BeagleBone® AI

يستند (BeagleBone® AI) إلى نهج (Linux) مفتوح المصدر المعتمد من (®BeagleBoardorg)، ويغطي الفجوة بين الوحدات الصغيرة الفعالة، وأجهزة الكمبيوتر الصناعية الأكثر قوة. بناءً على (Texas Instruments AM5729)، يحصل المطورون على وصول إلى النظام على شريحة "SoC" القوي بسهولة عبر رأس (BeagleBone® Black) والتوافق الميكانيكي. (BeagleBone® AI) يسهّل اكتشاف كيفية استخدام الذكاء الاصطناعي (AI) في الحياة اليومية من خلال نوى (TI C66x) لمعالجة الإشارات الرقمية (DSP) ونوى (embedded-vision-engine (EVE)) المدعومة من خلال واجهة (TIDL) المحسنة لتعلم الآلة (OpenCL) مع الأدوات المثبتة مسبقاً. مركزةً على التشغيل التلقائي اليومي في التطبيقات الصناعية والتجارية والمنزلية.



### Raspberry Pi 4

تعد سرعة وأداء الجهاز (Raspberry Pi 4) الجديد تطوراً عن الطرازات السابقة بسرعة 1.5 جيجاهرتز، ومعالج (ARM® Cortex®-A72). يأتي (Raspberry Pi 4) مع (Ethernet) بسرعة جيجابت، بالإضافة إلى الشبكات اللاسلكية المدمجة وبلوتوث، يحتوي على شاشات عرض مزدوجة بدقة 4K، اعتماداً على مقدار ذاكرة الوصول العشوائي التي تحتاجها - 2 غيغابايت أو 4 غيغابايت أو 8 غيغابايت، تبدأ أسعاره من 39.84 دولاراً لـ 2 جيجابايت و 86.37 دولاراً لـ 8 جيجابايت



# لغة البرمجة بايثون (Python)

بايثون (Python) هي لغة ديناميكية عالية المستوى، مجانية ومفتوحة المصدر، وتدعم لغة البرمجة بايثون البرمجة. غرضية التوجه، والبرمجة إجرائية التوجه، من أبرز الاختلافات في لغة البرمجة بايثون عما سبقها من لغات برمجة، لأننا لن نضطر إلى تحديد نوع المتغير عند تعريفه؛ بل يكفي أن نستخدم الأسلوب التالي:  $x=10$  وبعدها سيتم تحديد النوع بشكل تلقائي بناءً على القيمة. قد تبدو المقدمة غير مفهومة تماماً لكم لا تقلقوا هذا أمر طبيعي، لأننا سنوضحه خطوةً بخطوة مع شرح مبسط، فهل أنتم متحمسون لسلسلتنا هذه عن بايثون؟ إذا كان الجواب نعم، فهيا بنا نتعلم لغة بايثون واستخداماتها.

## بعض المصطلحات قبل البدء بالتعرف على لغة بايثون (Python) واستخداماتها :

لنبدأ الآن وقبل الحديث عن البايثون (Python) بشرح بعض المصطلحات التي لا بد من فهمها قبل الخوض في تجربة تعلم أي لغة برمجة وهي كما يلي:

### لغة البرمجة:

هي مجموعة من التعليمات المكتوبة باللغة الإنجليزية، والتي تسمى كل كلمة فيها كلمة مفتاحية، وتستخدم للقيام بمهمة محددة.

### البرمجة غرضية التوجه:

هي الأسلوب الحديث في البرمجة، والذي يدعم تقسيم البرنامج إلى مجموعة من الأجزاء وربطها معاً، بحيث نسمي كل جزءاً منها صف أو فئة (تسمى class). وتسمى باللغة الإنجليزية اختصاراً (OOP) أي (Object Oriented Programming).

### المترجم Compiler:

هو أداة تحول التعليمات البرمجية إلى لغة الآلة، وهي لغة يفهمها الحاسب قبل تنفيذ البرنامج، ومن ثم يتم تنفيذه إذا لم يحتوي على أخطاء برمجية.

### المفسر Interpreter:

هو أداة تنفذ التعليمات البرمجية المكتوبة بلغة عالية المستوى، تعليميةً تلو الأخرى.



# لغة البرمجة بايثون (Python)

## أهم مميزات لغة بايثون :

### 1. سهولة كتابة الكود:

باعتبار أن بايثون (Python) لغة عالية المستوى، فهي سهلة الاستخدام مقارنةً ببقية لغات البرمجة مثل: (C++)، و(C#) و(Java). حيث لن يتطلب منك الأمر سوى بضع ساعات أو أيام حتى تبدأ بفهم آلية الكتابة في البايثون (Python)، فهي لغة سلسلة جداً.

### 2. مجانية ومفتوحة المصدر:

تتوفر لغة البرمجة بايثون بشكل مجاني تماماً على الموقع الإلكتروني الرسمي لها، ويمكن لأي أحد تحميلها مباشرةً، وسنضع الرابط الرسمي في مقالنا هذا، فتابعوا معنا. ومعنى أنها مفتوحة المصدر: أي أنك تستطيع الاستخدام المباشر للكود في أي مشروع وأيضاً مشاركته.

### 3. لغة غرضية التوجه:

كما تحدثنا سابقاً فهي لغة تدعم التوجهات الحديثة في البرمجة، وخاصةً البرمجة غرضية، التوجه وكل ما يتعلق بها من مفاهيم.

### 4. دعم واجهات المستخدم الرسومية (GUI):

تعد واجهات المستخدم الرسومية (تسمى Graphical User Interface) من أهم الميزات التي تدعمها أي لغة برمجة، وتعني إمكانية برمجة تطبيقات تحوي عناصر مرئية للمستخدم، مثل: الواجهات والأزرار والحقول النصية وغيرها، ولغة البرمجة بايثون تدعم هذه الخاصية مما يجعلها لغةً مميزةً.

### 5. ميزة قابلية التوسع:

تخيل أنك تستطيع أن تكتب جزء من الكود البرمجي – وأنت تبرمج بلغة البايثون (Python) – بلغة برمجية أخرى، مثل: (C++)، أي أن الكود سيتحول إلى ما يسمى (byte code)، ومن ثم يمكن استخدامه من قبل أي منصة أخرى.

### 6. لغة قابلة للتفسير Interpreter:

هي لغة قابلة للتنفيذ، تعليمية تلو الأخرى، مثل: بعض لغات البرمجة الأخرى كجافا مثلاً.

### 7. مكتبة ضخمة خاصة بلغة بايثون:

تملك لغة البرمجة بايثون (Python) مكتبة ضخمة من الإضافات والحزم، التي تسهل عليك عملية كتابة الكود البرمجي الخاص بك، عبر تقديم مجموعة من التوابع والوحدات الجاهزة، والمعدة للاستخدام الفوري. أليس هذا رائعاً؟

### 8. لغة ديناميكية:

ومعنى ذلك، أن نوع المتغير يحدد أثناء التنفيذ، ولست بحاجة إلى تعريفه قبل ذلك، كما في لغات البرمجة الأخرى (Java مثلاً).

# لغة البرمجة بايثون (Python)

أين تستخدم لغة بايثون في صناعة البرمجيات؟

Google

NETFLIX

 Dropbox

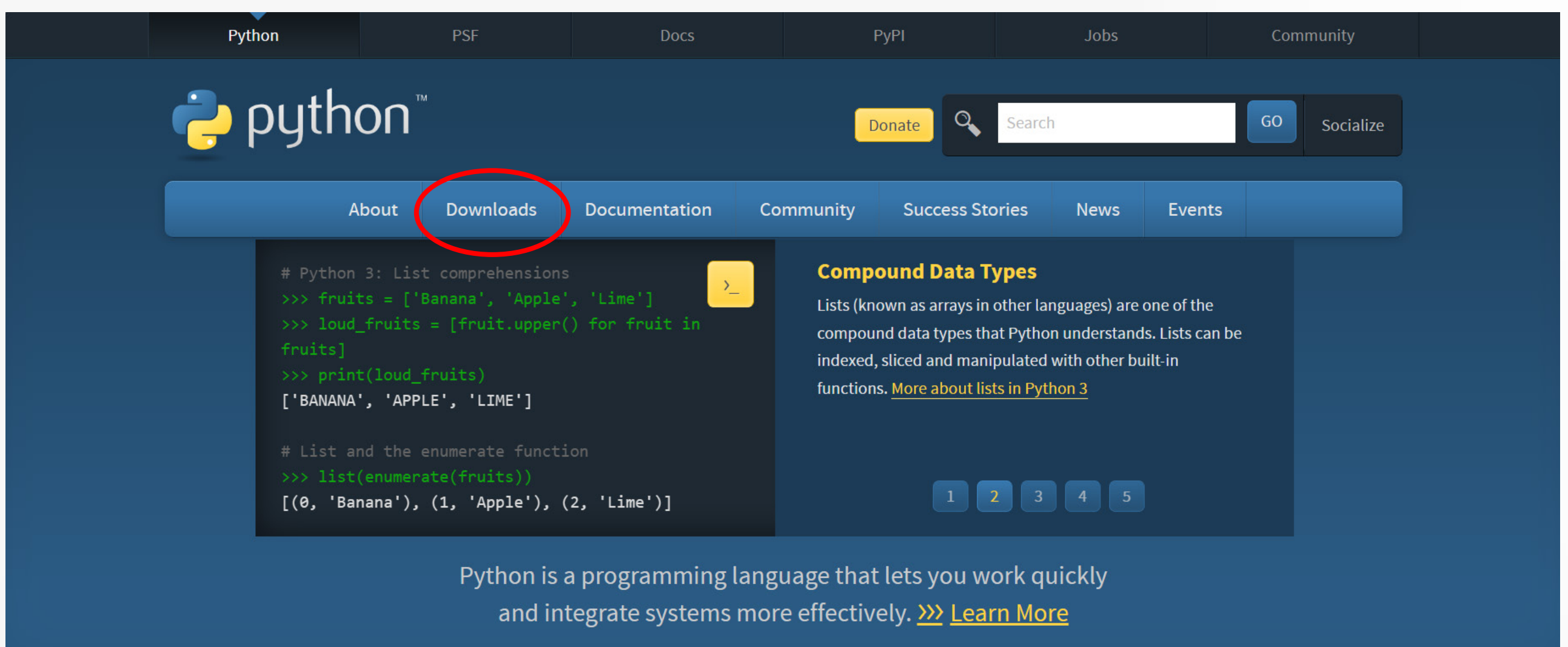
# تنصيب البرمجيات





# تنصيب البرمجيات

Start first by downloading python from its [website](#) then click download from the top bar.



# تنصيب البرمجيات

Then choose the version of python that ompatible with the opencv version you are using ( it depends on the year you are reading this post ). check OpenCV for recommendation.

Looking for a specific release?  
Python releases by version number:

Release version	Release date		Click for more
<a href="#">Python 3.7.7</a>	March 10, 2020	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.8.2</a>	Feb. 24, 2020	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.8.1</a>	Dec. 18, 2019	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.7.6</a>	Dec. 18, 2019	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.8.10</a>	Dec. 18, 2019	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.5.9</a>	Nov. 2, 2019	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.5.8</a>	Oct. 29, 2019	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.5.7</a>	Oct. 10, 2019	<a href="#">Download</a>	<a href="#">Release Notes</a>

[View older releases](#)

