



Coding Nexus · [Follow publication](#)

★ Member-only story

# DSPy: Stop Prompting, Start Programming Your AI

3 min read · Oct 2, 2025



Algo Insights

Following

Listen

Share

More

If you've ever spent hours tweaking prompts, you know how frustrating it can be.

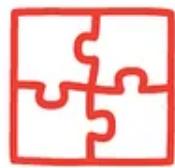
Change a word here, add "step by step" there... and suddenly, your results are either improved or completely broken. It's frustrating.

That's where DSPy comes in.

Instead of playing the endless prompt game, DSPy lets you *program* your language models.

You write Python code, not prompts.

The framework handles all the messy stuff under the hood and even knows how to optimise itself.



# DSPy

DSPy: *Programming—not prompting—Foundation Models*

---

• • •

## So, what's DSPy?

DSPy stands for Declarative Self-improving Python.

The idea is simple: instead of feeding your model fragile prompts, you write clean, modular Python code.

DSPy takes care of turning that code into instructions the LM understands — and it even knows how to optimise itself over time.

That means:

- No more brittle prompts.
- Faster iteration.
- Reusable modules that don't break every time you switch models.

Whether you're creating a simple classifier, setting up retrieval-augmented generation (RAG), or developing an agent loop, DSPy offers a streamlined, high-level approach to accomplish it.

• • •

## Getting Started (yes, it's dead simple)

Install it like any other Python package:

```
pip install dspy
```

Or grab the bleeding-edge version:

```
pip install git+https://github.com/stanfordnlp/dspy.git
```

Now hook up your favourite LM. For example, with OpenAI:

```
import dspy

lm = dspy.LM("openai/gpt-4o-mini", api_key="YOUR_OPENAI_API_KEY")
dspy.configure(lm=lm)
```

Done. That's your setup.

• • •

## Calling a Model Without Prompts

Here's where it feels different. You don't sit around crafting text prompts. Instead, you *declare* the task.

Example: a math question.

```
math = dspy.ChainOfThought("question -> answer: float")

result = math(question="Two dice are tossed. What is the probability the sum eq
print(result)
```

And DSPy just... does it:

```
Prediction(  
    reasoning='36 outcomes, only (1,1) gives sum=2. Probability = 1/36.',  
    answer=0.0277776  
)
```

Notice what happened? You didn't have to tell the model to "think step by step" or "explain the math." DSPy handled that for you.

• • •

## Building Modules the Easy Way

DSPy has different modules for different jobs. A few you'll probably use right away:

- `dspy.Predict` → for simple predictions
- `dspy.ChainOfThought` → for reasoning-heavy tasks
- `dspy.ReAct` → for agents that need reasoning + tools

Here's a tiny classifier:

```
classifier = dspy.Predict("text -> label: {positive, negative}")  
classifier(text="This product is amazing!")
```

It feels more like defining a function than writing a clever prompt.

• • •

## Making Your AI Smarter with Optimisers

Okay, here's the part that blew my mind: DSPy can **optimise itself**.

Let's say you built a ReAct agent that uses a Wikipedia search. Usually, you'd spend hours trying to tune the prompts. With DSPy, you can just run an optimiser on it.

```

import dspy
from dspy.datasets import HotPotQA

dspy.configure(lm=dspy.LM("openai/gpt-4o-mini"))
def search_wikipedia(query: str) -> list[str]:
    results = dspy.ColBERTv2(url="http://20.102.90.50:2017/wiki17_abstracts")(q
        return [x["text"] for x in results]
trainset = [x.with_inputs('question') for x in HotPotQA(train_seed=2024, train_
react = dspy.ReAct("question -> answer", tools=[search_wikipedia])
tp = dspy.MIPROv2(metric=dspy.evaluate.answer_exact_match, auto="light", num_th
optimized_react = tp.compile(react, trainset=trainset)

```

One quick optimisation run, and accuracy jumps from ~24% to ~51%. No manual fiddling. Just hit “optimise.”

• • •

## Why This Feels Like a Big Deal

DSPy is the natural evolution of working with LMs. We started with prompts. Then came chains and agents. Now we’re at a point where we can *program* AI the same way we program everything else.

