## Metagpt day1

Metagpt 推荐步骤安装：

1.下载miniconda

2.配置虚拟环境，安装python3.9+ 和 metagpt 和jupyter notebook，

命令：

conda creat -n metagpt python3.10

conda activate metagpt

pip install metagpt

3.再项目文件夹里新建文件夹config，新建文件config.yaml或者key.yaml

推荐注册智谱的api，在config.yaml文件里写入ZHIPUAI\_API\_KEY: "你的api"

4.打开jupyter notebook，命令为jupyter

5.新建ipynb文件，按教程里的步骤运行代码即可

人工智能现代方法第41页的智能体内容和metagpt的智能体理论很相似，以下是我的笔记

我们在本书中设计的智能体程序都有相同的框架：它们将当前感知作为传感器的输入，并将 动作返回给执行器。

>智能体程序框架还有其他选择。例如，我们可以让智能体程序作为与环境异步运行的协程。每个这样的协程都有一个输入和 输出端口，并由一个循环组成，该循环读取输入端口的感知，并将动作写到输出端口。

协程是一种轻量级的线程，可以在程序中实现并发执行。在这种框架中，每个智能体被看作一个协程，它拥有一个输入端口和一个输出端口。智能体的运行由一个循环组成，该循环不断地读取输入端口的感知信息，并将动作写入输出端口。

这种异步协程的方式可以实现多个智能体并行地与环境进行交互。每个智能体都独立地感知环境的状态，并根据感知信息生成相应的动作。通过异步的方式，智能体可以在不同的时间点独立地与环境交互，而不需要等待其他智能体完成它们的操作。

以下是几个例子，说明这种框架的应用：

多智能体博弈：在一个多智能体博弈环境中，每个智能体可以作为一个协程运行。它们可以独立地感知其他智能体的动作，并根据自己的策略生成相应的动作。

自动驾驶车辆：在一个自动驾驶系统中，每个车辆可以被建模为一个智能体协程。它们可以同时感知周围的交通情况，并根据感知信息做出相应的驾驶决策。

多智能体机器人协作：在一个机器人团队中，每个机器人可以作为一个独立的智能体协程运行。它们可以通过感知信息和交流来协同工作，实现一些复杂的任务，如搬运物品或协作搜索。