# تنصيب البرمجيات

For me i am using python 3.7.6 , and using windows

**Note that Python 3.7.6 *cannot* be used on Windows XP or earlier.**

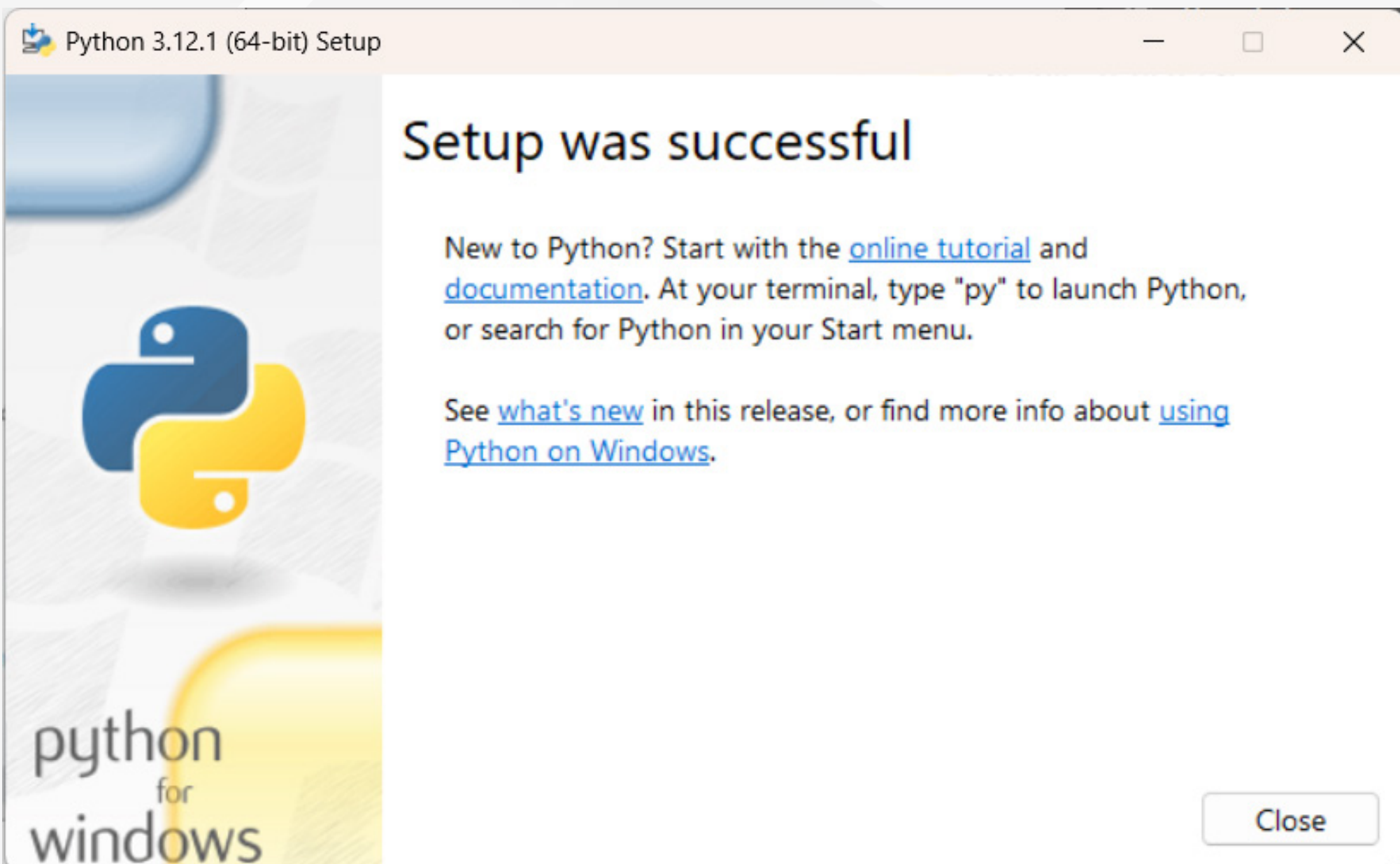
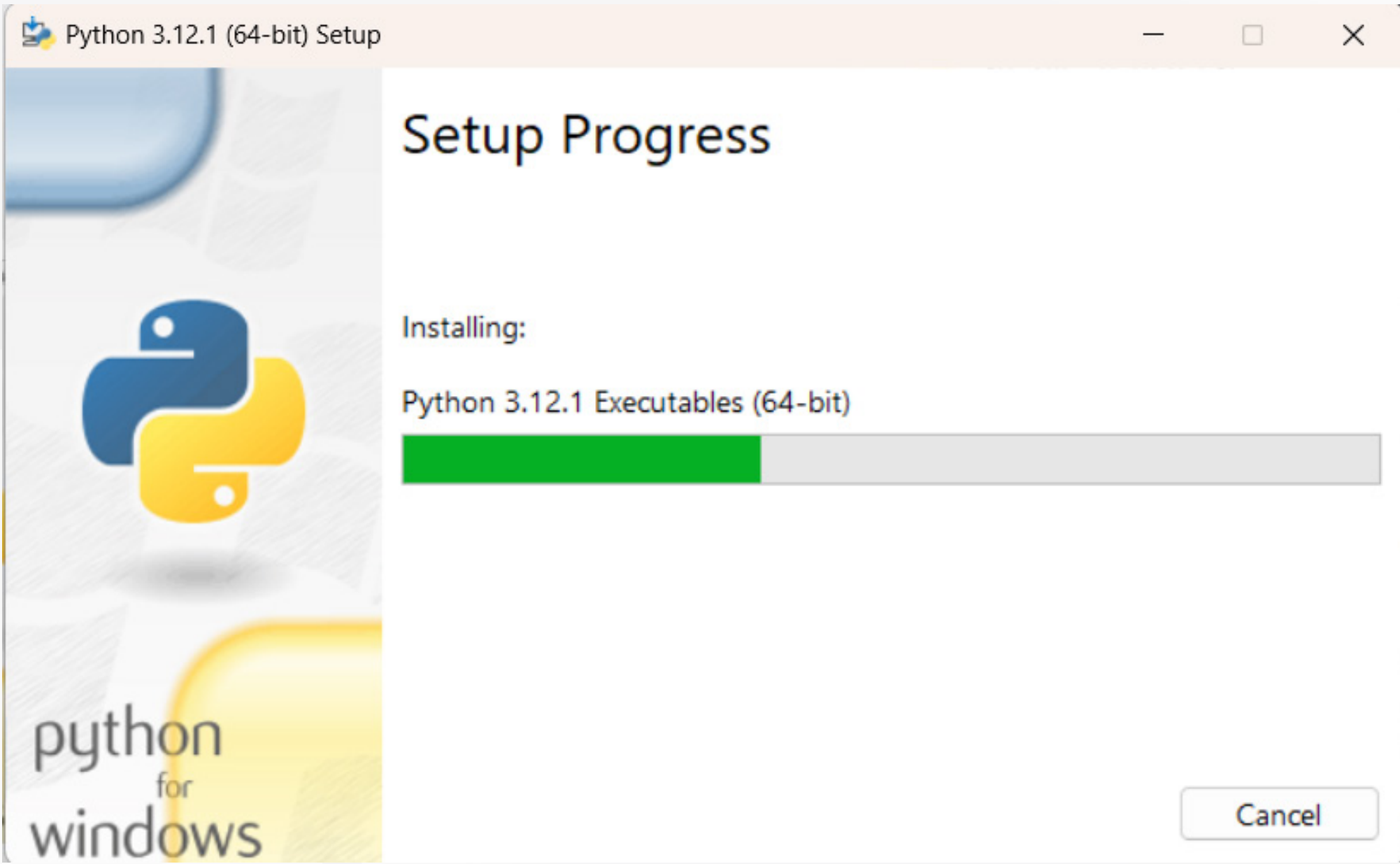
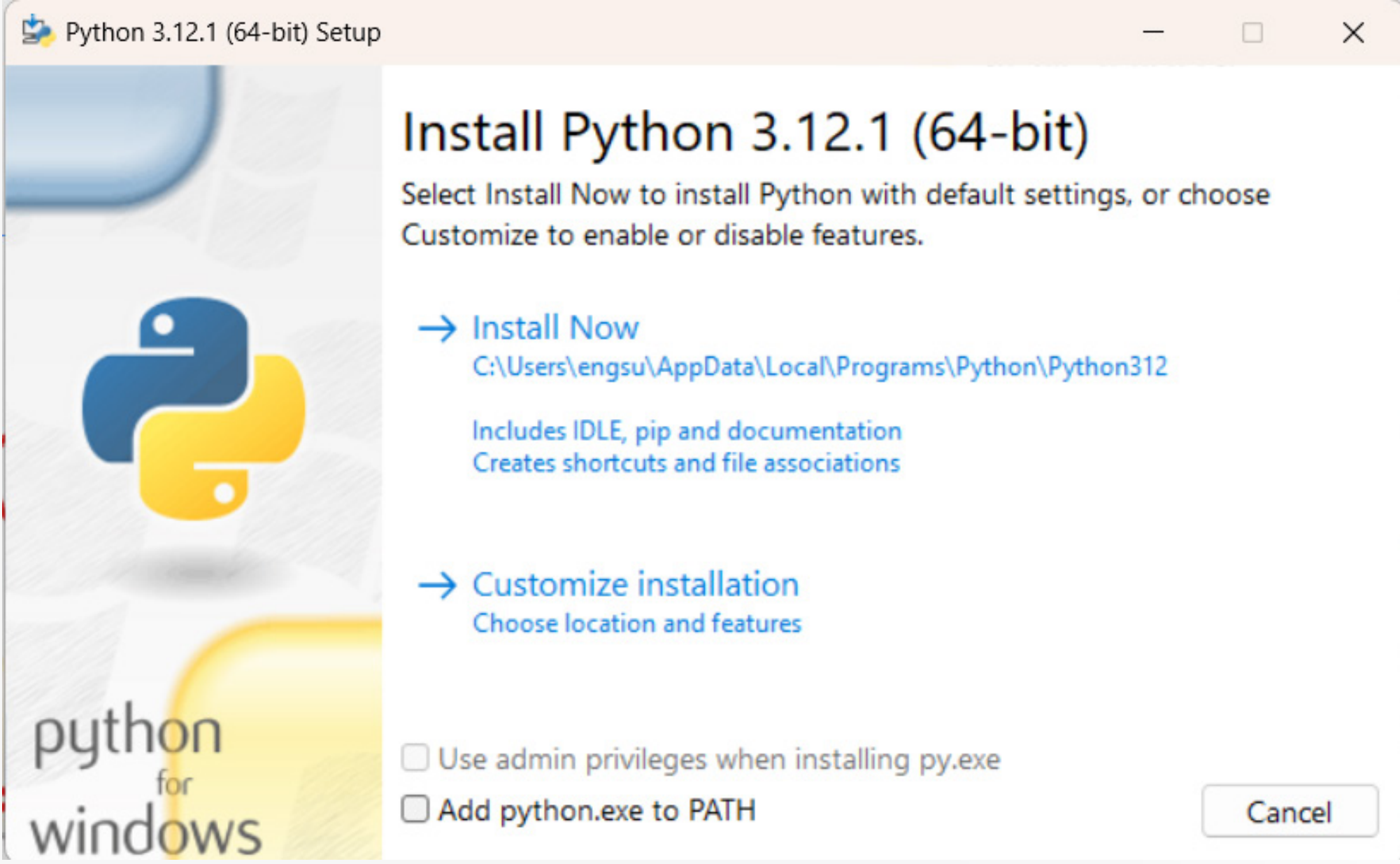
- Download [Windows help file](#)
- Download [Windows x86-64 embeddable zip file](#)
- [Download Windows x86-64 executable installer](#)
- Download [Windows x86-64 web-based installer](#)
- Download [Windows x86 embeddable zip file](#)
- Download [Windows x86 executable installer](#)
- Download [Windows x86 web-based installer](#)
- [Python 3.6.10 - Dec. 18, 2019](#)

## Files

Version	Operating System	Description	MD5 Sum	File Size	GPG	Sigstore
<a href="#">Gzipped source tarball</a>	Source release		51c5c22dcbc698483734dff5c8028606	27053241	SIG	<a href="#">.sigstore</a>
<a href="#">XZ compressed source tarball</a>	Source release		50f827c800483776c8ef86e6a53831fa	20583448	SIG	<a href="#">.sigstore</a>
<a href="#">macOS 64-bit universal2 installer</a>	macOS	for macOS 10.9 and later	eae2d617cbd978a4a6c167924b287572	45377529	SIG	<a href="#">.sigstore</a>
<a href="#">Windows embeddable package (32-bit)</a>	Windows		acc28815c74facc402469e917c8f8433	9849182	SIG	<a href="#">.sigstore</a>
<a href="#">Windows embeddable package (64-bit)</a>	Windows		019788d34af2c60a7be45bf8273e361f	11061655	SIG	<a href="#">.sigstore</a>
<a href="#">Windows embeddable package (ARM64)</a>	Windows		18058aa3c8ccbf3e8fee53386d38c711	10277136	SIG	<a href="#">.sigstore</a>
<a href="#">Windows installer (32 -bit)</a>	Windows		37a89a09138888e6331ec279d68fea8e	25253448	SIG	<a href="#">.sigstore</a>
<a href="#">Windows installer (64-bit)</a>	Windows	Recommended	3e3b6550e58772d324f7519bfa8066dc	26589696	SIG	<a href="#">.sigstore</a>
<a href="#">Windows installer (ARM64)</a>	Windows	Experimental	25fb741f175dc98d5630520f2df931ec	25806784	SIG	<a href="#">.sigstore</a>



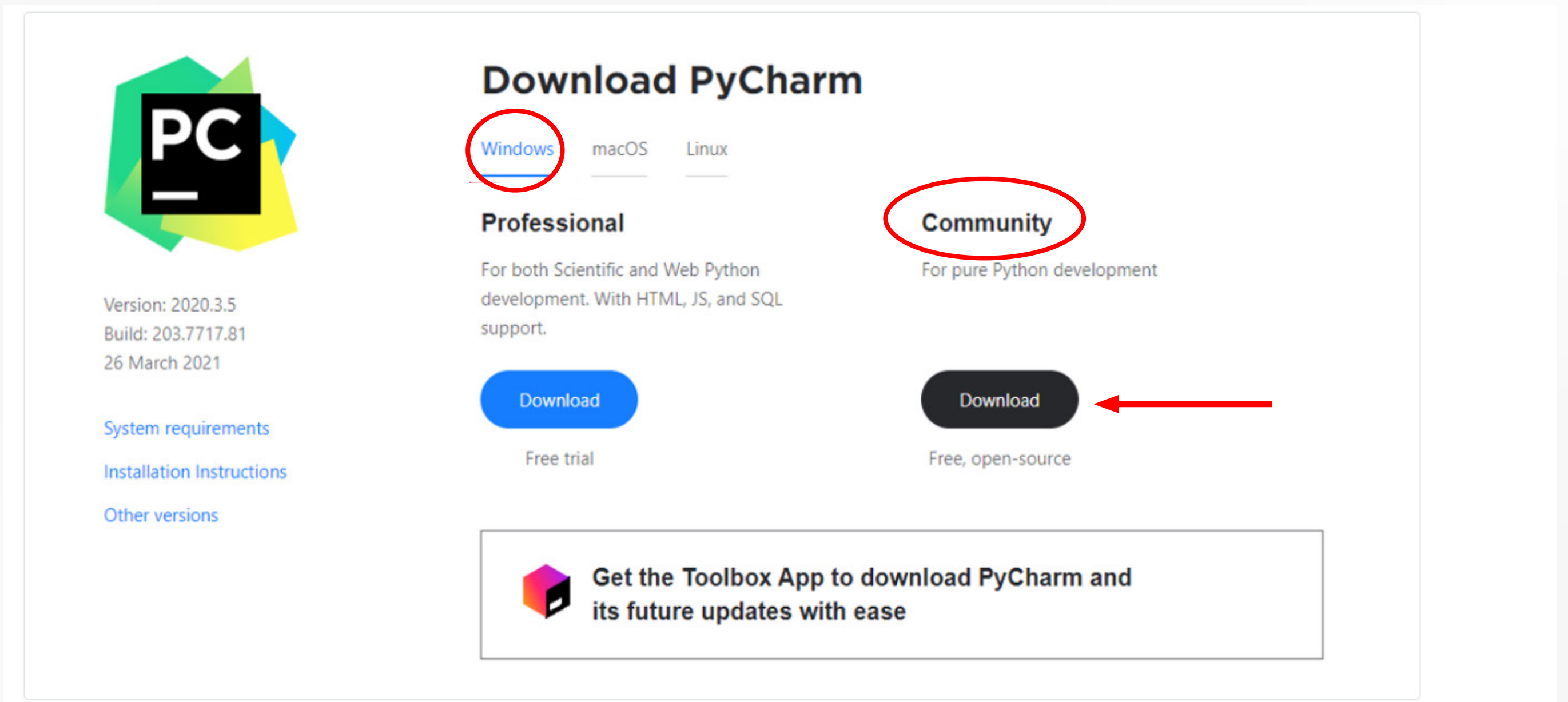
# تنصيب البرمجيات



# تنصيب البرمجيات

## Download and install pycharm

go to the [website](#) and download pycharm and download the community version because its free



