Poipoi's Agent Tutorial

一、Python 环境配置

1. 为什么选择 uv?

现代 Python 开发推荐使用 uv 作为包管理工具：

优势：

- **快得飞起** - 用 Rust 编写，安装速度比 pip/conda 快 10-100 倍

- **无商用风险** - Conda 在超过 200 人的组织有[潜在商用授权问题](https://www.anaconda.com/blog/is-conda-free)

安装 uv：

代码块

```
# macOS/Linux
curl -LsSf https://astral.sh/uv/install.sh | sh

# Windows
powershell -ExecutionPolicy ByPass -c "irm https://astral.sh/uv/install.ps1 | iex"
```

配置镜像源（提升下载速度）：

代码块

```
# macOS/Linux
mkdir -p ~/.config/uv
cat > ~/.config/uv/uv.toml << EOF
[[index]]
url = "https://mirrors.tuna.tsinghua.edu.cn/pypi/web/simple"
default = true
EOF

# Windows
# 设置环境变量:
UV_DEFAULT_INDEX=https://mirrors.tuna.tsinghua.edu.cn/pypi/web/simple
```

二、Agent 路线

1. 理解Agent相关概念（耗时两天）

- 文档：https://the-pocket.github.io/PocketFlow/
- 需要把文档里面的 Home、Core Abstraction、Design Pattern、Utility Function 都看一遍（有兴趣能看看Agentic Coding）

2. 实现一些简单的Agent（耗时一天）

- 克隆 https://github.com/The-Pocket/PocketFlow
- 尝试阅读和运行里面的“Below are basic tutorials”里面的代码
- 你可能需要llm api，推荐智谱https://open.bigmodel.cn（量大管饱）或者kimi https://platform.moonshot.cn/docs/overview
- 你现在懂了Agent是怎么写出来的，下一步你应该去理解100行源码
- 此时你可以查看 https://www.youtube.com/watch?v=0Zr3NwcvpA0&t=1145s 来理解100行源码

3. Vibe coding实现一个复杂的Agent（耗时一天）

- 你需要claude code（cursor也行，这里仅展示claude code教程）和llm api
- claude code安装方式为https://platform.moonshot.cn/docs/guide/agent-support
- Llm api推荐智谱https://open.bigmodel.cn（量大管饱）或者kimi https://platform.moonshot.cn/docs/overview
- 最后按照视频教程敲一遍 https://www.youtube.com/watch?v=HH7TZFgoqEQ
- 把项目写简历上，推荐免费开源简历网站https://rxresu.me
- 在面试时候不仅能吹项目足够复杂，还能吹自己会vibe coding

4. 理解为什么不用langchain或google-adk之类的大型框架（耗时1分钟）

- 出现了各种各样的agent框架：LangChain、Haystack、Google ADK、Dify、Coze等等

但它们都有各种问题共同问题：

1. **过度抽象** - 为了通用性包装了太多层，简单任务也要写大量代码

2. **依赖地狱** - 依赖包版本冲突频繁，pip install 经常失败

3. **Bug 多** - 生态不成熟，遇到问题难以追溯和修复

导致的实际问题：

1. ❌ 新人入门困难（即使从 ADK 跳到 LangChain 也难）
2. ❌ 部署时依赖冲突频繁
3. ❌ 遇到 Bug 难以定位和修复
4. ❌ 不灵活，难以定制

- 推荐用PocketFlow或者自建轮子直接使用 LLM API + 手写工具调用逻辑

5. 恭喜你已经学会了Agent所有知识

二、RAG/Vector DB 路线

1. 选择Chroma（耗时1小时）

- https://docs.trychroma.com/docs/overview/getting-started
- 注意，你可能需要一个embedding model api而不是llm api，你可以在智谱、阿里云百炼等官网找到对应的embedding模型

2. 为什么选Chroma

- ✅ 轻量级，嵌入式部署简单
- ✅ API 简洁，学习成本低
- ✅ 适合中小规模应用

3. 为什么不选Milvus、LanceDB、VectorDB等

- **Milvus和LanceDB** - 过于重量级，部署复杂
- **pgvector** - 需要维护 PostgreSQL 实例，且需要会写复杂的sql

4. 这就是RAG的全部，恭喜你已经学会了RAG

- 为什么这就是全部了，真有这么简单？是的这就是全部，当初提出RAG(Retrieval-Augmented Generation)概念时，可能觉得得有 检索-增强-生成 这三个功能。
- 但实际上大伙最终只用到了检索，VectorDB就能很完美执行这个任务
- 所以RAG是个很过时的概念，大伙只想要一个VectorDB而已，或者说RAG=VectorDB

三、行业现状与建议

1. Agent 应用商业化

现状：

- 大部分 Agent 应用投入非常非常高的离谱，难以回本
- 真正好用的只有 **Cursor** 和 **Claude Code**（编程助手场景）

2. 招聘与就业

实习

- 很好投，基本是个人都要，但强烈推荐去核心部门别去小作坊
- 建议实习时自己学习pytorch sft rlhf的东西，只会Context Engineering是秋招找不到工作

秋招

- 如果实习做的东西是纯Context Engineering，秋招会很大问题
- 秋招招聘普遍要求会算法RL，纯Context Engineering会投不进
- 个人认为Agent工程应该算是算法序列，薪资可能会很高

3、未来方向

- 强化学习：纯Context Engineering的效果太烂了，现在是直接提升模型效果

四、vibe coding

- 推荐学习vibe coding，推荐使用cursor和claude code(可以看moonshot的怎么用claude code教程)现在大厂都在强烈推荐ai生成代码，甚至还有指标要求提交的代码里面ai的比例要达到多少。
- vibe coding教程（30分钟）https://the-pocket.github.io/PocketFlow/guide.html 和 https://www.youtube.com/watch?v=wc9O-9mcObc