CENTER FOR SCALABLE DATA ANALYTICS AND ARTIFICIAL INTELLIGENCE

Lea Kabjesz, Lea Gihlein, Mara Lampert, Luisa Götze

# Coding effectivly with AI

## Case Study: Building a Snake Game

# Quick Poll

1. Who has used Cursor before?

# Quick Poll

2. Who uses LLMs (e.g. ChatGPT) regularly for work?

# Quick Poll

3. Who has experience with Prompt Engineering?

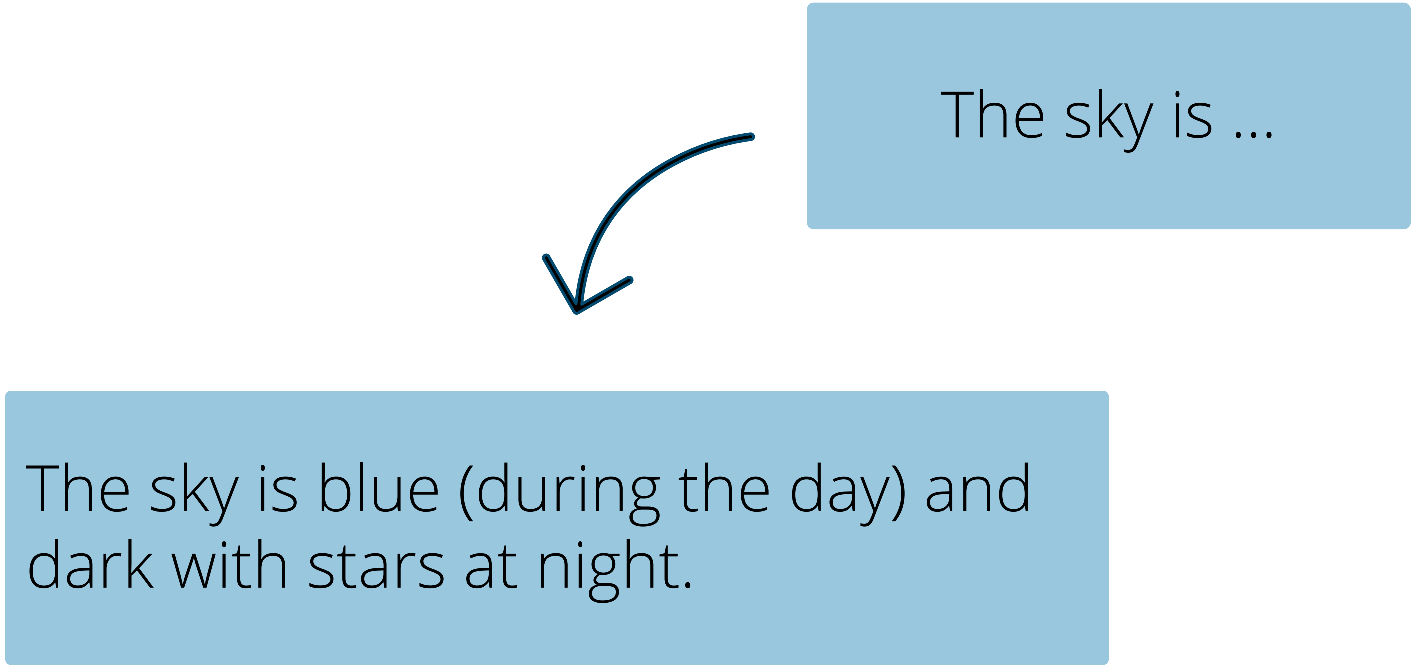
# Agenda

1. Introduction to AI Code Generation
2. Prompting Techniques – Theory
3. Practical: Snake Game Implementation
4. Lessons Learned & Pitfalls
5. Q&A

# What is a Prompt?

The sky is ...