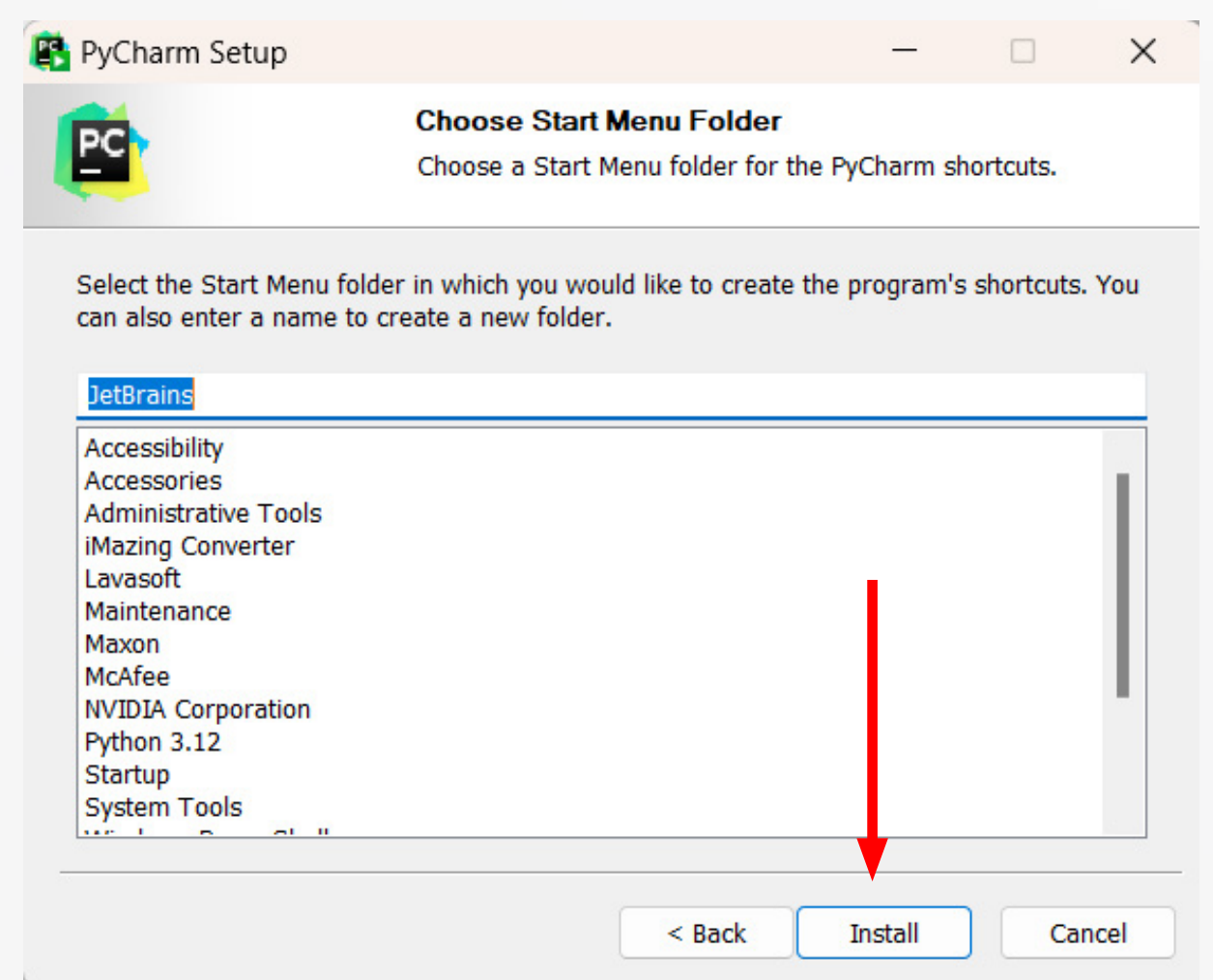
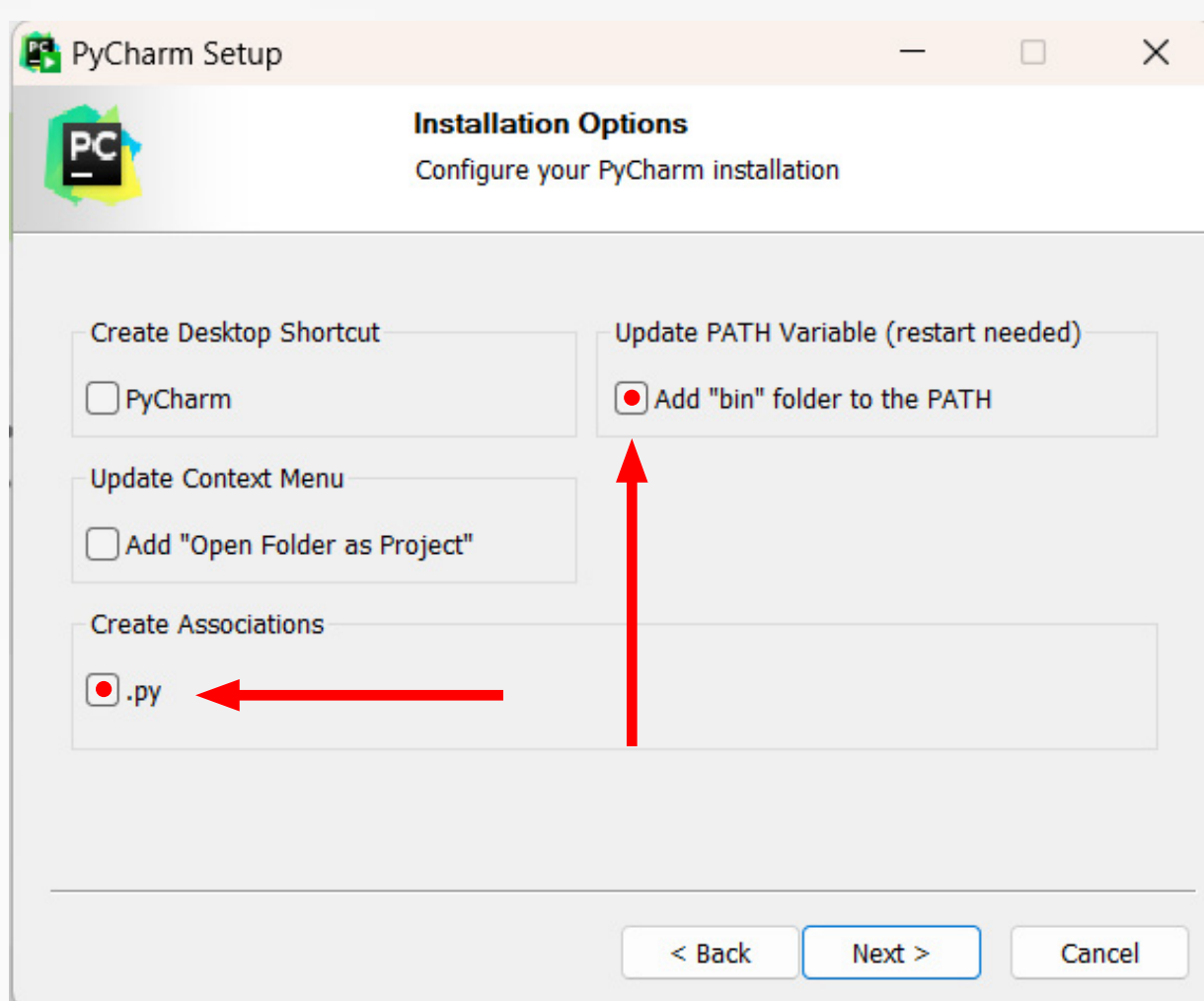
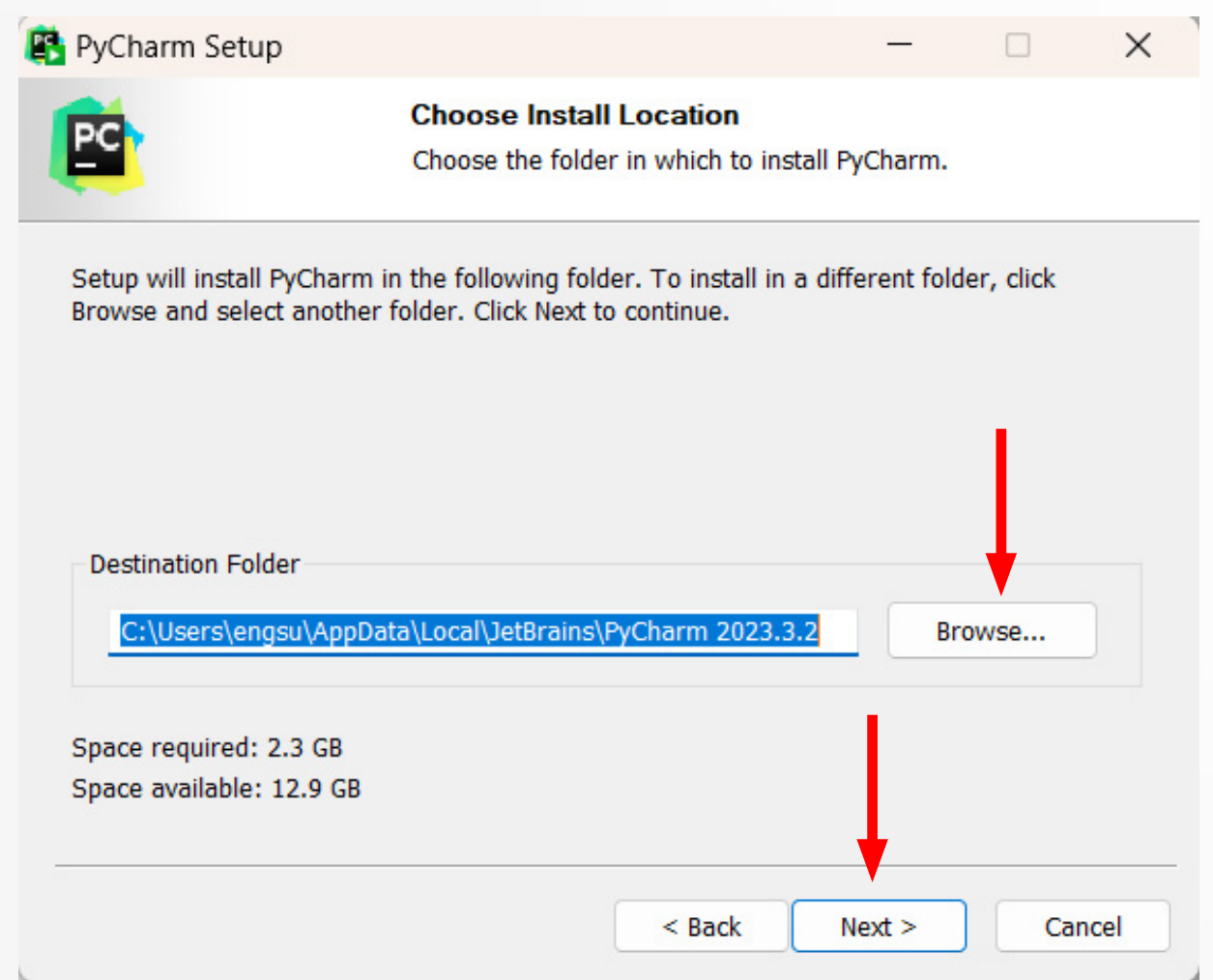
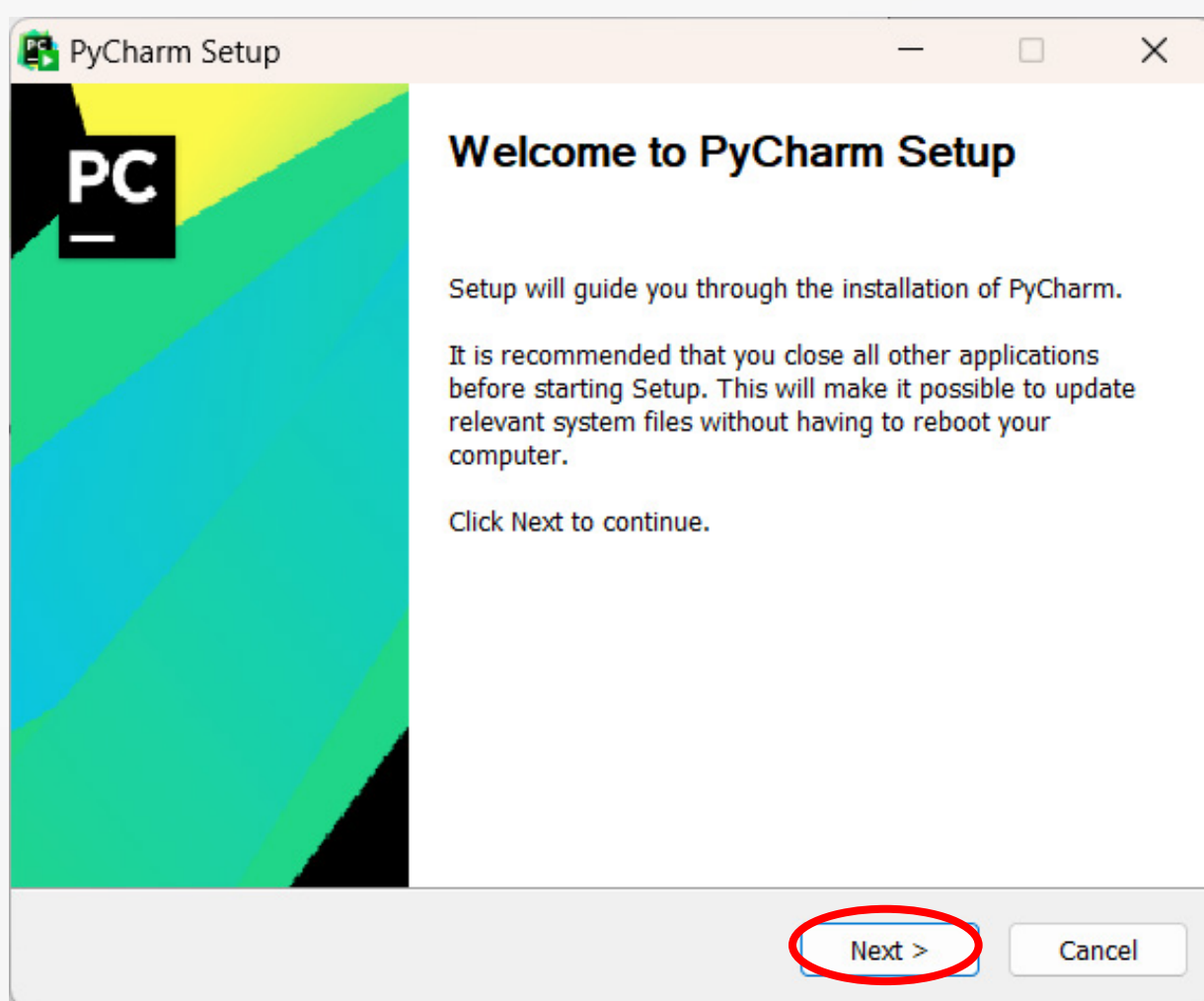
The screenshot shows the PyCharm download page. On the left is the PyCharm logo (PC) and version information: Version: 2020.3.5, Build: 203.7717.81, 26 March 2021. Below this are links for System requirements, Installation Instructions, and Other versions. The main content area is titled "Download PyCharm" and has three tabs: Windows (circled in red), macOS, and Linux. Under the Windows tab, there are two options: "Professional" (Free trial) and "Community" (Free, open-source). The "Community" option is circled in red, and a red arrow points to its "Download" button. At the bottom, there is a box with the text "Get the Toolbox App to download PyCharm and its future updates with ease" and a small icon.



# تنصيب البرمجيات

## Download and install pycharm

Complete installation process and make sure to add path

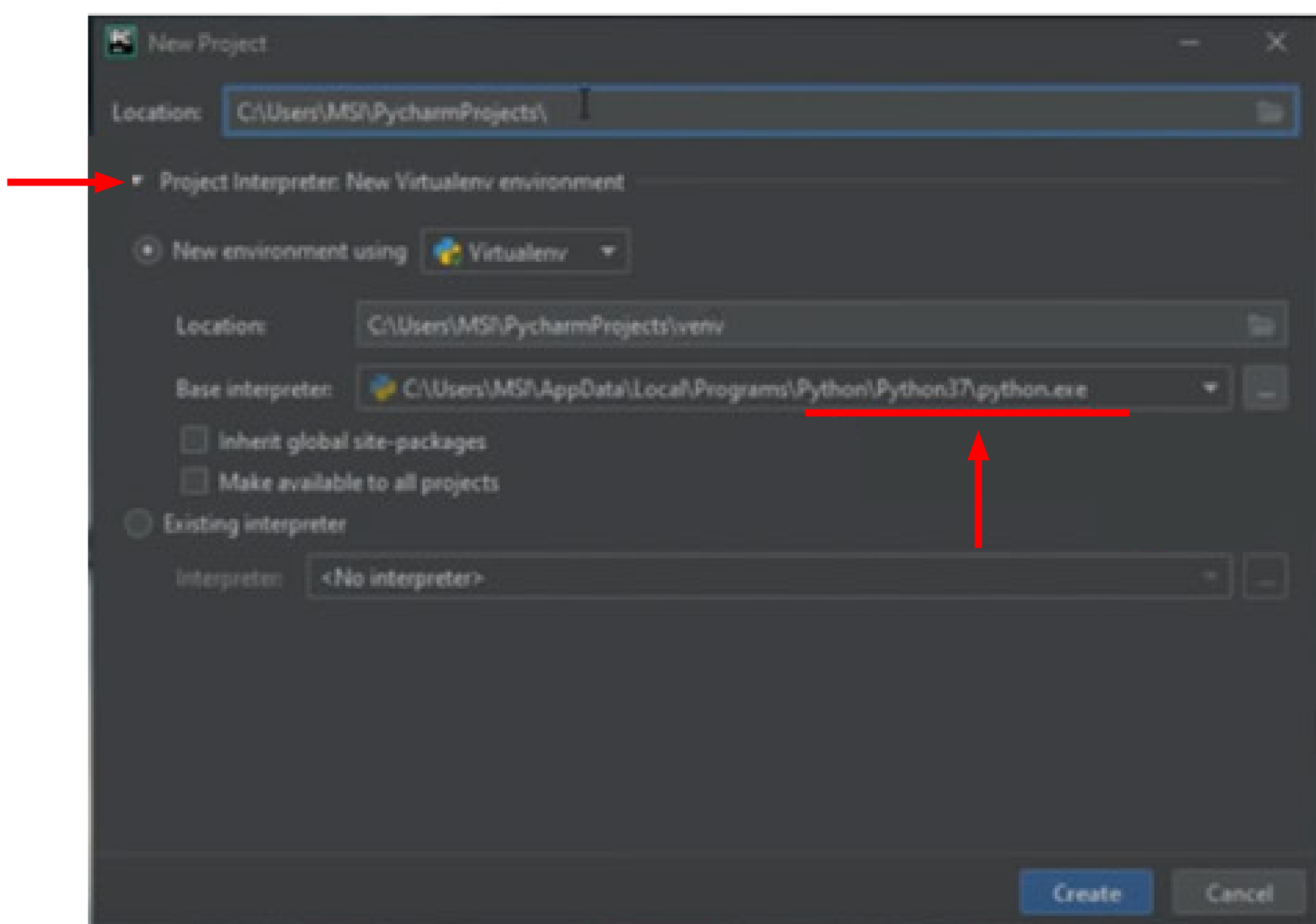
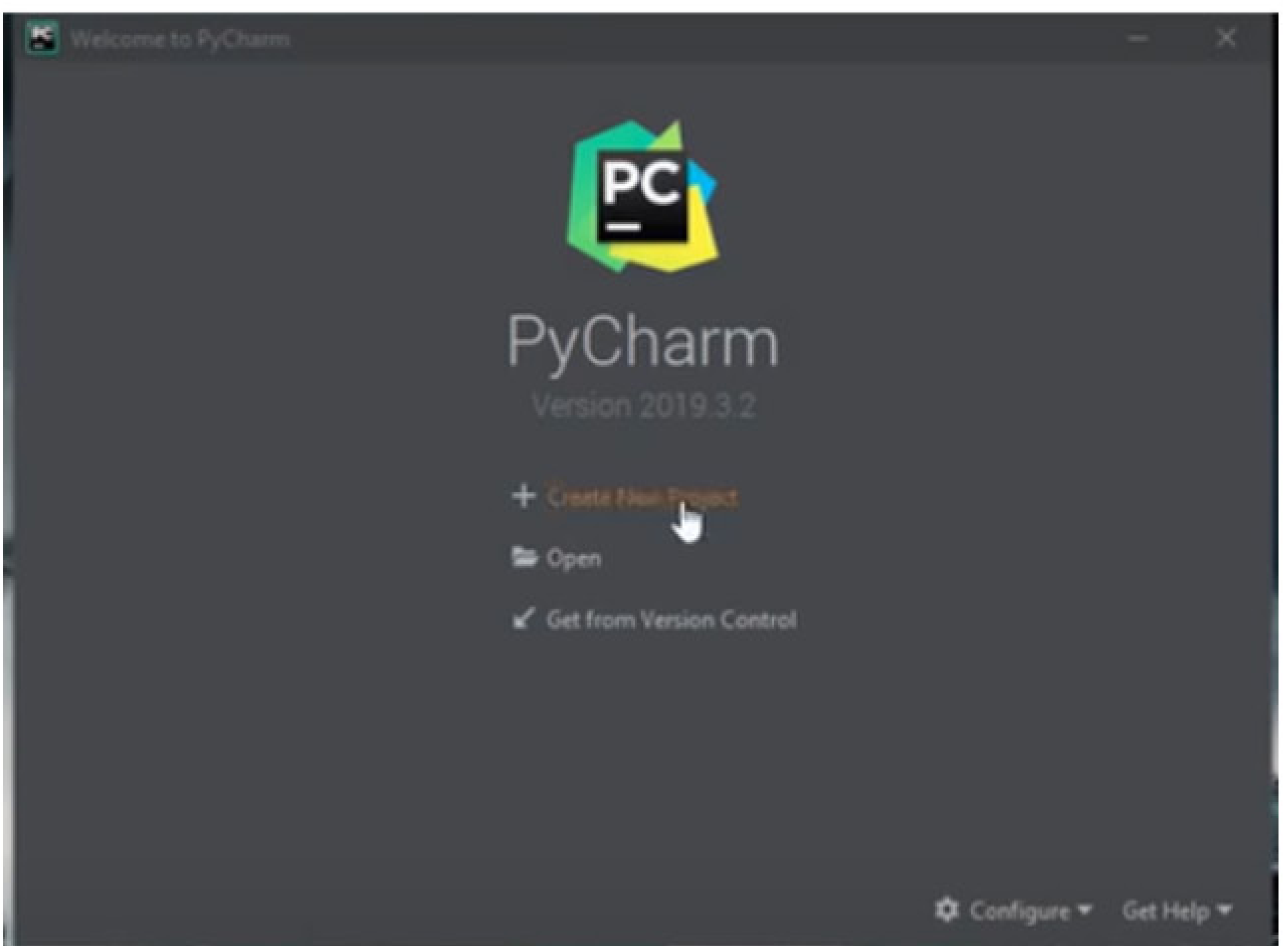




