

- 学前准备
 - Langchain 开发环境安装
 - 安装 python 3.12
 - 安装 vscode 编程环境
 - 配置 python 解释器
 - 注册搜索引擎账号
 - 安装一款 AI 编码辅助工具（可选，强烈推荐缺少 python 编程基础的学员）
 - 注册一个 AI 聊天账号（可选，强烈推荐缺少 python 编程基础的学员）
 - 检查课程代码是否就绪
 - 检查运行环境文件
 - 利用 VS Code Terminal 和 pip 安装依赖包
 - 运行 Hello World

学前准备

Langchain 开发环境安装

请务必安装或配置如下的工具和环境：

安装 python 3.12

前往[python 3.12 官网](#)，下载相应版本并安装。并在后续安装过程中确保以下选项被选中（缺省选中的也保留）：

- 第一页
 - 选中 Use admin priviledges when installing py.exe
 - 点击 “Customize installation”
- Optional Features
 - 选中 py launcher
 - 选中 for all users
- Advanced Optoions
 - 选中 Install for all users
 - 选中 Add Python to envirionment variables
 - 有时无法生效，仍需要手工设置

****测试安装效果****

1. 在命令行中运行 `python --version`，应出现`python 3.12.7`等字样。
2. 如果失败，一般是没有添加环境变量的问题。请重新安装python，并确保选中 “Add Python to environment variables”
3. 仍然无效请重启电脑

****注意****

1. Langchain 已经停止支持Python3.8之前的版本（2024年11月5日）
2. 另外某些Langchain每次升级后，某些功能在最新的python版本、相关的模块（如pydantic）中有可能不支持。

安装 vscode 编程环境

推荐使用 vscode (<https://code.visualstudio.com/Download>)，以方便与其他语言对接。

vscode中需要安装以下插件（Extension）：

- python （解析和运行python）
 - 打开左侧的Extensions按钮（第5个），搜索python并选择安装。
 - 按下Ctrl + Shift + P，在搜索框中输入Python: Select Interpreter
 - 选择安装的python可执行文件。
 - 在顶端菜单中选择 Terminal / New Terminal，运行 `python --version`，应该出现 python及其版本。
- GitHub Copilot （选装，如果你使用Copilot）
- GitHub Markdown Preview （选装，MD格式是本课程优先使用的文档存储格式）

配置 python 解释器

- 打开 vscode
- 按 Ctrl + Shift + P 打开搜索框
- 输入 Python: Select Interpreter
- 此时可能会看到多个安装的 python 版本，请选择本课程推荐的版本
 - 错误选择可能会导致某些问题，如提示某个包（如 dotenv）未找到。只需要再次选择正确的版本即可。

注册搜索引擎账号

课程中需要用到实时搜索功能，可以注册 Tavily 搜索引擎。方法如下：

- 前往 Tavily 的报价网站(<https://tavily.com/#pricing>)

- 选择**Free**方案，每月可以获得 1000 次免费调用额度，足以支撑课上学习、课下练习所用。
- Tavily 是 Langchain 官网案例中推荐的 AI 搜索引擎，从 Tavily 官网中可见其面向 AI 搜索做了特定优化以保证结果的准确性（因此在使用其他引擎时必须注意广告、排名等导致的问题）。

安装一款 AI 编码辅助工具（可选，强烈推荐缺少 python 编程基础的学员）

推荐 [GitHub Copilot](#)。注意，此软件需要**_科学上网_**方可使用，每年费用 100 美元。

或 [百度搜索平替](#)（如通义灵码等）。

注册一个 AI 聊天账号（可选，强烈推荐缺少 python 编程基础的学员）

首选 [OpenAI 的 ChatGpt](#)。也可以选择[百度的文心一言](#)等替代品。

检查课程代码是否就绪

在拿到课程代码后，请做以下检查活动。

检查运行环境文件

在 src 目录下有一个.env 文件，双击打开后，里边应该至少有如下的内容：

```
OPENAI_API_BASE="https://api.agicto.cn/v1"
OPENAI_API_KEY="sk-一个公开共享但也因此经常欠费甚至失效的密钥（也可以替换成自己的密钥）"
TAVILY_API_KEY="tvly-请输入自己前面注册Tavily获得的KEY"
```

注意：用于教学的公开密钥在教学完成后将在短期内失效。可自行在 agicto.com 注册自己的密钥。过程大致如下：

- 前往一个供应商网站如 <https://agicto.com/>。

- 注册后前往个人中心 <https://agicto.com/space/apikey> 可创建 Key，将 Key 的内容替换到 .env 即可。
- 注册后有时会获赠 10 元免费额度（可供 gpt-4o-mini 大约 1000 万汉字的总吞吐量）。
- 使用本课程的代码，可使用一套代码访问不同的模型，只需要在 `model = ChatOpenAI(model = "gpt-4o-mini")` 切换模型的名称即可。
 - 可用模型名称见：<https://agicto.com/model>

利用 VS Code Terminal 和 pip 安装依赖包

为运行项目或课程中的示例代码，可能需要安装特定的依赖包。可以使用 `pip` 命令安装这些包。在安装依赖包前，请确保已经在 VS Code 的 Terminal 中启动虚拟环境（`venv`），以避免对全局 Python 环境造成影响。

- **启用虚拟环境：** 在 VS Code Terminal 中，确保在当前项目根目录下（一般有一个 README.md）。运行

```
python -m venv venv
.\venv\Scripts\activate
pip install -r requirements.txt
```

此时会发现提示行的最前面有一个蓝色的（`venv`），则表示成功。 **⚠ 注意 ⚠：** 每次启动项目都需要重新运行第二行。

- **安装单个依赖包（选）：** 如果运行项目时提示某个模块找不到（例如 `langchain`），可以单独安装该模块：

```
pip install langchain
```

运行 Hello World

- 双击进入第一课的 `Get_Started/hello_world.py`
- 点击顶端 tab 页右侧的三角符号，运行代码。
- 如果能看到类似 “Hello World!” 的输出，即表明环境正常。

- 如果收到欠费错误 (`402 Payment Required` 或其他类似信息) 无需处理, 课上会提供新的秘钥。
- 如果出现 `ModuleNotFoundError: No module named 'dotenv'` 之类的错误, 可能是:
 - 没有所需依赖包, 请参考 [预装依赖包](#)
 - 配置了错误的 python 解释器, 请参考[配置-python-解释器](#)