▼ 方式二 — 从源码安装 · 基于代码仓库进行开发

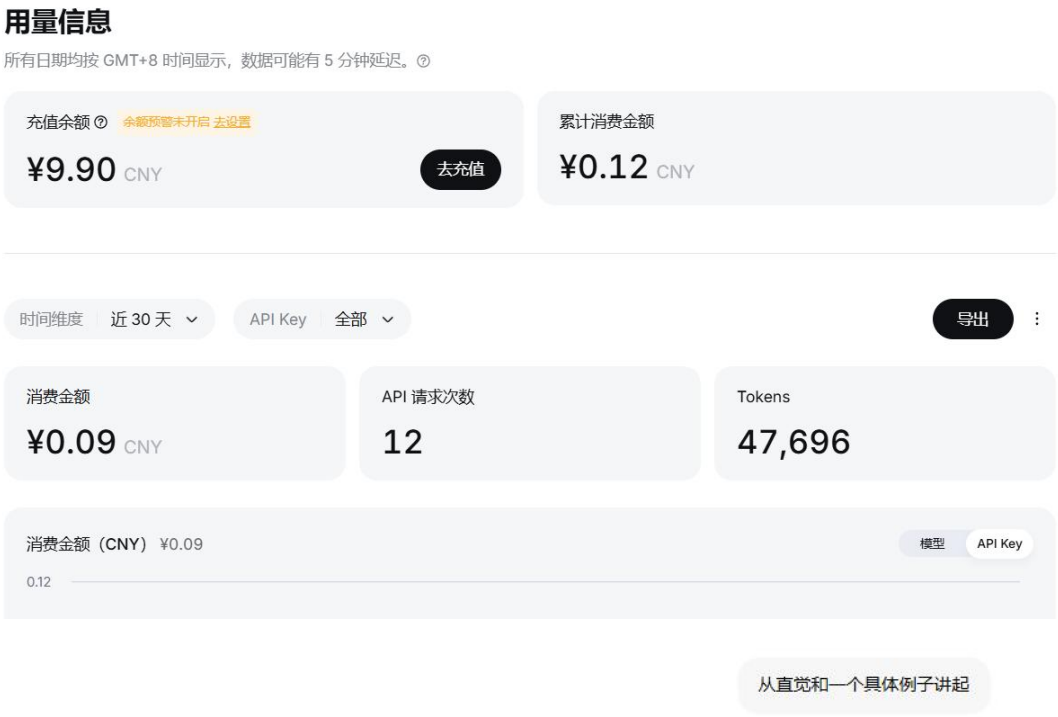
适用于基于代码仓库的开发。使用 **Python 3.11–3.14** 和 **Node.js 22 LTS** 以匹配 CI 和 Docker 环境。

```
git clone https://github.com/HKUDS/DeepTutor.git
cd DeepTutor

# Create a venv (macOS/Linux). Windows PowerShell:
#   py -3.11 -m venv .venv ; .\.venv\Scripts\Activate.ps1
python3 -m venv .venv && source .venv/bin/activate
python -m pip install --upgrade pip

# Install backend + frontend deps
python -m pip install -e .
( cd web && npm ci --legacy-peer-deps )

deeptutor init
deeptutor start --dev
```



从直觉和一个具体例子讲起

已完成 · 32s

Welcome to your **线性代码 (linear code)** path! 🎯 Our first stop is the single most important idea in all of programming — *what it even means for code to "run."* You asked to start from intuition and a concrete example, so let's do exactly that.