# تنصيب البرمجيات

## Create project

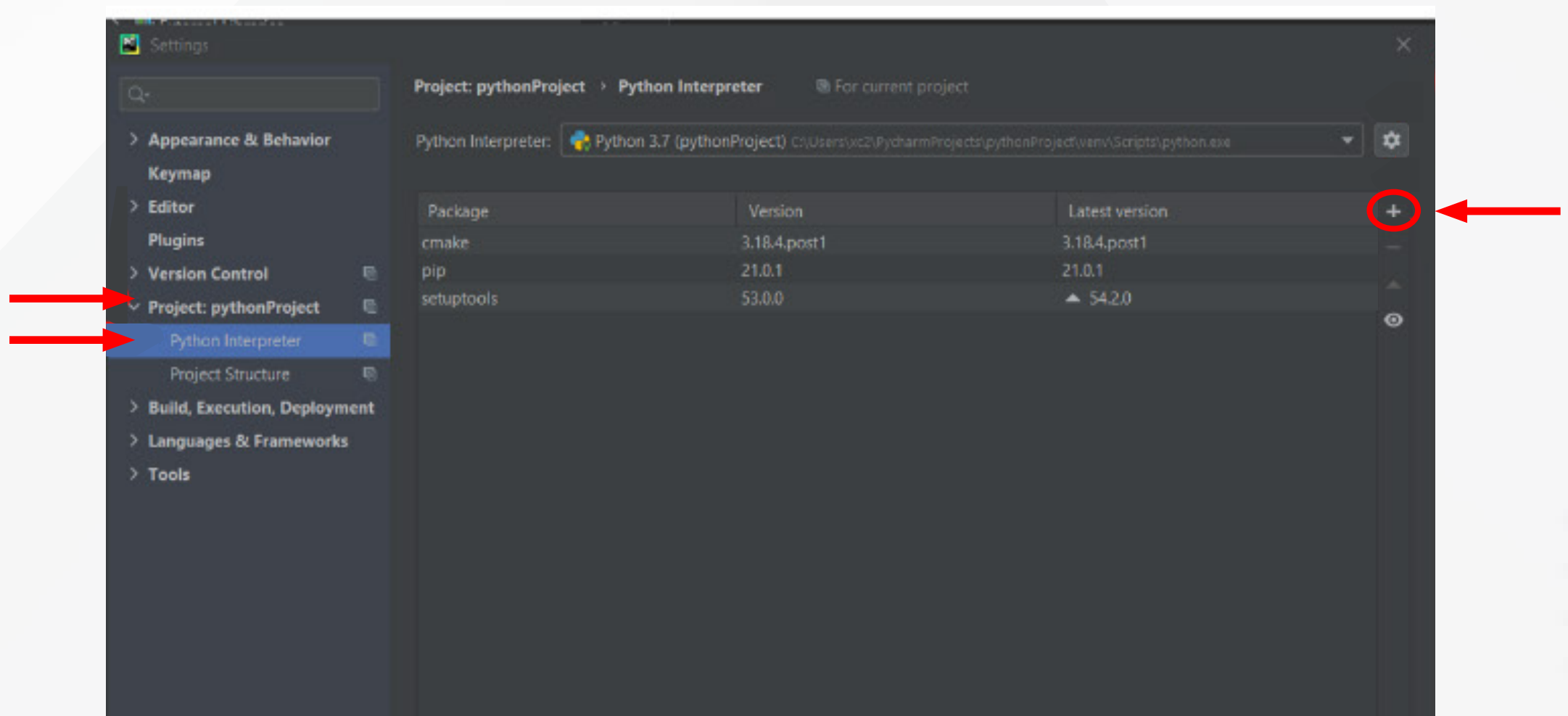
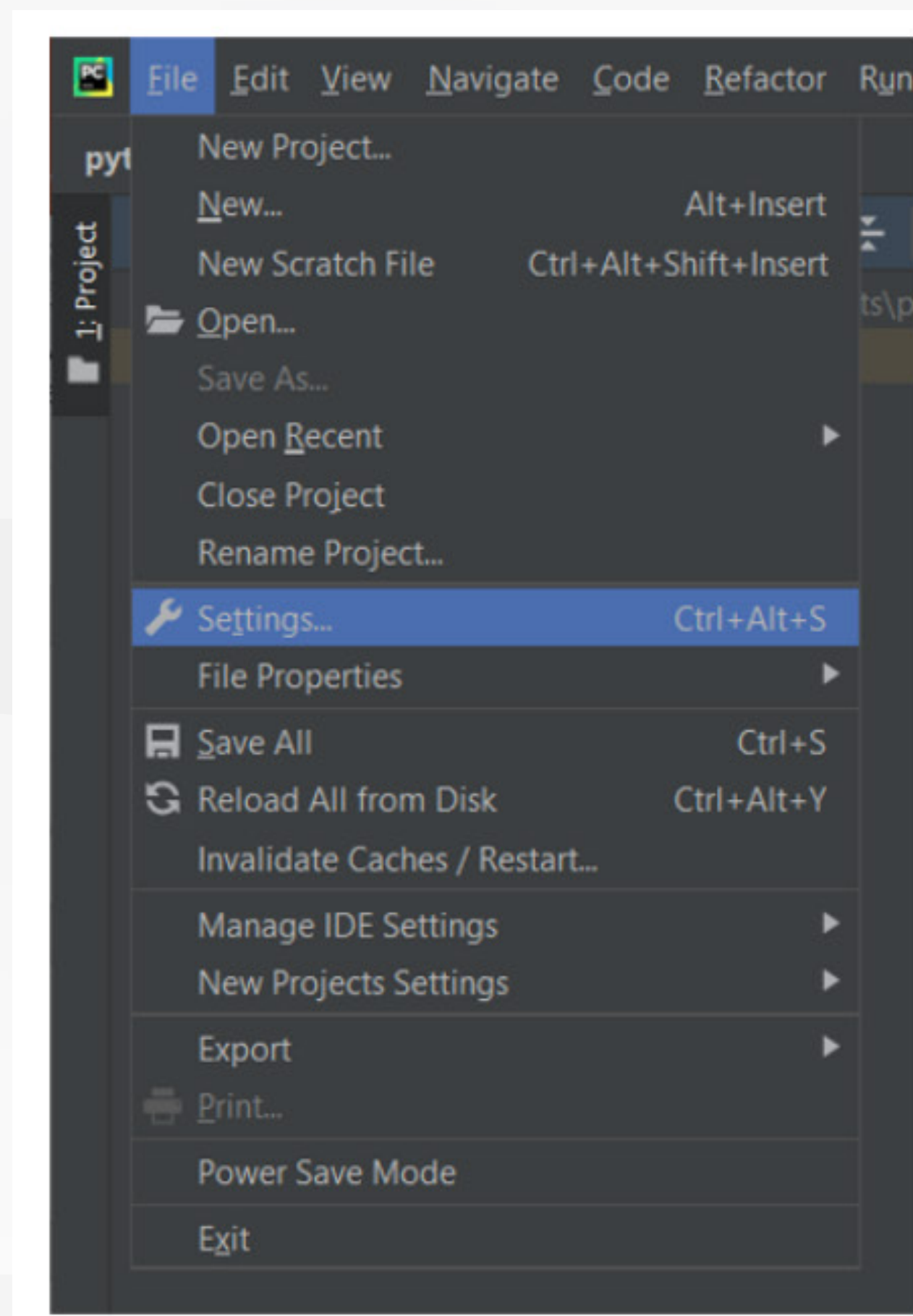
Create a new project , and make sure to make the interrupter as python 3.7



# تنصيب البرمجيات

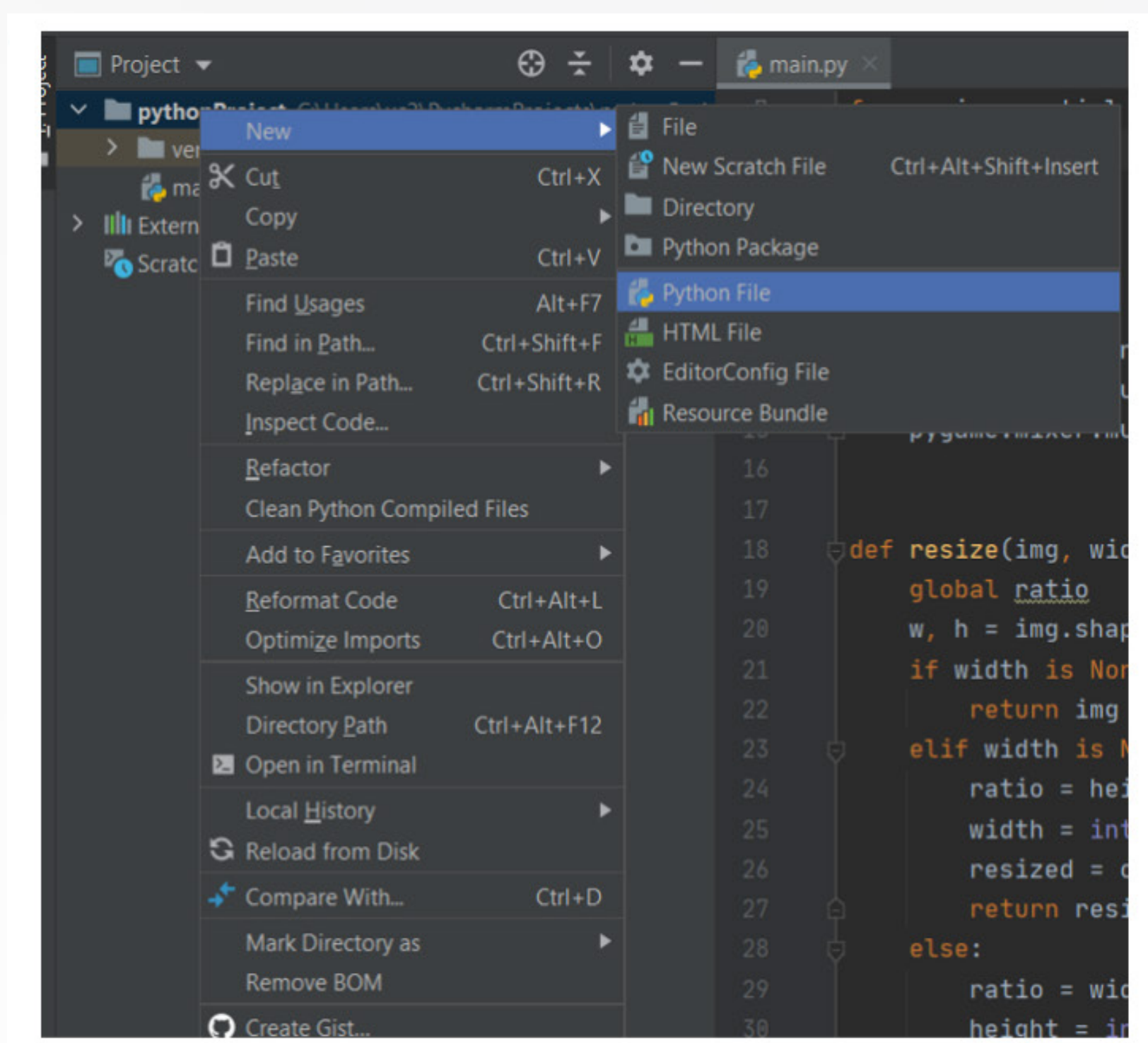
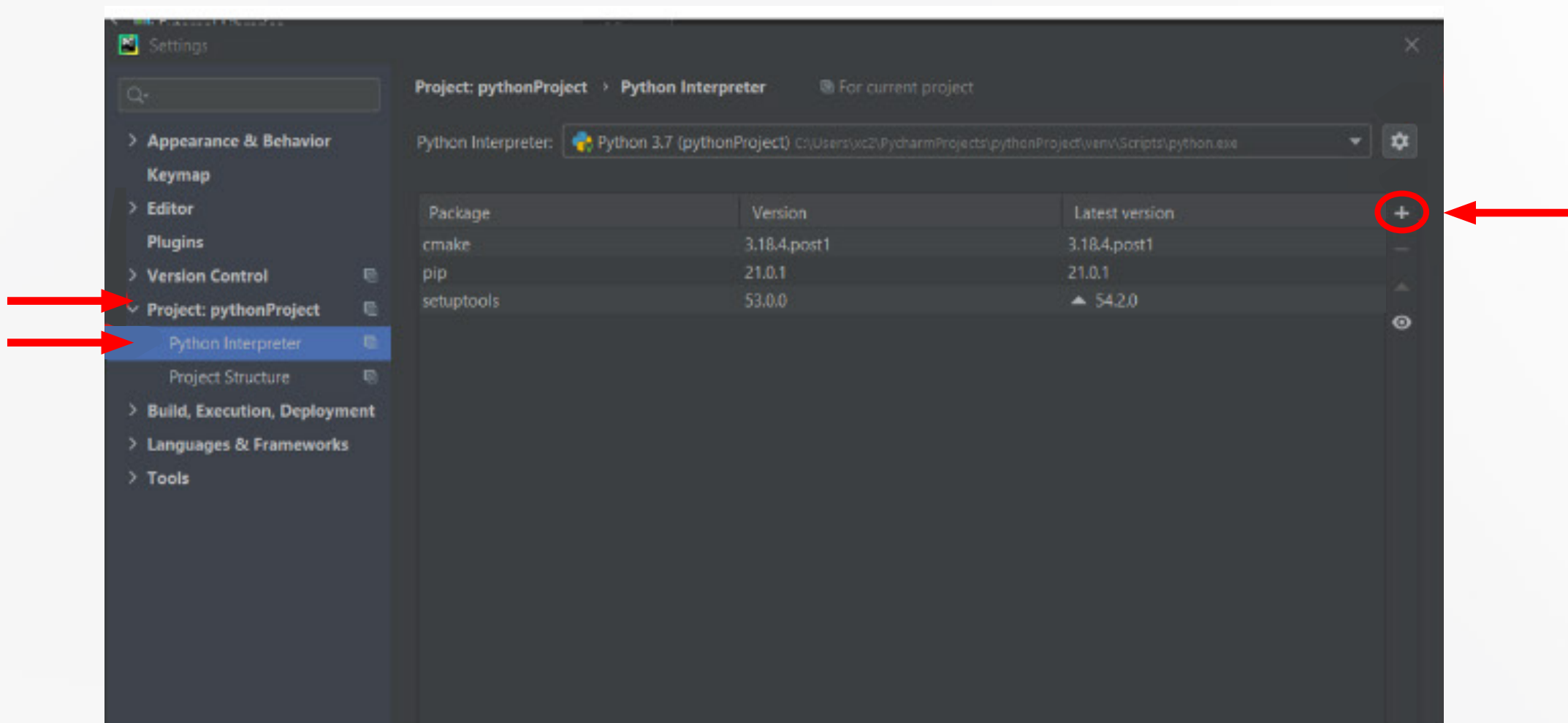
## install opencv

