SELF-PACED FIELD GUIDE

automation/diary

LEARN AI · FROM GPTS TO EVERYTHING

The AI Field Guide.

Understand how GPTs really work — and build with them.

A lean, sequenced path through the best free courses on the internet — built for a busy professional with two hours a day. No fluff, no tutorial hell, no $2,000 bootcamp. The right things, in the right order, with the tips that save you months.

Pace 2 hrs / day **Length** 6 months · 24 weeks **Cost** Free resources only

Level Beginner → builder

BEFORE YOU START

How to use this guide.

This isn't a reading list — it's a *route*. Each chapter is a phase; do them in order. Each stands on its own, so you gain real ability even if you stop early. Tap any course title to open it. The single rule that decides whether this works: *you must write code, not just watch it.*

THE ONE TRAP THAT KILLS MOST LEARNERS

Tutorial hell — watching video after video, feeling productive while building nothing. Watching ≠ knowing. If a week passes with no code *you* wrote, you're consuming, not learning. Every phase here ends in something you build.

The daily two-hour loop

The same 120 minutes, every day — the real accelerant. It roughly triples retention versus re-watching.

~80 MIN · LEARN

Watch & build

Follow the lesson with the editor open. Pause, type it yourself, break it, fix it.

~25 MIN · ENCODE

Teach it back

Drop notes into NotebookLM for a cited recap + audio for walks. Then explain it to an AI like it's a 12-year-old — the gaps appear.

~15 MIN · RECALL

Anki

Make 5–10 cards from today; clear the due queue. Card only what you already understand.

WHY THIS WORKS — THE SCIENCE IN ONE LINE

Active recall + spaced repetition beat re-reading for long-term memory; teaching-to-learn forces the gaps open; source-grounded AI recaps cut the time to "I get it." You're not studying harder — you're studying in the order your brain keeps.

DAY ZERO · 20 MINUTES

Set up your workspace.

A little setup now saves hours later. Get these in place before Week 1 — every one is free.

Code — Python & an editor

Install Python 3 and a free editor like VS Code. Prefer zero install? Run everything in the browser with Google Colab notebooks.

Free GPUs — no hardware needed

You don't need an expensive machine. Kaggle ↗ and Google Colab give you GPUs for free — plenty for this entire guide.

Accounts (all free)

Create a GitHub account to host your work, a Hugging Face account for models and datasets, and — by Phase 3 — one LLM API key.

Your two learning tools

Install Anki ↗ for spaced-repetition flashcards, and open a NotebookLM ↗ notebook for cited recaps and walk-friendly audio.

> THE ONE RULE THAT MAKES IT WORK
>
> Two hours, daily, with the editor open. Protect the slot like a meeting with someone important — because it is.

THE PATH · FIVE PHASES

The route.

Foundations to frontier. Each phase stands alone; the *spine* (Phase 1) is the one to never skip.

0 Intuition & warmup

Week 1 · ~12h

Before mechanics, build a *picture*. You'll see what a neural network and a GPT actually are — and knock the rust off Python so the real work flows.

3Blue1Brown — Neural Networks ↗ · Kaggle Learn — Python ↗

1 Build a GPT SPINE

Weeks 2–6 · ~50h

The chapter that separates people who *use* AI from people who *understand* it. You build a working GPT from scratch, in Python, guided by the person who helped build them for real.

Karpathy — Deep Dive into LLMs ↗ · Karpathy — Zero to Hero ↗

2 Foundations backfill

Weeks 7–11 · ~40h

Now that you can build, fill the theory you actually hit — not a 90-hour survey "just in case." *Pick one* track below. Doing both is the classic time-sink.

fast.ai — Practical Deep Learning ↗ · Andrew Ng — ML Specialization ↗

3 Build real things

Weeks 12–20 · ~70h

The chapter that *pays you back*. APIs, retrieval (RAG), and agents — the skills behind almost every AI product shipping today. Learn each by building, not bookmarking.