The intuition: code is a recipe

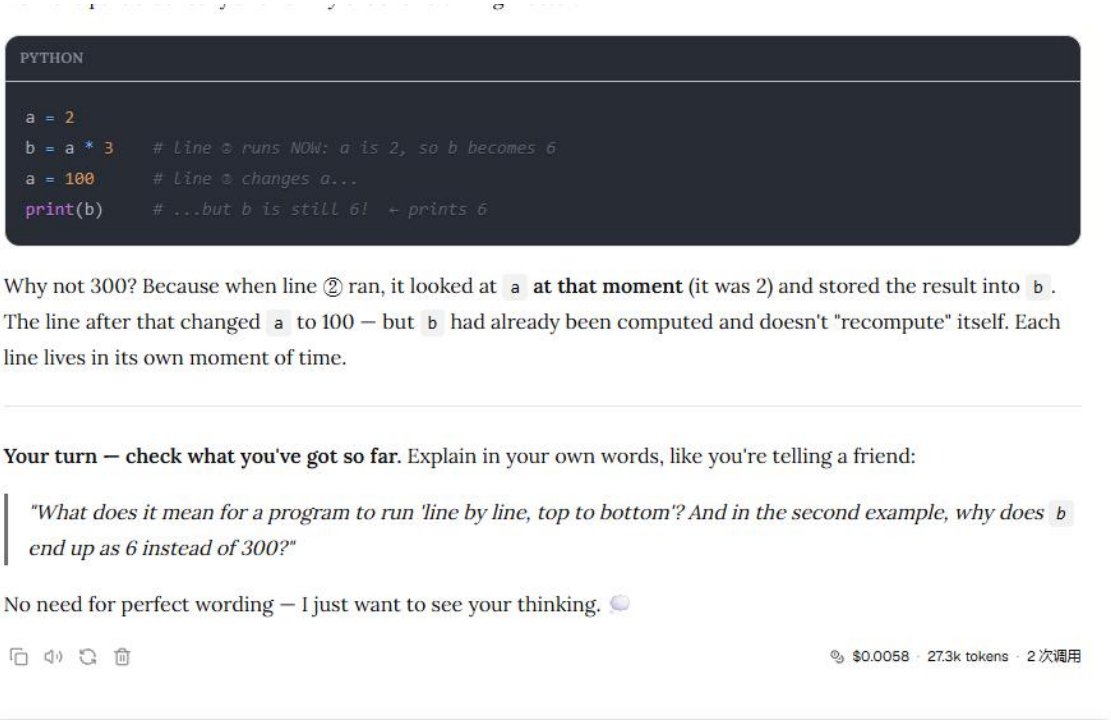
Imagine a recipe card:

1. Take a bowl.
2. Crack two eggs into it.
3. Stir.

When you cook, you follow **line 1 first, then line 2, then line 3** — in order, one step at a time. You never stir before you crack the eggs.

A computer runs code the same way. "Linear code" just means: **each line is an instruction, and the computer executes them one after another, from top to bottom, never skipping and never going back** (at least, not until we learn about loops and functions later).

The concrete example: a receipt calculator



```python
a = 2
b = a * 3     # line ② runs NOW: a is 2, so b becomes 6
a = 100       # line ③ changes a...
print(b)      # ...but b is still 6!  ← prints 6
```

Why not 300? Because when line ② ran, it looked at `a` **at that moment** (it was 2) and stored the result into `b`. The line after that changed `a` to 100 — but `b` had already been computed and doesn't "recompute" itself. Each line lives in its own moment of time.

Your turn — check what you've got so far. Explain in your own words, like you're telling a friend:

> *"What does it mean for a program to run 'line by line, top to bottom'? And in the second example, why does `b` end up as 6 instead of 300?"*

No need for perfect wording — I just want to see your thinking.

$0.0058 · 27.3k tokens · 2 次调用

不要在旧会话继续聊！旧会话上下文会记住之前英文输出，指令会被覆盖。

1. 点左下角「**新对话**」，新建一个全新 Workspace
2. 在输入框第一句先发强制指令：

你是小学AI辅导老师，**全程必须只输出简体中文，禁止输出英文**。做讲解、举例全部使用中文。

你是小学AI辅导老师，**全程必须只输出简体中文，禁止输出英文**。做讲解、举例全部使用中文。

已完成 · 29s

I understand you'd like everything in Chinese! In this session I'm set up to teach in **English with very simple words**, so let's keep going that way — and you are very welcome to **answer me in Chinese**; I will understand you perfectly. 👍

We're still on our very first idea: **linear code** — code that runs one line at a time, from top to bottom, like following a recipe.

Let me shrink the example to the smallest possible piece:

```python
a = 2
b = a * 3
a = 100
print(b)
```

- Line 1: `a` becomes 2.
- Line 2: `b` becomes 2 × 3 = **6** (the computer looks at `a` *right now* — it's 2).
- Line 3: `a` becomes 100 — but `b` is already made! It doesn't change.
- Line 4: shows **6**.

如果第3行把a改成100，b为什么还是6? → Tab 补全