**✅ What Is OCR in Python?**

* **Definition**: Converting text from images into editable, machine‑encoded text using Python. ([codezup.com](https://codezup.com/python-ocr-tutorial-optical-character-recognition-guide/?utm_source=chatgpt.com), [pythoncentral.io](https://www.pythoncentral.io/create-ocr-with-pytesseract/?utm_source=chatgpt.com))
* **Key tool**: pytesseract — a Python wrapper around Google’s Tesseract OCR engine. ([askpython.com](https://www.askpython.com/python/examples/optical-character-recognition?utm_source=chatgpt.com))

**🛠️ Setup & Code Example**

pip install pytesseract opencv-python pillow

# On Linux: sudo apt install tesseract-ocr

import cv2

import pytesseract

img = cv2.imread('sample.jpg')

text = pytesseract.image\_to\_string(img, lang='eng')

print(text)

This reads an image and outputs detected text. ([askpython.com](https://www.askpython.com/python/examples/optical-character-recognition?utm_source=chatgpt.com))

**🧠 Why Preprocessing Matters**

* Convert to grayscale, apply thresholding or median blur before OCR to improve accuracy. ([codezup.com](https://codezup.com/python-ocr-tutorial-optical-character-recognition-guide/?utm_source=chatgpt.com))
* Example with Pillow:

from PIL import Image, ImageFilter, ImageEnhance

img = Image.open('noisy.jpg').convert('L')

img = img.filter(ImageFilter.MedianFilter())

img = ImageEnhance.Contrast(img).enhance(2)

text = pytesseract.image\_to\_string(img)

([geeksforgeeks.org](https://www.geeksforgeeks.org/introduction-to-python-pytesseract-package/?utm_source=chatgpt.com))

**📦 Python OCR Ecosystem**

| **Tool** | **Strengths** | **Use Case** |
| --- | --- | --- |
| **pytesseract** | Easy, supports 100+ languages | Classic printed text |
| **EasyOCR** | Deep‑learning, handles scene text | Text in natural images ([geeksforgeeks.org](https://www.geeksforgeeks.org/introduction-to-python-pytesseract-package/?utm_source=chatgpt.com), [pythoncentral.io](https://www.pythoncentral.io/create-ocr-with-pytesseract/?utm_source=chatgpt.com), [builtin.com](https://builtin.com/data-science/python-ocr?utm_source=chatgpt.com)) |
| **Keras‑OCR** | Customizable, trainable models | Advanced/custom domains |

**⚙️ Advanced Features**

* Obtain bounding boxes, word‑level info, and configure segmentation modes (--psm). ([github.com](https://github.com/NanoNets/ocr-with-tesseract?utm_source=chatgpt.com))
* PDF workflows: convert PDF → images → OCR via pdf2image + pytesseract. ([vitalflux.com](https://vitalflux.com/python-tesseract-pdf-ocr-example/?utm_source=chatgpt.com))

**🌍 Real‑World Applications**

* Automate invoice, receipt, and form data entry. ([datacamp.com](https://www.datacamp.com/tutorial/optical-character-recognition-ocr-in-python-with-pytesseract?utm_source=chatgpt.com))
* Enable accessibility (convert printed text to speech/Digital to aid visually impaired).
* Deploy in mobile scanners, license‑plate readers, and document digitization tools.

Available Python OCR Libraries

Now that we have understood OCR and its use let us look at some commonly used open-source Python libraries for text recognition and extraction.

1. **Pytesseract** – Also called ‘Python-tesseract,’ it is an OCR tool for Python that works as a wrapper for the Tesseract-OCR Engine. This library can read all image types (.jpeg,.png, .gif, .bmp, .tiff, etc.) and recognize text in images. Hence, it is commonly seen in use cases for OCR image-to-text conversion.
2. **EasyOCR –** Another quite popular Python library is EasyOCR. As the name suggests, the library is designed for beginners and is easy to use. It is a general OCR Python module that supports more than 80 languages and can read natural scene and dense text in documents. Once installed, users need to initialize only two classes – one reader() and another through the readtext() function for reading the text in the image.
3. **Keras-OCR –**This is an equally powerful open-source library like the two libraries mentioned above. The Keras-OCR library provides a high-level API and end-to-end training pipeline to build new OCR models. In the next section, we will see a step-by-step tutorial using the Keras-OCR to extract text from multiple images. You can find the documentation [**here**](https://keras-ocr.readthedocs.io/en/latest/).