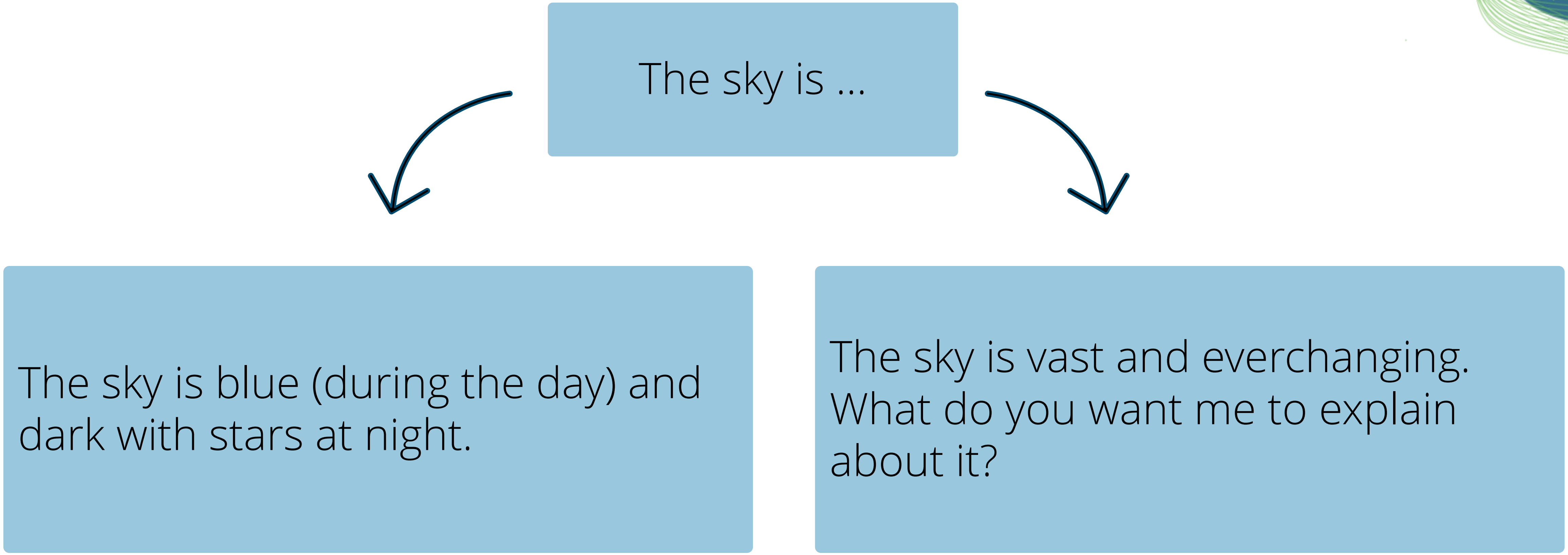
# What is a Prompt?



The sky is …

The sky is blue (during the day) and dark with stars at night.

# What is a Prompt?



The sky is ...

The sky is blue (during the day) and dark with stars at night.

The sky is vast and everchanging. What do you want me to explain about it?

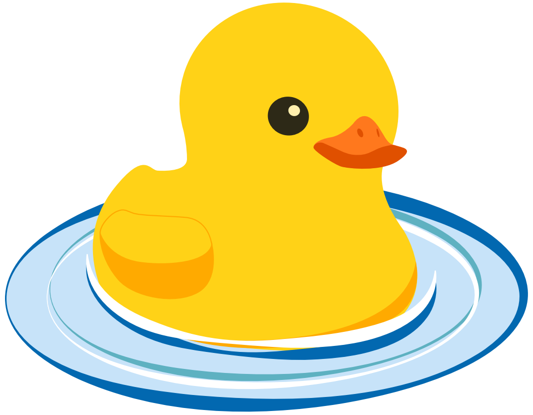
# What is Prompt Engineering?

**“Designing inputs for better outputs”**

- Quote by ChatGPT-5

Logo: © OpenAI. Quelle: https://openai.com/brand
Verwendet gemäß den Markenrichtlinien von OpenAI

# Good Practices



- Be specific!
- Talk to AI like to a “Rubber Duck”

## If you get stuck

- ask AI for a “radically different approach”
- Provide screenshots
- Ask AI to generate prompts („help me write better questions“)

# Prompting Frameworks

„CLEAR“

**C**ontext **L**anguage **E**xample **A**udience **R**equest

[CLEAR Framework: Conversational Language with Emphasis on Adaptable Results: Complete Guide with Examples | AiPromptsX]