Go to file > setting > project pythonProject > python interupeter , then click ad



# تنصيب البرمجيات

Now search for Open cv > then choose OpenCV-python and click install the package, close the window now and create your first python project by right clicking on the project you have created And go to new and create new python file





# مراجعة

## برنامج اليوم الأول:

اليوم التدريبي	العنوان	الموضوع	المحور التفصيلي	الزمن	
الاول	مقدمة البرمجة باستخدام بايثون (Python) + تطبيق عملي باستخدام PyCharm	الذكاء الاصطناعي	تعارف	30 دقيقة	
		المقدمة	التعريف بمركز ذكاء مقدمة عن المعسكر مقدمة عن رؤية الحاسب	60 دقيقة	
			استراحة		15 دقيقة
		تطبيقات الرؤية الحاسوبية	التعلم العميق مقابل رؤية الكمبيوتر التقليدية تطبيقات الرؤية الحاسوبية والذكاء الاصطناعي	60 دقيقة	
			استراحة		15 دقيقة
		العتاد المستخدم في الرؤية الآلية	البرامج المستخدمة في الرؤية الآلية العتاد المستخدم في الرؤية الآلية	60 دقيقة	

## برنامج اليوم الثاني:

اليوم التدريبي	العنوان	الموضوع	المحور التفصيلي	الزمن
الثاني	مقدمة البرمجة باستخدام بايثون + تطبيق عملي باستخدام (PyCharm)	الذكاء الاصطناعي	تعارف	30 دقيقة
		أساسيات البرمجة	أساسيات البرمجة (if Else State) أساسيات البرمجة (for , While loop) أساسيات البرمجة (functions) تنصيب (PyCharm)	60 دقيقة
			استراحة	15 دقيقة
		مقدمة (Open CV)	كيف يحلل الكمبيوتر الصور تنصيب (PyCharm)+ (Open CV)	60 دقيقة
			استراحة	15 دقيقة
		(Open CV)	مقدمة (Open CV) كتابة مثال وتطبيق عملي	60 دقيقة



# بايثون (Python)

## قواعد كتابة الكود في بايثون (Python) هي :

1. Case Sensitivity تعني أن لغة البرمجة تميز بين الأحرف الكبيرة و الأحرف الصغيرة. بايثون (Python) تعامل الأسماء التي نستخدمها بتأن، سواءً كنا نضع هذه الأسماء للمتغيرات، أو الدوال، أو الكلاسات، أو الكائنات إلخ. مثال: (note) و (Note) ليسوا شيئاً واحداً.
2. لا تقم بإضافة أي مسافة فارغة باستخدام (TAP)؛ لأن المسافة التي يعطيها هذا الزر غير مسموح استخدامها في لغة بايثون (Python).
3. استخدام 4 مسافات فارغة (Space) عند وضع الكود بشكل متداخل.

### اسم الكلاس (Class Name)

دائماً ابدأ اسم الكلاس بحرف كبير، وفي حال كان اسم الكلاس يتألف من أكثر من كلمة، اجعل أول حرف من كل كلمة كبيراً.

- ضع سطرًا فارغًا على الأقل بين السطر الذي تم فيه تعريف الكلاس، والدوال المعرفة بإدخاله.

# بايثون (Python)

## قواعد كتابة الكود في بايثون هي التالية :

1. ضع سطرًا فارغًا على الأقل بين كل دالتين.
2. ضع سطرًا فارغًا بين كل اثنين بلوك تضيفها بداخل الدوال.
3. ضع مسافةً فارغةً حول جمل التحكم، وجمل الشرط.
4. عند وضع التعليقات يفضل استخدام الرمز # في بداية كل سطر حتى وإن كان التعليق يتألف من عدة أسطر.
5. عدد الأحرف القصوى التي يمكن وضعها في كل سطر هو 79 حرفاً.

## Examples:

```
message = input("Type a message to yourself: ")  
print("You said:", message)
```

```
number = int (input ("Type a nuber:"))  
print("You entered: ", number)
```



int ( )

float ( )

bool ( )

## تمرين 1

اكتب سؤالين للإجابة على:

- اسم الشخص
- لونه المفضل
- ثم اطبع، مثال: ( Sara Likes Blue )

```
name = input ( 'What is your name? ' )  
favorite_color = input ( 'What is your favorit')  
Print (name + ' likes ' + favorite_color)
```



## تمرين 2

اكتب سؤالاً عن:

- الاسم
- وسنة الميلاد
- ثم اطبع الاسم والعمر

```
name=input('what is your name: ')
birth_date=input('what is your birth date: ')
birth_date=int(birth_date)
now_year=2021
age= now_year - birth_date
print( 'your name is '+name + 'your age is'+
format(age))
```

```
birth_year = input( 'Birth year:  ')
age = 2019 - birth_year
print (age)
```

Ask a user their weight (in pounds), convert it to kilograms and print on the terminal.



```
weight_lbs = input ('Weight (lbs): ')
weight_kg = int (weight_lbs) * 0.45
print (weight_kg)
```

## Strings:

```
course = "Python's Course for Beginners"  
print (course)
```

```
course = 'Python for "Beginners" '  
print (course)
```

```
course = '''  
Hi John,
```

```
This is our first email to you. Thank you for  
showing interest in our course.
```

```
Best regards,  
The Support Team  
'''
```

```
print(course)
```

## Strings:

```
course = 'Python for Beginners'  
#012345  
print (course [0])
```

```
course = 'Python for Beginners'  
print (course [-1])
```

```
course = 'Python for Beginners'  
#0123  
print (course [0:3])
```

```
course = 'Python for Beginners'  
print (course [0:])
```

```
course = 'Python for Beginners'  
print (course [1:])
```

```
course = 'Python for Beginners'  
print (course [:5])
```

```
course = 'Python for Beginners'  
print (course [:])
```

## Strings:

```
course = 'Python for Beginners'  
another = course [:]  
print (another)
```

```
name = 'Jennifer'  
print (name [1:-1])
```

```
ennife
```

```
ennife
```



## Formatted text:

```
first = 'John'
last = 'Smith'
message = first + '[' + last + ']' is a code'
John [Smith] is a coder
```

```
first = 'John'
last = 'Smith'
message = first + '[' + last + ']' is a code'
print (message)
```

```
first = 'John'
last = 'Smith'
message = first + '[' + last + ']' is a code'
msg = f'{first} [{last}] is a coder'
print (message)
```

## Function vs method:

```
course = 'Python for Beginners'  
print(len(course))
```

```
course = 'Python for Beginners'  
print (len(course) )  
course. upper ()
```

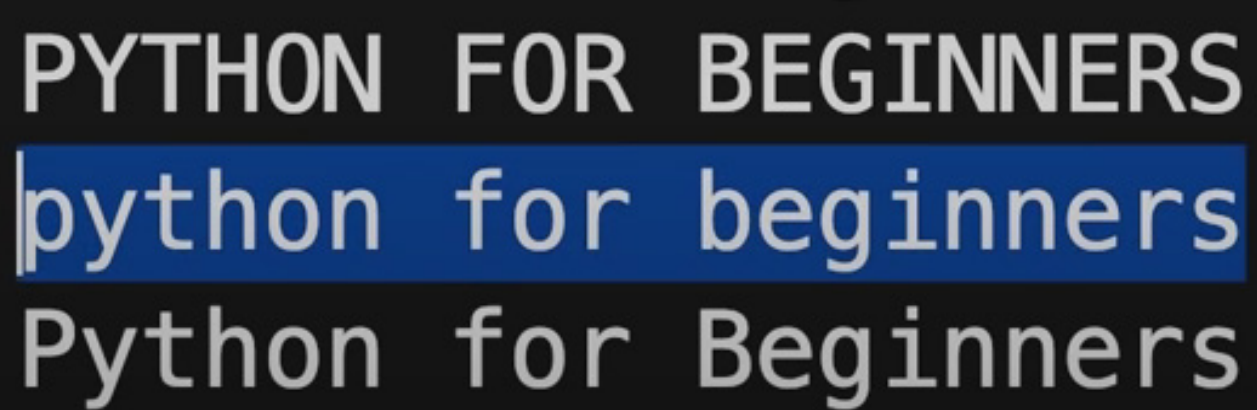
```
1 course = 'Python for Beginners'  
2 print(len(course))  
3 course.  
   • upper(self) str  
   • replace(self, old, ... str  
   • capitalize(self) str  
   • lower(self) str  
   • casefold(self) str  
   • center(self, width,... str  
   • count(self, x, __st... str  
   • encode(self, encoding, errors) str  
   • endswith(self, suff, str
```

```
course = 'Python for Beginners'  
print (len (course))
```

## Function vs method:

```
course = "Python for Beginners"
print (course. upper ())
print (course)
```

```
course = 'Python for Beginners'
print (course. upper () )
print (course. lower ())
print (course)
```

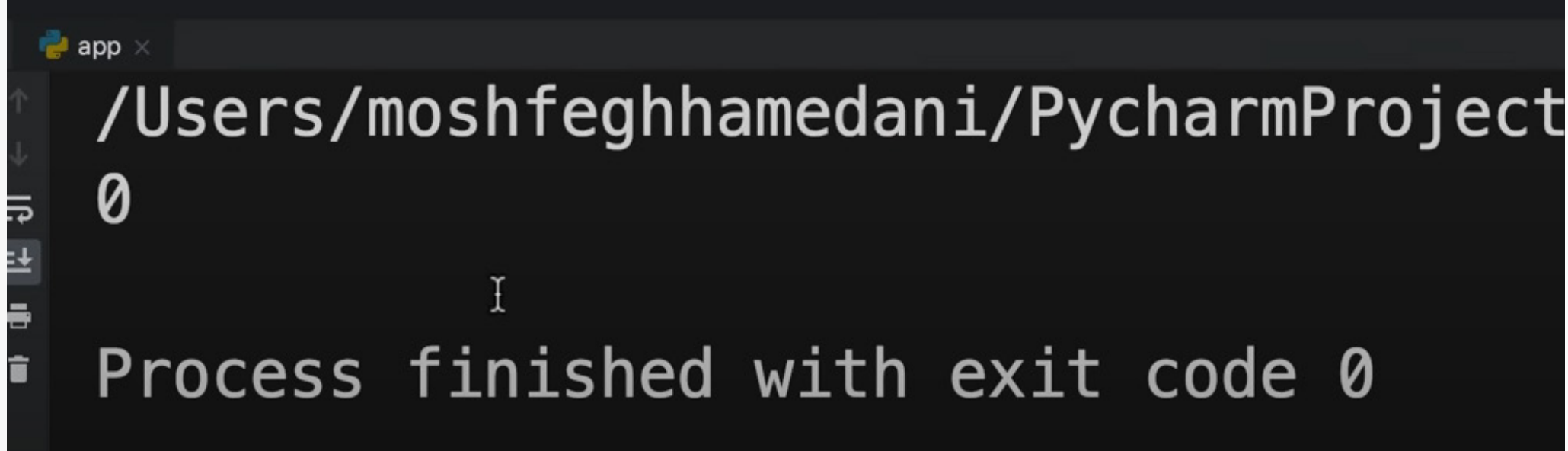


A terminal window with a black background and white text. The first line is 'PYTHON FOR BEGINNERS'. The second line is 'python for beginners', where the text is highlighted in blue. The third line is 'Python for Beginners'.

```
PYTHON FOR BEGINNERS
python for beginners
Python for Beginners
```

## Function vs method:

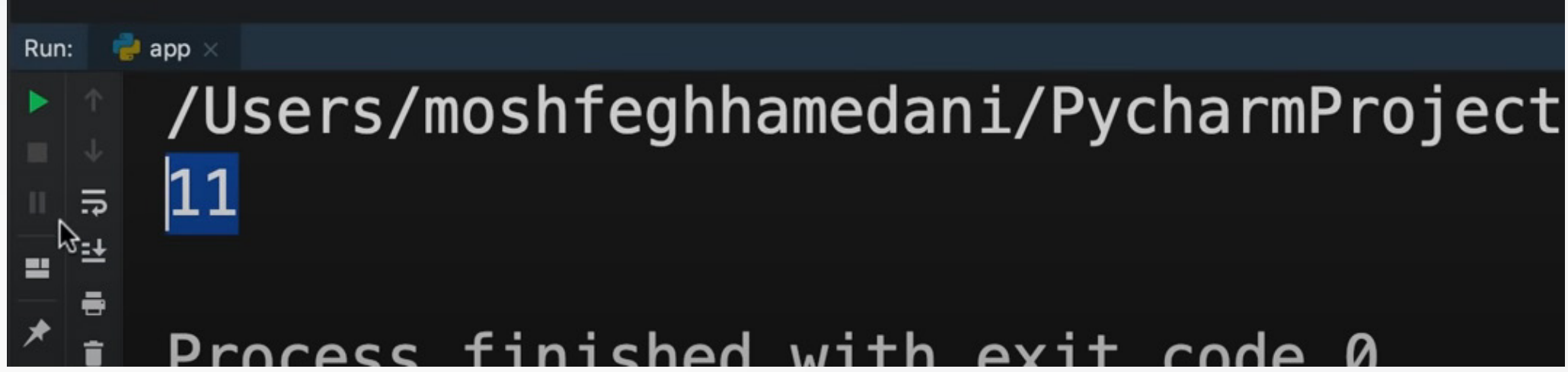
```
course = 'Python for Beginners'
print(course.find('P'))
```



```
/Users/moshfeghhamedani/PycharmProject
0
Process finished with exit code 0
```

```
course = 'Python for Beginners'
print (course.find ( 'p' ))
```

```
1 course = 'Python for Beginners'
2 print(course.find('Beginners'))
```



```
Run: app x
/Users/moshfeghhamedani/PycharmProject
11
Process finished with exit code 0
```

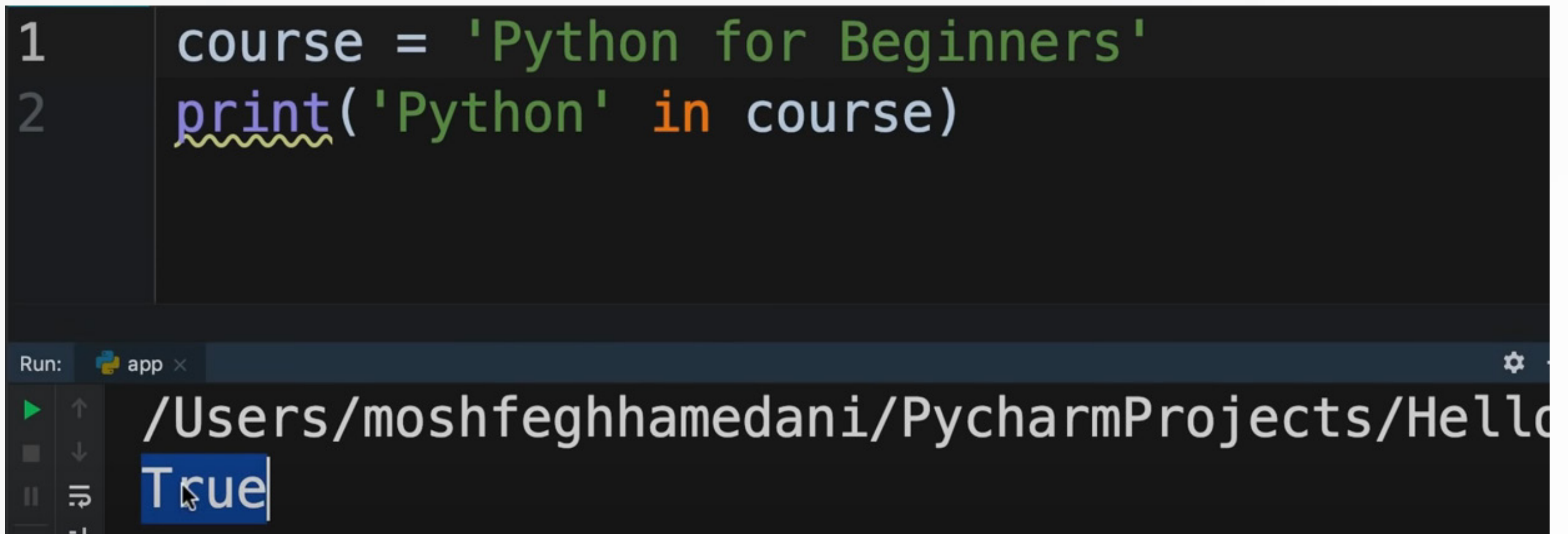
```
course = 'Python for Beginners'
print(course. find( 'Beginners'))
```

```
course = 'Python for Beginners'
print(course. replace('Beginners', 'Abs'))
```