DeepLearning.AI — Short Courses ↗ · Hugging Face — Agents Course ↗

4 Depth & frontier

Weeks 21–24 · ~35h

The final mile: make a model *yours*. Fine-tuning and post-training turn a general model into a specialist — and take you to the edge of the field.

DeepLearning.AI — Finetuning ↗ · UC Berkeley — LLM Agents MOOC ↗

THE PLAN · 24 WEEKS · ~14H EACH

The 6-month map.

Every week, what to do and what you'll have to show for it. The Full edition expands each into resources, milestones, and tips.

MONTH 1 Foundations & first build

WK 01	Intuition + setup
WK 02	Deep Dive map + micrograd
WK 03	micrograd
WK 04	makemore

MONTH 2 Build a GPT, then ground it

WK 05	Let's build GPT
WK 06	Tokenizer + your own GPT · gate
WK 07	fast.ai 1–2
WK 08	fast.ai 3–4

MONTH 3 Foundations → first LLM app

WK 09	fast.ai 5–6
WK 10	fast.ai 7–8
WK 11	consolidate + mini-project
WK 12	APIs: prompt eng + systems

MONTH 4 RAG & your first product

WK 13	Hugging Face LLM course
WK 14	RAG fundamentals
WK 15	RAG over your data (v1)
WK 16	Agents I

MONTH 5 Agents & ship it

WK 17	Agents II + LangGraph
WK 18	eval + UI
WK 19	production (optional)
WK 20	ship v1 + feedback · gate

MONTH 6 Depth, frontier & next

WK 21	fine-tuning
WK 22	post-training
WK 23	apply tuning to project
WK 24	frontier + plan next

FORCE MULTIPLIERS

The learn-faster toolkit.

Three tools that turn the daily loop into a system — each free or near-free, each

backed by real learning science.

NotebookLM ↗

Google · free

Drop transcripts, notes and papers in; it answers *grounded only in your sources*, with citations — so it doesn't hallucinate. Auto-generates study guides and an Audio Overview for walks.

TIP One notebook per phase. Turn each day's lesson into an audio recap and listen the next morning.

Anki ↗

Open-source · free

Spaced-repetition flashcards. The algorithm shows a card right as you're about to forget it — the most evidence-backed way to make knowledge permanent.

TIP Have an AI draft cards from your notes, then edit. Card a concept only AFTER you understand it.

Feynman-with-an-AI

Any chat LLM

After each lesson, explain the concept to an AI as if teaching a beginner; tell it to play a skeptical student. The instant you can't explain it cleanly, you've found your gap.

TIP Prompt: "You're a curious 12-year-old. I'll explain X — interrupt with the obvious questions and tell me where I hand-waved."

HONEST SUBTRACTIONS

What to skip.

Knowing what *not* to do is half the speed. These are the time-sinks that look like progress.

- **Don't** grind the full 90-hour Ng survey *and* fast.ai. Pick one — they overlap.
- **Don't** start with heavy math courses. Visual intuition + just-in-time math is enough until proven otherwise.
- **Don't** framework-hop (LangChain vs LlamaIndex vs CrewAI). Learn one deeply — the ideas transfer.
- **Don't** chase paid certificates. Your portfolio of built things is the credential that gets noticed.
- **Don't** wait until you "feel ready" to build. You build TO get ready. Ship small, ship early.
- **The big one — don't live in tutorial hell.** A week with no code you wrote yourself means you're collecting lessons, not learning.

THE ONE SENTENCE TO REMEMBER

Right things, right order, built with your own hands — that beats every shortcut and every $2,000 course. You have the route. Now walk it, two hours at a time.

AUTOMATION/DIARY FIELD GUIDES

“Right things, right order, built with your own hands — that beats every shortcut.”

automation/diary

THE AI FIELD GUIDE · v1 · 2026 · MADE WITH THE SLOW INK SYSTEM