

# Lecture III — First Steps with an Agent

## AI-Assisted Programming for PhD Researchers

Dr. Tobias Vlček  
Helmut Schmidt University — September 2026

### Meet Your Agent

#### What Is OpenCode?

- An open-source coding agent that lives in your **terminal**
- Works with any model provider — no lock-in
- Switch models or editors later, keep the same workflow
- Today it is your hands-on tool for the rest of the course

#### Install It (Together, Now)

Pick the line for your system:

```
# macOS / Linux
curl -fsSL https://opencode.ai/install | bash
# macOS via Homebrew - alternative
brew install anomalyco/tap/opencode
```

Then confirm it installed:

```
opencode --version          # expect X.Y.Z
```

#### Connect Your Model

```
opencode auth login
```

- Choose **Mistral**, paste the API key from setup
- Then select the model: **Mistral Medium 3.5** for our labs

Key stored locally in OpenCode's config — never committed to git.

#### The Same Agent in Zed

- Zed's agent panel speaks the same protocol (ACP)
- Open the panel, pick **OpenCode** — same session, same power
- Bonus: review each edit as an inline diff in the editor
- Use whichever surface you prefer today — terminal or Zed

## Safety Rails

### What May It Do?

- OpenCode asks permission per action class — editing files, running commands
- Start restrictive; loosen consciously as trust grows
- Every permission is configurable in `opencode.json`

...

- The default stance: it proposes, you approve

### Your Undo Buttons

- `/undo` reverts OpenCode's last change; `/redo` puts it back
- Beneath that: `git diff` to see it, `git restore` to revert it
- Nothing is scary while your working tree is clean
- **Commit before you let an agent loose** — a clean tree is your rescue point

## Giving Context

### The Agent Reads, It Does Not Know

- Recall the **context window**: the agent only sees what is in it
- It reads what you show it — or what it opens itself
- It does **not** see your screen, your intent, or yesterday's session
- No memory between sessions unless you write it down

### Pointing at Things

- `@path/to/file` pulls a file into the conversation
- Paste error messages **verbatim** — the full traceback
- Name constraints explicitly: "Python 3.12, pandas only"

```
@legacy_analysis.py why does this crash on empty input?
```

### AGENTS.md — Standing Instructions

- A file the agent reads **every** session, automatically
- Holds your project's purpose, commands, and rules
- Keep it short and current
- A stale AGENTS.md misleads more than no file at all

### What Belongs In It

Keep it to what the agent needs to act:

- **Yes:** build/test commands, structure overview, hard constraints
- **Yes:** style choices — "snake\_case, type hints on public functions"
- **Constraint example:** "`data/raw` is read-only"
- **No:** essays, wish lists, secrets or API keys

## /init Writes a Draft

- `/init` scans your repo and generates a first AGENTS.md
- Treat it as a draft — read it, cut it down, correct it
- The agent's guess about your project is a start, not truth
- You will do exactly this in Lab 1

## Tokens, Cost, and Models

### Tokens Are the Meter

- Everything in and out is counted in **tokens** (~4 characters each)
- The whole conversation is resent to the model each turn
- Long sessions cost more, and the agent degrades as they grow

...

- Start a fresh session per task — cheaper and sharper

### Context Windows

Each model has a hard limit on tokens it can hold at once:

| Text                    | Rough tokens |
|-------------------------|--------------|
| One page of prose       | ~500         |
| A 500-line source file  | ~6,000       |
| Our models' full window | 128k–262k    |

Plan what you load — do not pour the whole project in at once.

### Picking a Model

- **Mistral Small**: fast and cheap, for routine edits and boilerplate
- **Mistral Medium 3.5** (our lab default): harder reasoning, for design and tricky bugs
- Match the model to the task; skip reasoning you do not need
- Switching is one command — no restart

### When the Free Tier Throttles

- Free tiers have **rate limits** — you will hit them eventually
- Symptoms: 429 errors, stalled or truncated responses
- Do: wait a moment, shrink your context, or drop to a smaller model
- Still stuck? Raise your hand — I have backup keys

## The Core Loop

### Explore → Plan → Implement → Verify

THE workflow of this course, four phases, each with a job:

- **Explore**: load the right context; understand before touching
- **Plan**: agree on the approach before any edit

- **Implement:** make the change, one step at a time
- **Verify:** prove it works; skipping explore means the agent guesses

## Plan Mode vs Build Mode

- **Tab** toggles between the two modes
- **Plan mode** reads and suggests but never edits — use it to explore
- **Build mode** acts — it edits files and runs commands
- Stay in Plan until the plan is right, then switch to Build

## Explore First

Good opening prompts cost little and load the right context:

```
Explain this repository – structure and entry points
How does the humidity parsing work?
What would break if I changed the date format?
```

Exploring is cheap and fast — and it sets up everything after.

## Verify Last, Every Time

- Run the code — does it actually execute?
- Read the diff — is that really what changed?
- Check the claim against the output, not the agent’s summary

## Still the Scientist of Record

⚠ Warning

“The agent said so” is not verification. You are still the scientist of record.

## Lab 1

### Your Mission

In Lab 1 you meet the inherited script and put the loop to work:

- Make the agent **explain** the repository you cloned
- Write a short **AGENTS.md** with `/init`, then trim it
- Fix the first obvious problem — and **verify** the fix
- The **Explain-it boxes** in the lab are mandatory — answer them before moving on

## Continue Your Journey

### Next Up

- **Lab 1 — Understanding Inherited Code** starts now
- [Lab 1 — Explore the inherited code](#)
- [Lecture IV — Planning with AI](#) — 15:00
- [Course literature and references](#)

## **Bibliography**