## Boolean expression:

```
1 course = 'Python for Beginners'
2 print('Python' in course)
```



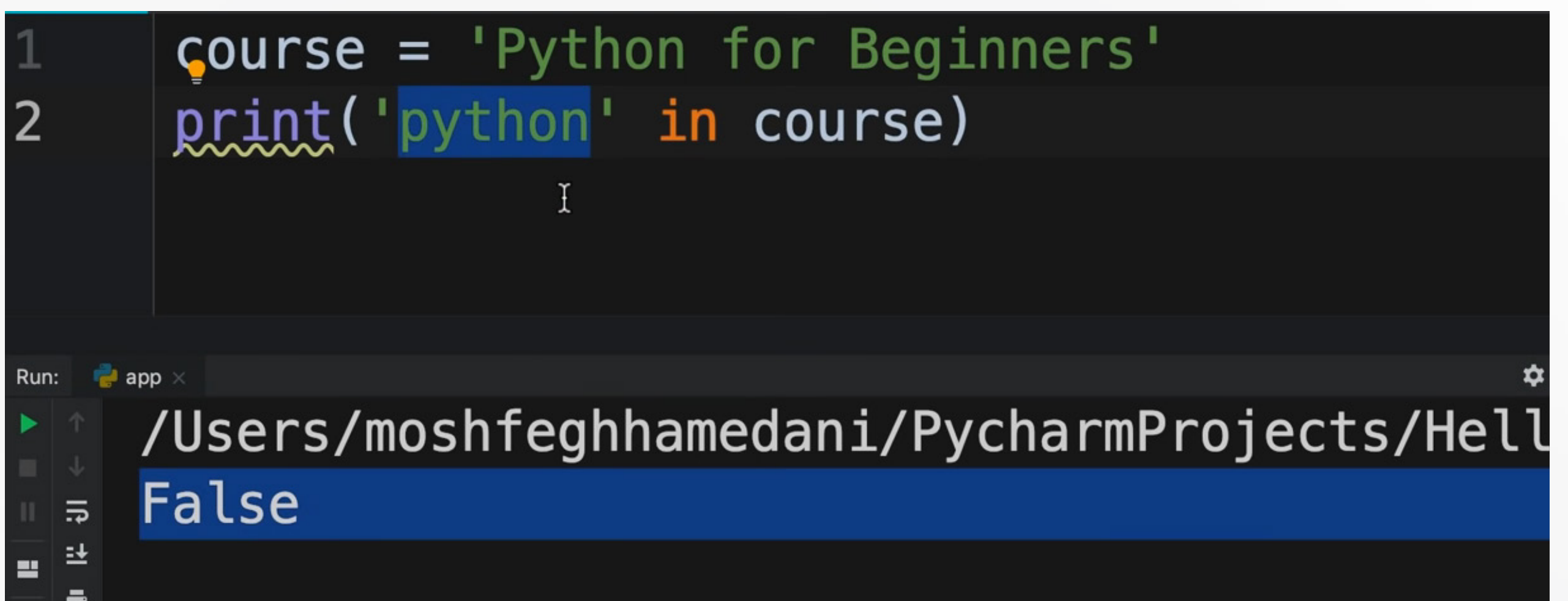
Run: app x

/Users/moshfeghamedani/PycharmProjects/Hello

True

```
course = 'Python for Beginners'
print ('Python' in course)
```

```
1 course = 'Python for Beginners'
2 print('python' in course)
```



Run: app x

/Users/moshfeghamedani/PycharmProjects/Hello

False

```
course = 'Python for Beginners'
print ('python' in course)
```

## Strings:

```
1 course = 'Python for Beginners'  
2 len()  
3 course.upper()  
4 course.lower()  
5 course.title()  
6 course.find()  
7 course.replace()  
8 ' ' in course
```

# العمليات الحسابية والمتغيرات:

(+,-,/,\*,)

Variable

```
pi@raspberrypi: ~  
Python 2.7.16 (default, Oct 10 2019, 22:02:15)  
[GCC 8.3.0] on linux2  
Type "help", "copyright", "credits" or "license" for more information.  
>>> 2+4  
6  
>>> 255+80  
335  
>>> 20-9  
11  
>>> x=5  
>>> y=8  
>>> y+x  
13  
>>> y-x  
3  
>>> print(x)  
5  
>>> w=x+y  
>>> print w  
13  
>>> e=13.5  
>>> print e  
13.5  
>>>
```

## Operator:

Operator	Description	Example
<code>+</code>	Adds two numbers	<code>2 + 3</code> returns <code>5</code>
<code>-</code>	Subtracts one number from another	<code>8 - 5</code> returns <code>3</code>
<code>*</code>	Multiplies two numbers together	<code>4 * 6</code> returns <code>24</code>
<code>**</code>	Raises the first number to the power of the second number	<code>2 ** 4</code> returns <code>16</code>
<code>/</code>	Divides the first number by the second number	<code>5 / 4</code> returns <code>1.25</code>
<code>//</code>	Divides the two numbers and rounds down to the nearest integer (divide and floor)	<code>5 / 4</code> returns <code>1</code>
<code>%</code>	Divides the first number by the second number and gives the remainder (modulo)	<code>19 % 8</code> returns <code>3</code>



## العمليات الحسابية والمتغيرات:

(+,-,/,\*,.)

Variable

```
print (10 + 3)
```

```
print (10 - 3)
```

```
print (10 * 3)
```

```
print (10 ** 3)
```

```
print (10 / 3)
```

```
print (10 // 3)
```

```
print (10 % 3)
```

# العمليات الحسابية والمتغيرات:

(+,-,/,\*,)

Variable

```
x = 10 + 3 * 2
```

```
x = 10 + 3 * 2 ** 2  
print (x)
```

parenthesis

exponentiation 2 \*\* 3

multiplication or division

addition or subtraction

## تمرین:

(+,-,/,\*,.)

Variable

$$x = (2 + 3) * 10 - 3$$

## Mathematical module:

```
x = 2.9  
print (round (x))
```

```
x = 2.9  
print (abs(-2.9))
```



## Mathematical module:

```
1 import math
2
3 math.
4 x = math.acos(x)
5 print(math.acosh(x))
   math.asin(x)
   math.asinh(x)
   math.atan(x)
   math.atan2(y, x)
```

```
1 import math
2
3 print(math.ceil(2.9))
4
```

Run: app x

/Users/moshfeghamedani/PycharmProjec

3 I

```
import math
print (math.ceil (2.9))
```

## Mathematical module:

```
1  import math
2
3  math.
4  x = math.acos(x)
5  print(math.acosh(x))
      math.asin(x)
      math.asinh(x)
      math.atan(x)
      math.atan2(y, x)
```

```
1  import math
2
3  print(math.floor(2.9))
4
Run: app x
/Users/moshfeghhamedani/PycharmProjects/
2
```

```
import math
print (math. floor (2.9))
```

## Mathematical module:

<https://docs.python.org/3/library/math.html>

## If else Statement:

```
is_hot = False
is_cold = True

if is_hot:
    print("It's a hot day")
    print("Drink plenty of water")

elif is_cold:
    print("It's a cold day")
    print("Wear warm clothes")

else:
    print("It's a lovely day")
```



## If else Statement:

```
temperature = 30

if temperature>30:
    print("It's a hot day")

else:
    print("It's not a hot day")
```

## Operator:

Operator	Description	Example
<	<code>True</code> if the first number is less than the second, <code>False</code> otherwise	<code>5 &lt; 3</code> returns <code>False</code>
>	<code>True</code> if the first number is greater than the second, <code>False</code> otherwise	<code>5 &gt; 3</code> returns <code>True</code>
<=	<code>True</code> if the first number is equal to or less than the second, <code>False</code> otherwise	<code>2 &lt;= 8</code> returns <code>True</code>
>=	<code>True</code> if the first number is equal to or greater than the second, <code>False</code> otherwise	<code>2 &gt;=</code> returns <code>False</code>
==	<code>True</code> if the first number is equal to the second, <code>False</code> otherwise	<code>6 == 6</code> returns <code>True</code>
!=	<code>True</code> if the first number is not equal to the second, <code>False</code> otherwise (not equal)	<code>6 != 6</code> returns <code>False</code>

## Example:

```
answer = "yes"
guess = input ("Is the sky blue?")
if guess == answer:
    print ("Correct!")

else:
    print("Try again")
```

## Operator:

Operator	Description	Example
<code>not</code>	Gives the opposite ( <code>True</code> becomes <code>False</code> and vice versa)	<code>x = False; not x</code> returns <code>True</code>
<code>and</code>	Returns <code>True</code> if both operands are <code>True</code> , <code>False</code> otherwise	<code>x = True; y = False; x and y</code> returns <code>False</code>
<code>or</code>	Returns <code>True</code> if either of the operands are <code>True</code> , <code>False</code> otherwise	<code>x = True; y = False; c or y</code> returns <code>True</code>



## Operator:

Operator	Description	Example
<code>&amp;</code>	Returns a 1 in each bit position for which the corresponding bits of both operands are 1 (bitwise AND)	<code>3 &amp; 5</code> returns <code>1</code>
<code> </code>	Returns a 1 in each bit position for which the corresponding bits of either or both operands are 1 (bitwise OR)	<code>3   5</code> returns <code>7</code>
<code>^</code>	Returns a 1 in each bit position for which the corresponding bits of either but not both operands are 1 (bitwise XOR)	<code>3 ^ 5</code> returns <code>6</code>
<code>~</code>	Inverts the bits in the given number (bitwise NOT)	<code>~5</code> returns <code>-6</code>
<code>&lt;&lt;</code>	Shifts the bits of the first number to the left by the number of bits specified by the second number	<code>5 &lt;&lt; 2</code> returns <code>20</code>
<code>&gt;&gt;</code>	Shifts the bits of the first number to the right by the number of bits specified by the second number	<code>5 &gt;&gt; 2</code> returns <code>1</code>

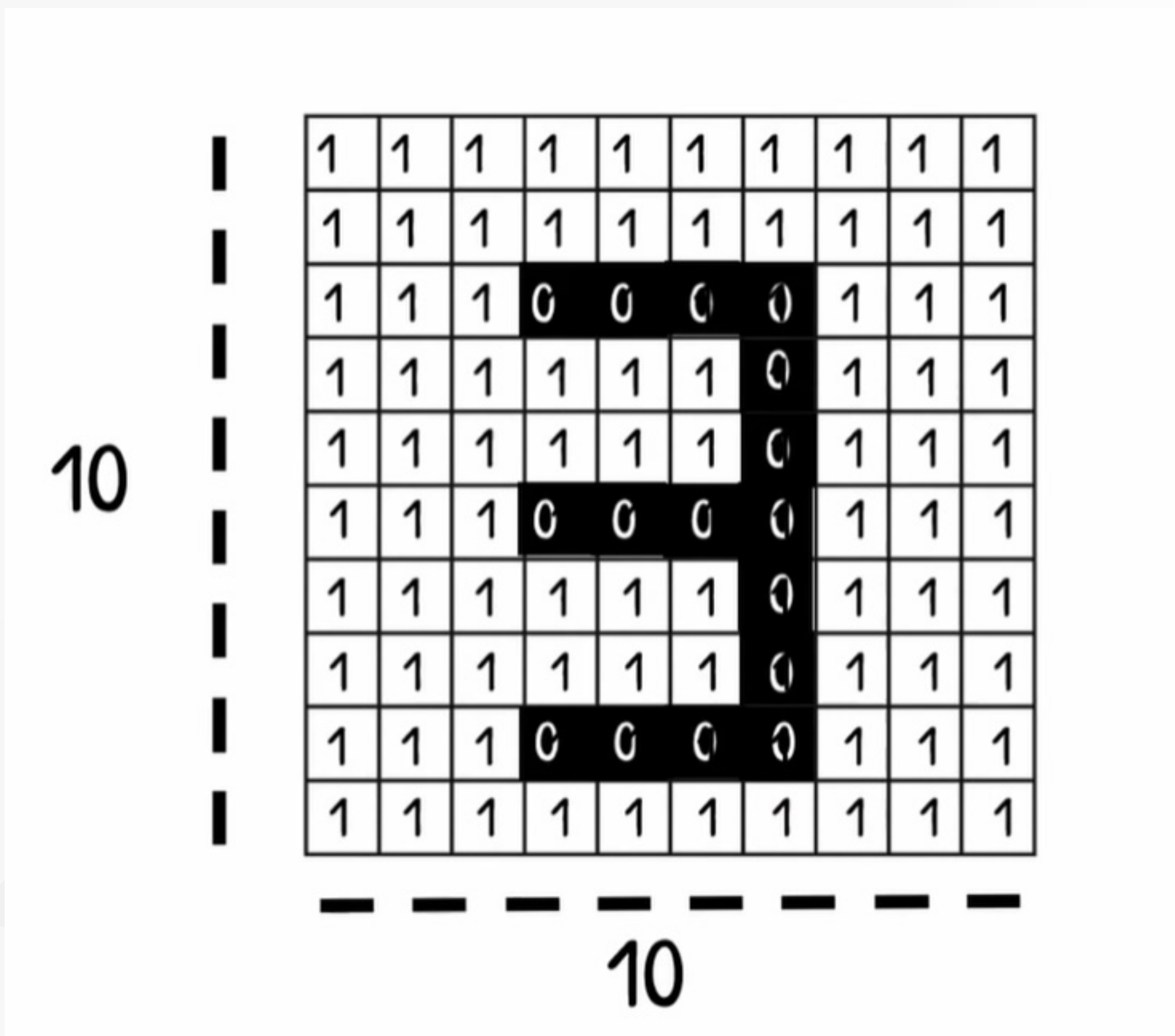
# Control:

Statement	Description	Example
<code>if</code> <code>elif</code> <code>else</code>	If a condition is true, execute the block of code underneath the <i>if statement</i> . If not, see if the condition is true in one or more <i>else if</i> ( <code>elif</code> ) statements. If one of those is true, execute the code block under that. Otherwise, execute the code block underneath the <i>else statement</i> . <code>elif</code> and <code>else</code> statements are optional.	<pre>number = 42 guess = int(input("Guess a number between 1- 100: ")) if guess == number:     print("You win!") elif guess &lt; number:     print("Nope")     print("Too low") else:     print("Nope")     print("Too high") print("Run the program to try again")</pre>

## Control:

<code>while</code>	A <i>while loop</i> executes the block of code underneath it repeatedly as long as the condition is true.	<pre>counter = 15 while counter &gt;= 5:     print(counter)     counter = counter - 1</pre>
<code>for..in</code>	Iterate over a sequence of numbers or objects. The variable declared in a <i>for loop</i> assumes the value of one of the numbers (or objects) during each iteration of the loop.	<pre>for i in range(1, 11):     print(i)</pre>
<code>break</code>	Use the <i>break statement</i> to exit out of a loop.	<pre>while True:     message = input("Tell me when to stop: ")     if message == "stop":         break     print("OK")</pre>
<code>continue</code>	The <i>continue statement</i> works similar to <i>break</i> , but instead of exiting the loop, it stops the current iteration and returns to the top of the loop.	<pre>for i in range(1, 6):     if i == 3:         continue     print(i)</pre>

## Understand Pixel:



If you want to write the number 3 you will divide the picture into small boxes then fill the boxes with black and mark it as zeros.



## Image of black and white:

Binary Image



2 Levels

0 = Black

1 = White

We called a bineray image it has only values of 0 and one and with 2 layer.

# Gray scale image



2 Levels



6 Levels



16 Levels

$$2^8 = 256$$

8 bits = 256 Levels



0

255

To make the picture have more details we will add more layer and divide the colors for 255 values and make it 255 shades of grays and we call this image a gray scale image.

Gray Scale Image



8 Bits  
OR  
256 Levels

## Colored image



By the same way we can have the colored image we divide it into 255 shade for each color.