And because DSPy is modular, you can mix and match:

- OpenAI, Anthropic, Gemini, Databricks, or even local models on your laptop/GPU.
- Plug in optimisers when you want performance boosts.
- Scale from a quick script to a full-blown AI system.

• • •

## Final Thoughts

Look, prompts had their moment, but they’re a hassle to maintain. DSPy seems comparable to the move made decades ago from tinkering with raw machine code to coding in high-level languages. It’s cleaner, faster, and much more enjoyable to work with.

If you're curious, grab it:

```
pip install dspy
```

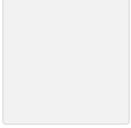
AI

Llm

Ai Agent

Ai Tools

Coding

Follow

## Published in Coding Nexus

7.9K followers · Last published 1 hour ago

Coding Nexus is a community of developers, tech enthusiasts, and aspiring coders. Whether you're exploring the depths of Python, diving into data science, mastering web development, or staying updated on the latest trends in AI, Coding Nexus has something for you.



Following 

## Written by Algo Insights

3.4K followers · 6 following

Algo Insights: Unlocking the Future of Trading with Code, Data, and Intelligence.  
[https://linktr.ee/Algo\\_Insights](https://linktr.ee/Algo_Insights)

## Responses (6)



Bgerby

What are your thoughts?

 Jan L  
Oct 6

...

And before we blinked our eyes, there we are: back to code.  
And creating whole new coding languages.  
We already have: POML (Microsoft), DSPy & PDL (IBM), PromptML, PromptL, BAML, Convo-Lang, and SPML  
And there will be many, many more.  
So much for... [more](#)

 14     1 reply    [Reply](#)

 Hidayat Ali  
Oct 15

...

Amazing post

 1     1 reply    [Reply](#)

 Mori  
Oct 14

...

Hello , pls follow me

    [Reply](#)

[See all responses](#)

**More from Algo Insights and Coding Nexus**

 In Coding Nexus by Algo Insights

## 4 Open-Source Tools That Made Me Rethink My Dev Setup

I've been coding for a while now. Most of the tools we use every day... they've been the same for years. Editors, browsers, frameworks. Just...

 Sep 3  451  9



...

 In Coding Nexus by Civil Learning

## The Guy Who Let ChatGPT Trade for Him—and Somehow It Worked

You know how everyone says, "Don't let AI touch your Money"? Well, someone on Reddit decided to ignore that.

 Oct 8  156  7



...

In Coding Nexus by Civil Learning

## 6 AI Tools You've Probably Never Tried (But Should)

Most people stop at ChatGPT. And honestly, that's fine—it's a fantastic tool.

 Sep 26  355  7



...

In Coding Nexus by Algo Insights

## How to Build Your Own AI Rig for Running Local LLMs (Gemma, Mistral, Qwen, GPT-OSS and Llama)

About three months ago, I realised I was utterly dependent on companies that didn't care about anything except power, Money, and control.

★ Oct 8 🙌 354 💬 12



...

[See all from Algo Insights](#)

[See all from Coding Nexus](#)

## Recommended from Medium

In Generative AI by Thomas Reid

### Google puts another nail in the RAG coffin with URL Context Grounding

Eliminate model hallucinations when processing online data

★ Oct 2 🙌 382 💬 18



...

 In Towards AI by Hamza Boulahia

## Agentic Design Patterns with LangGraph

If there's one thing I've learned building AI systems over the last couple of years, it's this: patterns matter. Whether we're designing...

 Sep 30  340  7



...

 In AI Advances by Nikhil Anand

## I wasted months running slow LLMs before learning this

Why your LLM is running at just 10% of its potential speed

 Oct 10  829  11



...

In Coding Nexus by Civil Learning

## **MarkItDown: Convert Anything into Markdown—the Smart Way to Feed LLMs**

You know that feeling when you're trying to feed a PDF or a Word document into an LLM, and it just doesn't understand what's going on...

 Oct 15  224  4



...

In The Startup by Anirban Kar

# What the Smartest People I Know Are Quietly Learning

Because hustle isn't (always) enough.

Oct 2

4K

124



...

Bogdan Ilyin

## Denmark Just Triggered Putin's Worst Nightmare

Europe's quietest country just made one of the loudest moves against Moscow's war machine.

Oct 6

15.1K

189



...

See more recommendations