# Prompting Frameworks

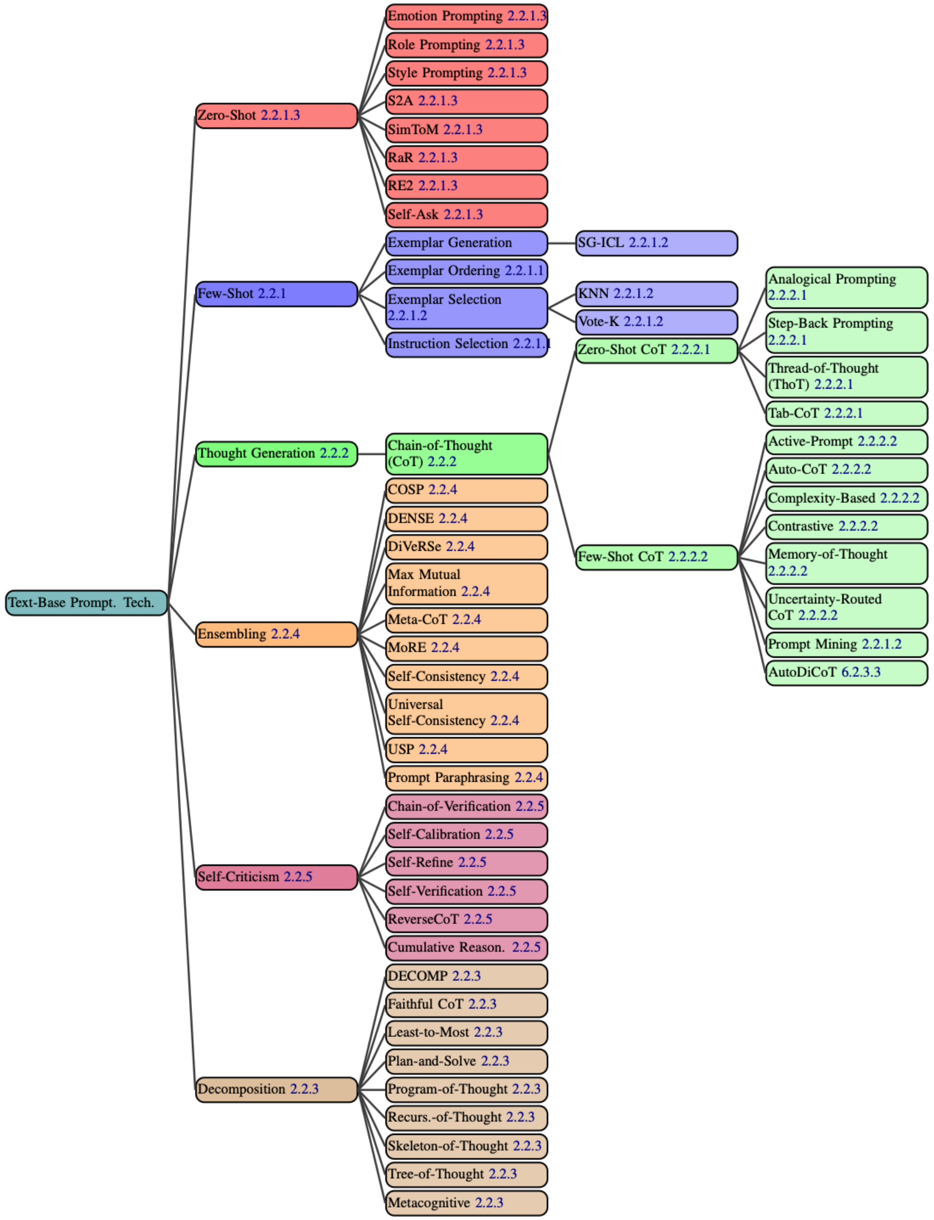
## ContextLanguageExampleAudienceRequest

- "You work at a mid-sized AI company, as a team leader focusing on the rain-forest. (Context)
- Use clear, professional language. (Language)
- Include a phrase like 'on track to meet the deadline'. (Example)
- The audience is senior leadership. (Audience)
- Summarize the project progress for a report. (Request) "

CLEAR Framework: Conversational Language with Emphasis on Adaptable Results: Complete Guide with Examples | AiPromptsX

# Prompt Engineering Techniques

- Chain-of-Thought
- Chain-of-Knowledge
- Rephrase & Respond
- Reflection Prompting
- Self-Consistency



- Text-Base Prompt. Tech.
  - Zero-Shot 2.2.1.3
    - Emotion Prompting 2.2.1.3
    - Role Prompting 2.2.1.3
    - Style Prompting 2.2.1.3
    - S2A 2.2.1.3
    - SimToM 2.2.1.3
    - RaR 2.2.1.3
    - RE2 2.2.1.3
    - Self-Ask 2.2.1.3
  - Few-Shot 2.2.1
    - Exemplar Generation
      - SG-ICL 2.2.1.2
    - Exemplar Ordering 2.2.1.1
    - Exemplar Selection 2.2.1.2
      - KNN 2.2.1.2
      - Vote-K 2.2.1.2
    - Instruction Selection 2.2.1.1
  - Thought Generation 2.2.2
    - Chain-of-Thought (CoT) 2.2.2
      - Zero-Shot CoT 2.2.2.1
        - Analogical Prompting 2.2.2.1
        - Step-Back Prompting 2.2.2.1
        - Thread-of-Thought (ThoT) 2.2.2.1
        - Tab-CoT 2.2.2.1
      - Few-Shot CoT 2.2.2.2
        - Active-Prompt 2.2.2.2
        - Auto-CoT 2.2.2.2
        - Complexity-Based 2.2.2.2
        - Contrastive 2.2.2.2
        - Memory-of-Thought 2.2.2.2
        - Uncertainty-Routed CoT 2.2.2.2
        - Prompt Mining 2.2.1.2
        - AutoDiCoT 6.2.3.3
  - Ensembling 2.2.4
    - COSP 2.2.4
    - DENSE 2.2.4
    - DiVeRSe 2.2.4
    - Max Mutual Information 2.2.4
    - Meta-CoT 2.2.4
    - MoRE 2.2.4
    - Self-Consistency 2.2.4
    - Universal Self-Consistency 2.2.4
    - USP 2.2.4
    - Prompt Paraphrasing 2.2.4
  - Self-Criticism 2.2.5
    - Chain-of-Verification 2.2.5
    - Self-Calibration 2.2.5
    - Self-Refine 2.2.5
    - Self-Verification 2.2.5
    - ReverseCoT 2.2.5
    - Cumulative Reason. 2.2.5
  - Decomposition 2.2.3
    - DECOMP 2.2.3
    - Faithful CoT 2.2.3
    - Least-to-Most 2.2.3
    - Plan-and-Solve 2.2.3
    - Program-of-Thought 2.2.3
    - Recurs.-of-Thought 2.2.3
    - Skeleton-of-Thought 2.2.3
    - Tree-of-Thought 2.2.3
    - Metacognitive 2.2.3