## Familiar With this

VGA = 640 x 480  
HD = 1280 x 720  
FHD = 1920 x 1080  
4K = 3840 x 2160

480  
(height)

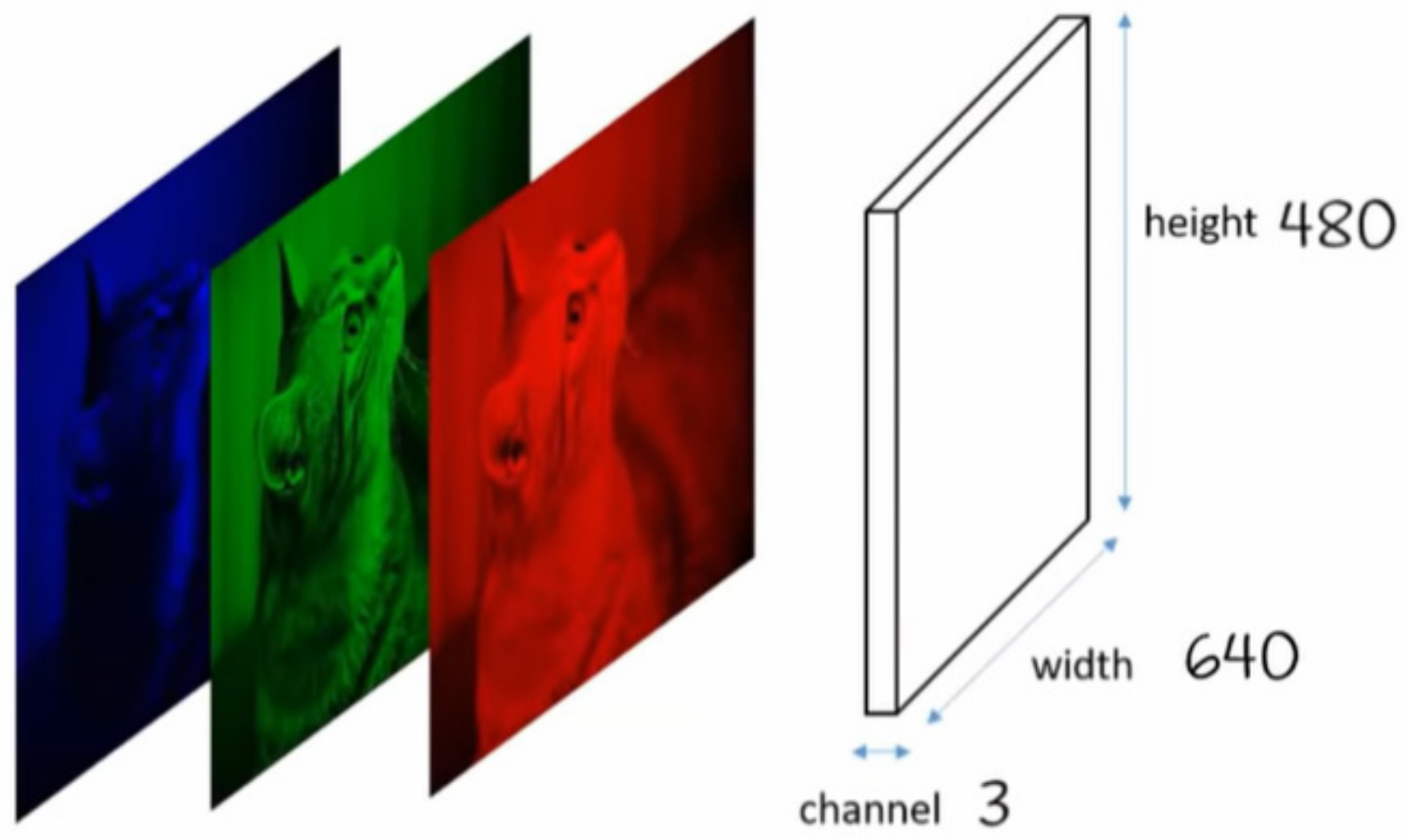
VGA

640 (width)

The diagram shows a list of resolution standards on the left. To the right, a vertical dashed line indicates a height of 480 pixels. Further right, an image of an EPSON 664 Cyan ink bottle is shown within a rectangular frame. The frame is labeled 'VGA' at the top. A horizontal dashed line at the bottom of the frame indicates a width of 640 pixels.



## Colored VGA Image



$$\text{RGB VGA} = 640 \times 480 \times 3$$

Adding these layer in top of each other will represent the full colored images.





```
import cv2
print ("package imported")
img = cv2. imread ("Resources/ 423005.jpg")
cv2. imshow ("OUtput", img)
cv2.waitKey(0)
```

- import image to opencv
- the first thing to do is to call opencv library using Import cv2 this is stand for computer vision
- to test the output and to print message to make sure its working Print ( " package imported " )
- then we use this function to read from image
- `img = cv2.imread()` and you put the path like this ("path to image"/"name of image.file type")
- to display image we use the function `imshow`
- `Cv2.imshow()` "name of the output", the image or the variable ( `img` )
- now we add delay to make the image hold before closing using `Cv2.waitKey( Millis )` if 0 its means infinite if number it in milisecond

# Import image

- Import cv2 // this is stand for computer vision
- Print ( " package imported " ) // to test the output and to declare message
- img = cv2.imread("path to image"/"name of image.file type") // this function to read from image
- // to display image we use the function imshow
- Cv2.imshow("name of the output", the image or the variable ( img )
- // now we add delay to make the image hold before closing
- Cv2.waitKey( Millis ) if 0 its means infinite if number it in milisecond

```
Import cv2
Print ( " package imported " )
Img = cv2.imread("Records/HP.png")
Cv2.imshow("Output1, img")
Cv2.waitKey(1000)
```

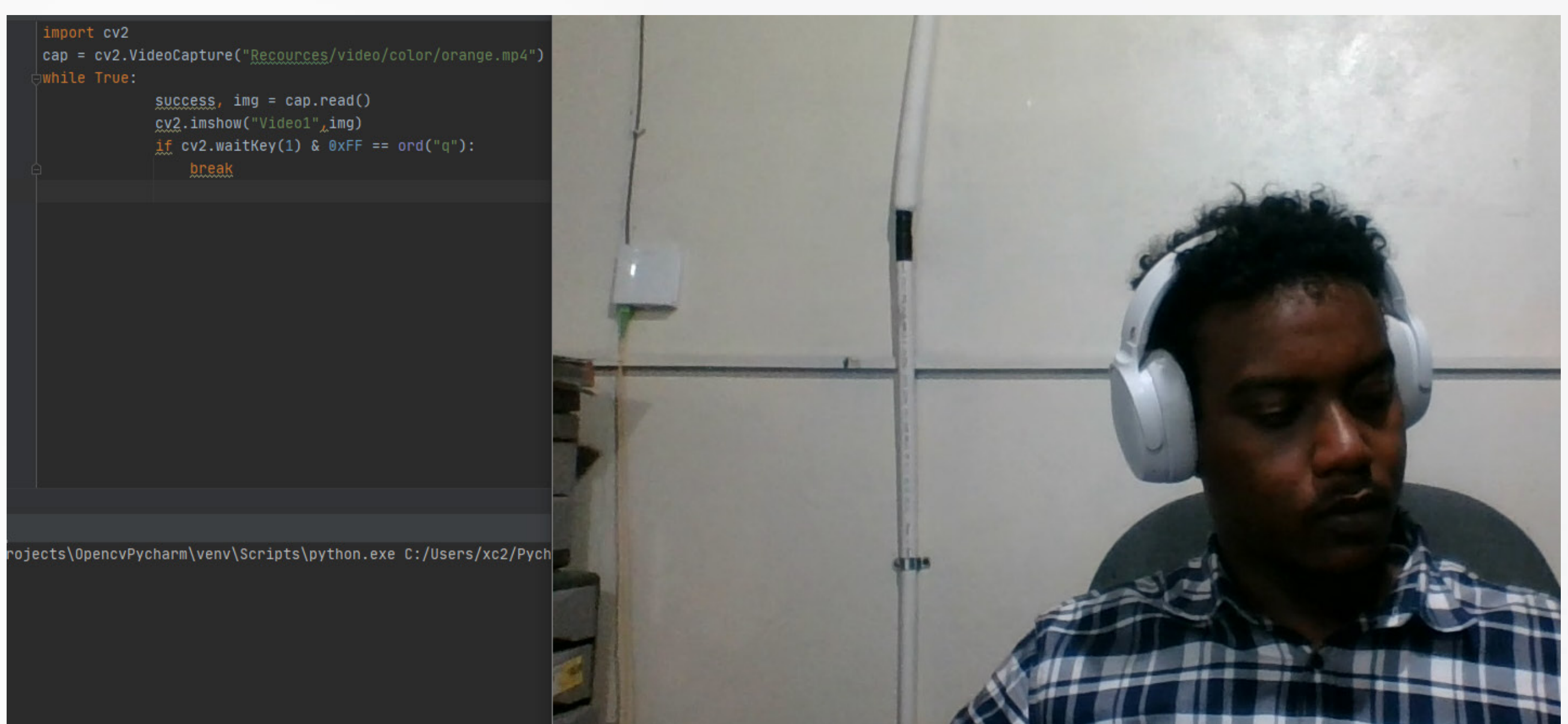
<https://hyperskill.org/curriculum>

## To import video

Video is a sequence of images we need to capture the video then play it frame by frame

```
import cv2
cap = cv2.VideoCapture("Video Path")
while True:
    success, img = cap.read()
    cv2.imshow("Video1",img)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
```

- `cap = cv2.VideoCapture("Video Path")` // `cap` is the variable and video path is where the video located
- The function is to break the video into a frames and save it in the variable `cap`
- Then we need a while loop to read the frames of the video
- And `success, img = cap.read()` this function return success if the frames was readed correctly
- If function is to break the loop after the video is played or to break with the keyboard button pressed Q





## Reading from a webcam

```
import cv2
cap = cv2.VideoCapture(camera id (0) for de-
fuolt)
Cap.set(3,640)
Cap.set(4,480)

While True:
    success, img = cap.read()
    cv2.imshow("Video1",img)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
```

Is similar to importing a video just instate of file path just writer the id of the camera , if you have one web cam or using a lapto just type 0 and it will run the default, if you have more than one webcam you can type the id number of that web cam

```
cap = cv2.VideoCapture(camera id ) // is the
camera id because im using my laptop
Cap.set ( 3,640) // we define parameter for
our webcam to make it with a specific size so
we defined the width with is id number 3
Cap.set (4,480) // id number 4 is the high
Cap.ser(10,100) // 10 is the id number for
brightness
if cv2.waitKey(1) & 0xFF == ord('q'):
    break
```



# Reading from a webcam

0. CV\_CAP\_PROP\_POS\_MSEC Current position of the video file in milliseconds.
1. CV\_CAP\_PROP\_POS\_FRAMES 0-based index of the frame to be decoded/captured next.
2. CV\_CAP\_PROP\_POS\_AVI\_RATIO Relative position of the video file
3. CV\_CAP\_PROP\_FRAME\_WIDTH Width of the frames in the video stream.
4. CV\_CAP\_PROP\_FRAME\_HEIGHT Height of the frames in the video stream.
5. CV\_CAP\_PROP\_FPS Frame rate.
6. CV\_CAP\_PROP\_FOURCC 4-character code of codec.
7. CV\_CAP\_PROP\_FRAME\_COUNT Number of frames in the video file.
8. CV\_CAP\_PROP\_FORMAT Format of the Mat objects returned by retrieve().
9. CV\_CAP\_PROP\_MODE Backend-specific value indicating the current capture mode.
10. CV\_CAP\_PROP\_BRIGHTNESS Brightness of the image (only for cameras).
11. CV\_CAP\_PROP\_CONTRAST Contrast of the image (only for cameras).
12. CV\_CAP\_PROP\_SATURATION Saturation of the image (only for cameras).
13. CV\_CAP\_PROP\_HUE Hue of the image (only for cameras).
14. CV\_CAP\_PROP\_GAIN Gain of the image (only for cameras).
15. CV\_CAP\_PROP\_EXPOSURE Exposure (only for cameras).
16. CV\_CAP\_PROP\_CONVERT\_RGB Boolean flags indicating whether images should be converted to
17. CV\_CAP\_PROP\_WHITE\_BALANCE Currently unsupported
18. CV\_CAP\_PROP\_RECTIFICATION Rectification flag for stereo cameras (note: only supported:

## Reading from a webcam



NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

## Reading from a webcam

```
import numpy as np
a= np.zeros (3)
print (a)
print (type(a))
z = np.zeros (10)
print(z)
print (z.shape)
z. shape = (10,1)
print (z)
```

## Reading from a webcam

```
] z = np.linspace(2, 10, 5) # From 2 to 10, with 5 elements  
z
```

```
] z = np.array([10, 20]) # ndarray from Python list  
z
```

```
] a_list = [1,2,3,4,5,6,7]  
z = np.array(a_list)  
z
```



## Basic Function ( convert to Grayscale )

Convert image to color spaces

```
import cv2
img = cv2.imread("Resources/423005.jpg")
imgGray = cv2.cvtColor(img,cv2.COLOR_
BGR2GRAY)
cv2.imshow("Gray image ",imgGray)cv2.wait-
Key(0)
```

```
imgGray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
// we use this function to convert the image into
gray scale we use the function callet cvt convert to
color img first this is the path for the image then the
function to convert the image .
```

## Basic Function ( convert to Grayscale )



Convert image to color spaces

```
import cv2
img = cv2.imread("Resources/423005.jpg")
imgGray = cv2.cvtColor(img,cv2.COLOR_
BGR2GRAY)
cv2.imshow("Gray image ",imgGray)cv2.wait-
Key(0)
```

Convert image to color spaces `imgGray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)` we use this function to convert the image into gray scale we use the function called `cvtColor` to convert to color `img` first this is the path for the image then the function to convert the image.

## Basic Function ( convert to Grayscale )

```
import cv2
import numpy as np
img = cv2.imread ("Resources/423005. jpg")
Kernel = np.ones ((5,5) unpuint8)
imgGray = cv2.cvtColor (img,cv2. COLOR_
BGR2GRAY)
cv2.imshow ("Gray image " imgGray)
cv2.waitKey (0)
```

## Basic Function ( Blur )

Convert image to color blur image using GaussianBlur function

```
import cv2
img = cv2.imread("Resources/423005.jpg")
imgGray = cv2.cvtColor(img,cv2.COLOR_
OR_BGR2GRAY)imgBlur = cv2.Gaussian-
Blur(imgGray, (7, 7), 0)
cv2.imshow("Gray image ",imgGray)cv2.wait-
Key(0)
```

`imgBlur = cv2.GaussianBlur(imgGray,(7,7))` // first define the image you want to blur then define the kernel value and it must be a odd number and then to define the sigma value 0.



