

**Yogesh Haribhau Kulkarni**

• You

...

AI Advisor (Helping organizations in their AI journeys) | PhD (Geometric Modeling) | Tech Colum...
now •

🚀 Are you looking to leverage your own/company's documents to build AI apps? The challenge often lies in processing documents with complex layouts, tables, pictures, etc .

Meet **Docling**, a free, opensource tool from **IBM** that simplifies document processing for the generative AI era. It handles diverse formats like PDFs, DOCX, and even images.

What makes Docling special is that it preserves the original layout, tables, and reading order, ensuring no critical information is lost.

All documents are converted into a unified structure, creating a clean and consistent data source for your AI models.

A powerful application of this is building a Retrieval-Augmented Generation (RAG) system. 🧠 With Docling, you can parse your documents, then use LlamaIndex to build a searchable knowledge base. Your language model can then find relevant information and answer specific questions about your content. Imagine asking your technical reports a question and getting a precise, context-aware answer in seconds!

Docling integrates seamlessly with popular AI frameworks like LlamaIndex and LangChain. For added peace of mind, it can run entirely locally, which keeps your sensitive data private and secure. 🔒

#DocumentAI #GenerativeAI #RAG #LLM #LlamaIndex #DataProcessing #AI #Tech #Innovation

The screenshot displays a JupyterLab environment. On the left, the 'EXPLORER' panel lists files and folders, including 'data', 'NVIDIAIAA.pdf', 'Systems Thinking Four Key Questions...', 'Systems Thinking in 25 Words or Less...', 'scratch', '1.21.0', 'AI-Engineer-Skool_booktutor.py', 'docling_book_loader.py', 'docling_examples_custom_conversion.py', 'docling_examples_ragwithllamaindex.py', 'docling_examples_simple_conversion.py' (selected), 'docling_examples_table_export.py', 'docling_ipynb', 'mayurji_docling-basic.ipynb', 'mayurji_docling-langchain.ipynb', 'README.md', 'requirements_alternative.txt', 'requirements.txt', 'setup_script.py', 'streamlit_app.py', and 'sudarshan_koirala_docling.ipynb'. The main editor shows the code for 'docling_examples_simple_conversion.py', which imports 'DocumentConverter' from 'docling.document_converter', sets a source URL from 'https://arxiv.org/pdf/2408.09869', converts it to a document, and exports it to markdown. The bottom status bar indicates the current file is 'docling_examples_simple_conversion.py' and the language is 'Python 3.10.18 (miniconda)'. The right sidebar shows the 'TERMINAL' panel with the output of the script, which is a technical report for Docling version 1.0, authored by Christoph Auer, Maksym Lysak, Ahmed Nassar, Michele Dolfi, Nikolaos Livathinos, Panos Vagenas, Cesar Berrospi, Ramis Matteo Omenet, Fabian Lindbauer, Kasper Dinkla, Lokesh Mishra, Yusik Kim, Shubham Gupta, Rafael Teixeira, de Lima Valery Weber, Lucas Morin, and Ingmar Meijer.