1. 创建虚拟环境
   1. python -m venv venv
2. 激活虚拟环境
   1. venv\Scripts\activate
3. 安装docling、pytorch（cpu版torch)
   1. pip install docling --extra-index-url <https://download.pytorch.org/whl/cpu>
4. 配置模型
   1. 下载docling依赖的5个模型
      1. 激活虚拟环境，命令行运行docling-tools models download
      2. 模型下载与配置的官方教程链接：<https://docling-project.github.io/docling/usage/advanced_options/>
      3. 没有下载成功的，从以下五个下载链接里面手动下载
         1. <https://huggingface.co/ds4sd/docling-layout-heron>
         2. <https://huggingface.co/ds4sd/docling-models>
         3. <https://huggingface.co/ds4sd/DocumentFigureClassifier>
         4. <https://huggingface.co/ds4sd/CodeFormulaV2>
         5. <https://www.jaided.ai/easyocr/modelhub/?utm_source=chatgpt.com> （要同时下载语言检测模型和语言识别模型）
      4. 把models文件夹复制粘贴到项目中，在使用docling库的py项目导入模型的环境路径变量
         1. os.environ["DOCLING\_ARTIFACTS\_PATH"] = r"D:\RAG\_docling\models"
5. 扩展成RAG
   1. 下载一个轻量级向量模型：<https://hf-mirror.com/sentence-transformers/all-MiniLM-L6-v2/tree/main>
   2. 向量模型被两次使用，第一次用于分词计数，第二次用于存储和查询的向量化，两次向量模型的使用配置为本地向量模型，具体方式参考示例代码。特别是分词计数的本地向量模型配置，参考官方示例：<https://docling-project.github.io/docling/examples/hybrid_chunking/?utm_source=chatgpt.com#configuring-tokenization>
   3. 构建这个RAG用的是langchain框架和chromadb向量数据库，全部要安装的库有
      1. python -m pip install sentence-transformers
      2. pip install -U langchain langchain-core langchain-community
      3. pip install langchain-docling
      4. pip install -U langchain-huggingface
      5. pip install -U chromadb
      6. pip install -U langchain-openai openai
   4. 代码有三个文件
      1. Docling\_test.py是测试docling的使用的
      2. Docling\_RAG.py是RAG的主体代码
      3. RAG\_utilis.py是用于将docling解析的文档的元数据扁平化成chroma数据库元数据可以使用的扁平化格式
6. 项目文档结构

├── RAG\_docling/

├── docling\_RAG.py

├── docling\_test.py

├── docling安装使用文档.docx

├── launch.bat

├── rag\_utils.py

├── data/

├── pdf/

├── Docling 项目深度分析报告.pdf

├── models/

├── all-MiniLM-L6-v2/

├── config.json

├── model.safetensors

├── modules.json

├── special\_tokens\_map.json

├── tokenizer.json

├── tokenizer\_config.json

├── vocab.txt

├── 1\_Pooling/

├── config.json

├── ds4sd--CodeFormulaV2/

├── .gitattributes

├── added\_tokens.json

├── chat\_template.json

├── config.json

├── generation\_config.json

├── model.safetensors

├── preprocessor\_config.json

├── processor\_config.json

├── README.md

├── special\_tokens\_map.json

├── tokenizer.json

├── tokenizer\_config.json

├── vocab.json

├── ds4sd--docling-layout-heron/

├── .gitattributes

├── config.json

├── model.safetensors

├── preprocessor\_config.json

├── README.md

├── ds4sd--docling-models/

├── .gitattributes

├── .gitignore

├── config.json

├── README.md

├── model\_artifacts/

├── tableformer/

├── accurate/

├── tableformer\_accurate.safetensors

├── tm\_config.json

├── fast/

├── tableformer\_fast.safetensors

├── tm\_config.json

├── ds4sd--DocumentFigureClassifier/

├── .gitattributes

├── config.json

├── model.safetensors

├── README.md

├── EasyOcr/

├── craft\_mlt\_25k.pth

├── english\_g2.pth

├── japanese\_g2.pth

├── kannada.pth

├── korean\_g2.pth

├── latin\_g2.pth

├── telugu.pth

├── zh\_sim\_g2.pth

├── scratch/

├── Docling 项目深度分析报告.doctags

├── Docling 项目深度分析报告.json

├── Docling 项目深度分析报告.md

├── Docling 项目深度分析报告.txt

├── vectorstores/

├── chroma/

├── venv/