# Reference



[www.youtube.com/watch?v=\\_uQrJOTkZlc](https://www.youtube.com/watch?v=_uQrJOTkZlc)

## Save to different format

```
import numpy as np
import cv2
img = cv2.imread('image.jpg')
img = cv2.imwrite('image.png', img)
cv2.imshow('original', ima)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

## Shifting

```
import cv2
import numpy as np
pic = cv2.imread('David.jpg')
cols = pic.shape[1]
rows = pic.shape[0]
M = np.float32([[1,0, 50], (0,1,70)])
shifted =cv2.warpAffine(pic, M, (cols, rows))
cv2.imshow('shifted', shifted)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

## blur

```
import cv2
import numpy

pic =cv2.Imread('beckham.jpg')
matrix =(7,7)
blur = cv2. GaussianBlur(pic, matrix,0)

cv2. imshow('blurred',blur)
cv2.waitKey(0)
cv2. destroyAllWindows()
```



## blur

```
import cv2
import numpy

pic = cv2.imread('horse.png')
kernal = 3
median = cv2.medianBlur(pic, kernal)
cv2.imshow('median', median)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

## blur

```
import cv2
import numpy

pic = cv2.imread('horse.png')
dimpixel=7
color = 100
space = 100
filter = cv2.bilateralFilter(pic, dimpixel,
color, space)
cv2.imshow('filter',filter)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

```
#img[0:300, 0:200]= 0,0,255
cv2.line(img, (0,0), ((img.shape[1]), (img.
shape[0])), (0,255,0), (1))
cv2.rectangle(img, (0,0), (250,300), (0,0,255), 2)
cv2.circle(img, (400,50), 30, (255,255.0),1)
cv2.putText(img, "HELLO", (150,200), cv2.FONT_HER-
SHEY_COMPLEX_SMALL, 5, (255,150,150), 2)
```

## Image stack

```
import cv2
import numpy as np
img = cv2.imread("Resources/download.jpg")
imgHor = np.hstack((img, img, img))
imgVer = np.vstack((img, img, img))
cv2.imshow("output image " imgHor)
cv2.imshow("output image " imgVer)
cv2.waitKey(0,
```



## Image perspective

```
import cv2
import numpy as np
img = cv2.imread("Resources/download.jpg")
print(img)
width,height = 600,350
pts1 =np.float32 ([[118,2],[288,106],[18,57],
[205,146]])

pts2 =np. float32 ([[0,0],[width,0],
[0,height].(width height)])
matrix = cv2.getPerspectiveTransform(pts1,
pts2)
imageOutnut = cv2.warpPerspective(img,matrix-
,(width height))

#118,2 288,106 18,57 205,164

imaGray. = cv2.cvtColor(img, cv2.COLOR_
BGR2GRAY)
cv2.imshow("Gray image",imagGray)
cv2.imshow("output image " imageOutnut)
cv2.waitKey(0)
```

# Color Dtetection 1

```
import cv2
path = "Recources/data/corossion/Picture1.png"
img = cv2.imread(path)
imsHSV = cv2.cvtColor(img,cv2.COLOR_BGR2HSV)
cv2.imshow("img Images",img)
cv2.imshow ("Stacked Images",imgHSV)

cv2.waitKey(0)
```

## Color Dtection 2

```
import cv2

path = "Recourses/data/coression/Picture1.png"
cv2.namedWindow("Trackbars")
cv2.resizeWindow("Trackbars", 640, 240)
img = cv2.imread (path)
imeHISV = cv2.cvtColor(ims, cv2.COLOR _BGR2HSV)
cv2.imshow("img Images", img)
cv2.imshow("Stacked Images", lmgHSv)
cv2.waitKey(0)
```

## Color Dtection 3

```
import cv2
def empty():
    pass

path = "Recources/data/corossion/Picture1.png"
cv2.namedWindow("Trackbars")
cv2.resizeWindow("Trackbars", 640, 240)
cv2.createTrackbar("Hue min", "Track-
bars", 0, 179, empty)

img = cv2.imread(path)

imgEHSV = cv2.cvtColor (img, cv2.COLOR_
BGR2HSV)
cv2.imshow ("img Images", ima)
cv2.imshow ("Stacked Images", imgHSv)
cv2.waitKey(0)
```

## Color Dtetection 4

```
import cv2
def empty(a):
    pass

path = "Recources/data/corossion/Picture1.png"
cv2.namedWindow("TrackBars")
cv2.resizeWindow("TrackBars", 640,240)
cv2.create Trackbar("Hue min", "TrackBars",
0,179, empty)

cv2.create Trackbar("Hue max", "TrackBars",
179,179,empty)

cv2.createTrackbar("Sat min", "TrackBars",
0,255,empty)

cv2.createTrackbar("Sat max", "TrackBars",
255, 255,empty)

cv2.create Trackbar("Val min", "TrackBars"
0,255,empty)

cv2.create Trackbar("Val max", "TrackBars",
255,255, empty)

ime = cv2.imread (path)

imeHSV = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
cv2.imshow("img Images", img)
C2.imshow("Stacked Images", imeHSY)

cv2.waitKey(0)
```



## Color Dtection 4.1

```
import cv2
def empty(a):
    pass

path = "Recources/data/corossion/Picture1.png"
cv2.namedWindow("TrackBars")
cv2.resizeWindow("TrackBars", 640, 240)
cv2.createTrackbar("Hue min", "TrackBars", 0, 179, empty)
cv2.createTrackbar("Hue max", "TrackBars", 179, 179, empty)
cv2.createTrackbar("Sat min", "TrackBars", 0, 255, empty)
cv2.createTrackbar("Sat max", "TrackBars", 255, 255, empty)
cv2.createTrackbar("Val min", "TrackBars", 0, 255, empty)
cv2.createTrackbar("Val max", "TrackBars", 255, 255, empty)

img = cv2.imread(path)
while True:
    imgHSY = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
    h_min = cv2.getTrackbarPos("Hue min", "TrackBaus")
    h_max = cv2.getTrackbarPos("Hue max", "TrackBaus")
    s_min = cv2.getTrackbarPos("Sat min", "TrackBars")
    S_Max = cv2.getTrackbarPos("Sat max", "TrackBars")
    v_min = cv2.getTrackbarPos("Val min", "TrackBars")
    v_Max = cv2.getTrackbarPos("Val max", "TrackBacs")
    print(h_min, h_max, s_min, s_max, v_min, v_max)

imgHSV = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
cv2.imshow("img, Images", img)
cv2.imshow("Stacked Images", imgHSY)

cv2.waitKey(1)
```

## Color Detection 4.1

```
lower = np.array([h_min, s_min, v_min])
upper = np.array ([h_max, s_max, v_max])
mask = cv2.inRange (imgHSV, lower, upper)

mask = cv2.inRange (imgHSV, lower, upper)
imgResult = cv2.bitwise_and
(img, img, mask=mask)
cv2.imshow ("Original", img)
```

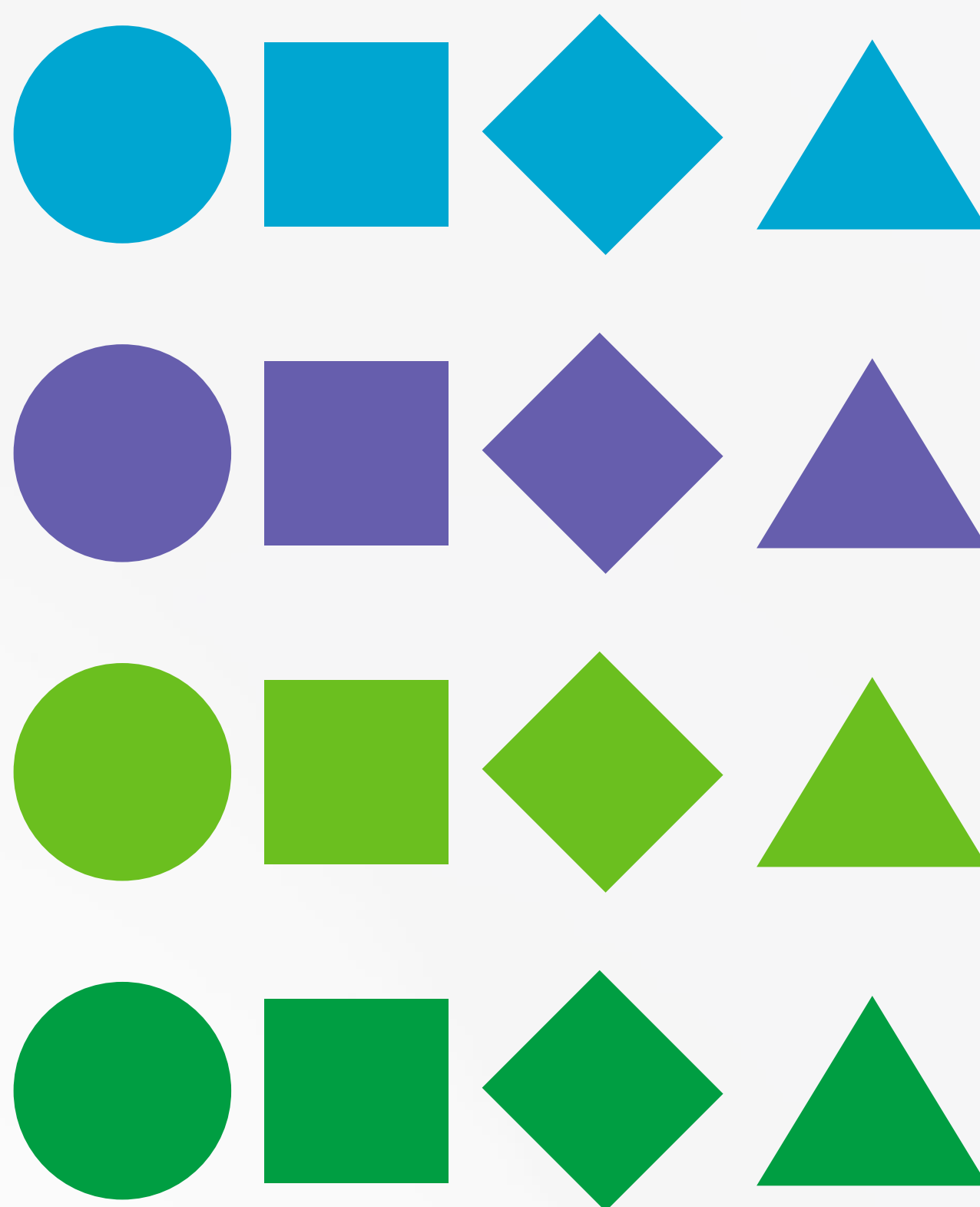
## Countor and shape 1

```
import cv2
import numpy as np
path = "Recources/data/corossion/Picture1.png"
img = cv2.imread(path)
cv2.imshow("Stacked Images", img)
cv2.waitKey(0)
```

## Countor and shape 2 then add stack

```
import cv2
import numpy as np
path = "Recources/VMS051904Sec1-21.gif.crdownload"
img = cv2.imread(path)
imgGray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
imgBlur = cv2.GaussianBlur(imgGray,(7,7),1)

cv2.imshow("normal Images", img)
cv2.imshow("gray Images", img)
cv2.imshow("blur Images", img)
cv2.waitKey(0)
```



## Countor and shape then add canny and stack it and add blank images

```
imgCanny = cv2.Canny(img,150,200)
imgCanny = cv2.Canny(img,150,200)
imgBlank = np.zeros_like(img)
```



## Create function

```
def getContours(img):  
    contours,hierachy =cv2.findContours (img,cv2.RETR_EXTER-  
NAL,cv2.CHAIN_APPROX_NONE)  
    for cnt in contours:  
        area = cv2.contourArea(cnt)  
        print(area)  
  
getContours (imgCanny)
```

## Copy img countour and draw countor

```
def getContours(img):  
    contours,hierachy =cv2.findContours(img,cv2.RETR_EXTERNAL,  
    cv2.CHAIN_APPROX_NONE)  
    for cnt in contours:  
        area = cv2.contourArea(cnt)  
        print(area)  
        cv2.drawContours(imgCountour,cnt,-1,(255,0,0),3)  
  
imgCountour = img.copy()
```

## Condition if to eliminate some contours + area and use length to count it

```
def getContours(img):  
    contours,hierachy =cv2.findContours(img,cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_NONE)for cnt in contours:  
        area = cv2.contourArea(cnt)  
        print(area)  
  
        if area>500:  
            cv2.drawContours(imgCountour, cnt, -1, (255, 0, 0),  
3)  
            peri = cv2.arcLength(cnt,True)  
            print(peri)  
  
            approx = cv2.approxPolyDP(cnt, 0.02 * peri, True)  
            print(len(approx))  
            objCor = len(approx)
```

## Get bounding box and draw it

```
approx = cv2.approxPolyDP(cnt, 0.02 * peri, True)
print(len(approx))
objCor = len(approx)

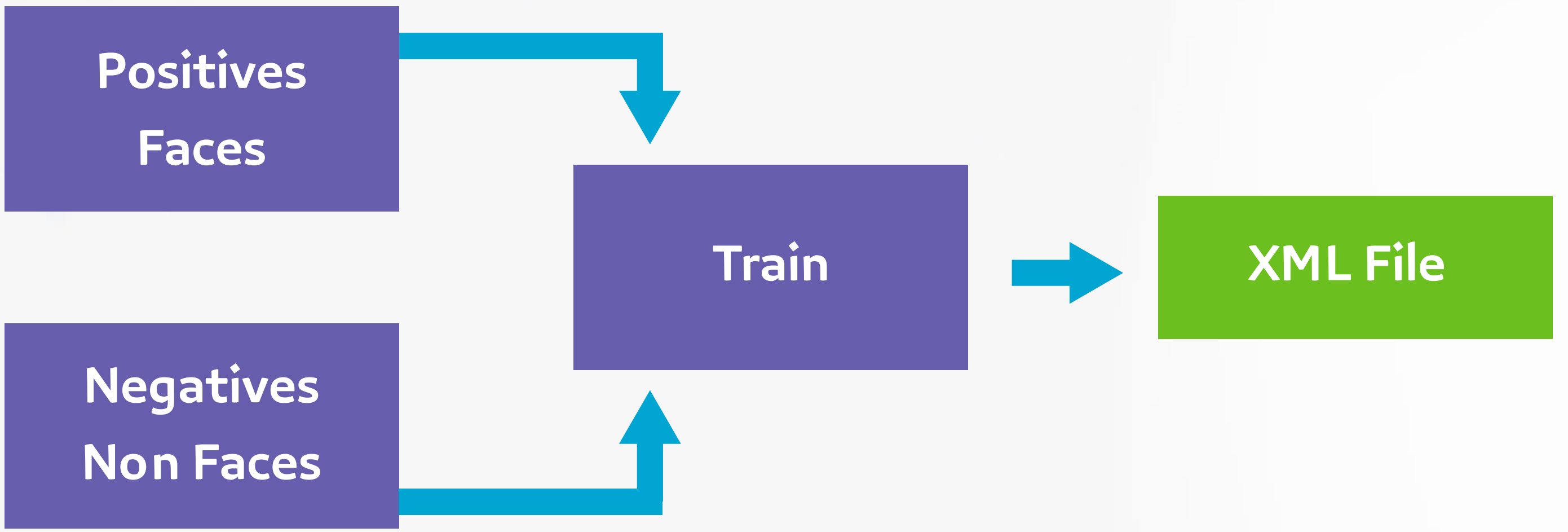
x, y, w, h = cv2.boundingRect(approx)
cv2.rectangle(imgCountour, (x, y), (x + w, y + h), (0, 255, 0),
2)
```

## Git bounding box and draw it

```
print(len(approx))
objCor = len(approx)
if objCor == 3:
    ObjectType = "Orange color Detected"
else:
    ObjectType = "none"
x, y, w, h = cv2.boundingRect(approx)
cv2.rectangle(imgContour, (x, y), (x + w, y + h), (0,
255, 0), 2)
cv2.putText(imgContour, ObjectType, (x + (w // 2) -
10, y + (h // 2) - 10), cv2.FONT_HERSHEY_DUPLEX, 0.7,
(0, 255, 255), 3)
```



**Ai**



## حدود المسؤولية

تقدم "منشآت" المصادر التعليمية، وهي خدمة من خدمات مكتبة مركز نكاء التي تقدمها "منشآت"، والتي تساهم وتساعد في إثراء المحتوى العربي لمصادر التعلم عبر الإنترنت؛ لتوفير المعرفة لفئات مختلفة في مجالات التقنية وريادة الأعمال، ولا تقدم "منشآت" أو من يمثلها أي قرارات أو ضمانات سواءً بشكل صريح أو ضمني حول اكتمال، أو دقة، أو موثوقية، أو ملاءمة، أو توافر هذه البيانات، أو المعلومات، أو المواد ذات الصلة الواردة في الكتيب لأي غرض كان، ولا يجوز استخدامها لغرض آخر غير الاستخدام العام، ولا تتحمل "منشآت" أو من يمثلها - بأي حال من الأحوال - أي أضرار مادية أو معنوية، مباشرة أو غير مباشرة قد تحصل، وتؤكد "منشآت"، أو من يمثلها أنها غير مسؤولة سواءً بشكل كامل أو جزئي عن أي ضرر مباشر أو غير مباشر، عرضي أو تبعي أو عقابي خاصاً كان أو عاماً، كما أنها غير مسؤولة عن أي فرصة ضائعة أو خسارة أو ضرر من أي نوع، ومنها على سبيل المثال لا الحصر: أي ضرر أو فيروس قد يتعرض له الحاسوب الشخصي نتيجة الدخول إلى هذه الصفحة، وأن "منشآت" أو من يمثلها تبذل الجهد للتأكد من أن المعلومات المتوفرة من خلال المصادر التعليمية شاملة ودقيقة قدر المستطاع. كما وتؤكد "منشآت" على الالتزام بحقوق النشر، وحقوق الملكية الفكرية لمحتويات المصادر التعليمية، بما في ذلك شعار "منشآت"، ولا يحق نشر أي معلومات أو رأي يتم التعبير عنه هنا دون الحصول على إذن خطي مسبق للقيام بذلك من قبل "منشآت".

مركز ذكاء

منشآت  
monsha'at  
لهيئة العامة للمنشآت الصغيرة والمتوسطة  
Small & Medium Enterprises General Authority

Thakaa.sa