



NEXILY AI ACADEMY · LEVEL 5

Build *systems*

The big leap: going from “I use AI” to “I build systems that use AI”.
Workflows, automations and specialised assistants — without
being a developer.

The path

Six modules to change your stance. Until now, you asked AI to do a task. Now, you design systems that chain tasks and put AI to work for you, again and again. Universal concepts, valid whatever the tools.

00	How to follow this course	4
01	Understanding workflows	5
02	Automating intelligently	8
03	Adding AI to a workflow	11
04	Creating specialised assistants	14
05	Building a library of systems	17
06	Workshop: your first workflow, click by click	20
★	Final project: a complete workflow that solves a real problem	25
	The end of level 5	27
	What's next?	28
	About Nexily	29

PUT IT INTO PRACTICE

Final project: a complete workflow that solves a real problem 27

NOTE

This level teaches principles, not specific tools — they change too fast. A badly designed system makes mistakes at scale: always test on a small volume, and keep a human in the loop for decisions and sends that matter. This level teaches principles that work with any tool; the module 6 workshop applies them in Zapier. A badly designed system makes mistakes at scale: always test on a small volume, and keep a human in the loop for decisions and sends that matter.

How to follow this course

This level is the most strategic. Here we think in “systems” rather than “tasks”. Good news: it takes common sense, not code.

Think in systems, not tasks

Using AI means solving a problem once. Building a system means making the problem solve itself, every time it comes up. This shift in stance is the heart of level 5.

You don’t need to be a developer. Here, we learn to **think** systems: which steps, in what order, where AI comes in. The tools that run all this change constantly; the principles don’t.

KEY POINT

A good system turns a one-off effort into a recurring result. You invest once to design it, then it works for you — the highest level of leverage AI allows.

How to read the boxes

BOX

WHAT IT HOLDS

Key point	The principle to keep, even if you forget the rest.
Copy this	A template ready to copy and adapt.
Pitfall	The common design mistake at this step.
Put it into action	What you build before continuing.

- 1 Pick a recurring problem, personal or professional, to be your running example.
- 2 Open a document “My systems” where you’ll document your workflows as the modules go.
- 3 Accept starting simple: one first system, however modest, beats a grand plan never built.

01

THE SYSTEMS PATH

Understanding workflows

The concept that structures the whole level. A workflow is a sequence of steps that turns an input into a result — learn to see them everywhere.

BY THE END OF THIS MODULE, YOU CAN

- Define simply what a workflow is
- Break a task down into input, processing and output
- Spot repetitive tasks and map a process

What is a workflow?

A workflow is simply an ordered sequence of steps leading from a starting point to a result. You run dozens of them every day without naming them: handling a letter, preparing a meal, replying to a client.

Naming it and writing it down is what lets you improve it, delegate it — to yourself systematically, or to AI. A process you've never laid out stays fuzzy and impossible to automate.

Input → processing → output

Every workflow, however complex, comes down to three stages. It's the reading grid to keep in mind to analyse any task.

— **The input** What triggers and feeds the workflow: an email received, a document, a request, a date.

⚙️ **The processing** What you do with that input: read, summarise, sort, decide, write. This is where AI comes in most often.

← **The output** The result produced: a reply sent, a file filed, a notification, a row added to a table.

EXAMPLE

Handling a received invoice. **Input:** the invoice arrives by email. **Processing:** you read the amount and due date, check, record. **Output:** the invoice is filed and a payment note is scheduled. Three stages, one workflow — which we'll soon be able to tool up.

Spotting repetitive tasks

The workflows worth building are the ones that recur. Over a week, spot what you redo identically: those are your best targets, because the design effort will be amortised dozens of times.

COPY THIS

"Here are my recurring tasks: [list]. For each, describe it as input → processing → output, and say whether it's suited to a workflow."