Cropped from Schulhoff et. al (2024), licensed CC-BY 4.0 https://arxiv.org/abs/2406.06608

# Chain-of-Thought

Idea: Ask the model to show reasoning steps

Why: Improves performance on complex, multi-step tasks. (--> Math!).

Paper: Wei et al. (2022) https://arxiv.org/abs/2201.11903

# Chain-of-Thought

## Standard Prompting

Model Input

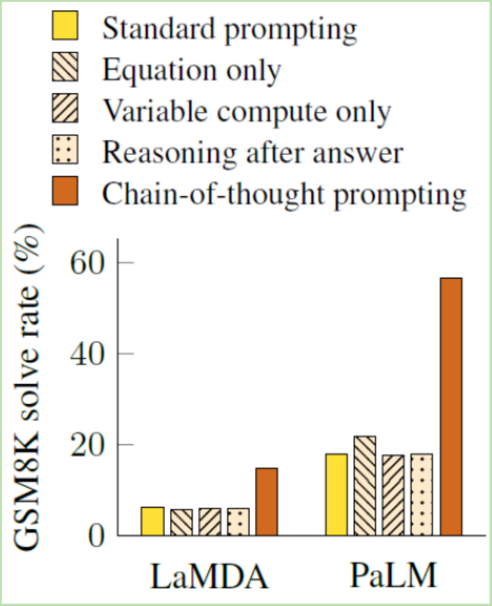
Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls . Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27



Wei et al 2022, licensed CC-BY 4.0
https://arxiv.org/abs/2201.11903

# Chain-of-Thought

## Chain of Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls . Each can has 3 tennis balls. How many tennis balls does he have now?

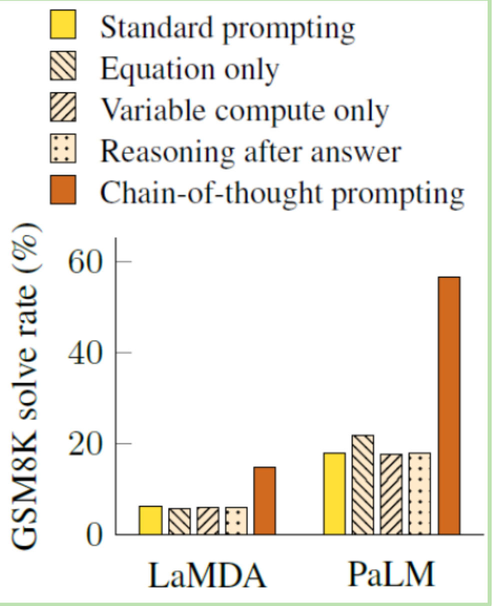
A: Roger started with 5 balls. 2 cans of tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 – 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9.

**„Lets think this step by step“**



Wei et al 2022, licensed CC-BY 4.0
https://arxiv.org/abs/2201.11903

# Chain-of-Knowledge

**Idea**: Introduce relevant facts into the reasoning chain.

**Why**: Reduces hallucinations. Keeps reasoning grounded in facts.

**Origin:** Wang et al. (2023) _arxiv.org/abs/2306.06427

# Chain-of-Knowledge

## a) Standard ICL Prompting

### Input

Q: Is the following sentence plausible?
'Joao Moutinho was out at third.‘

A: No.

Q: Is the following sentence plausible?
'Derrik White backhanded a shot.'

### Output 👎

A: Yes.

## b) Chain-of-Thought Prompting

### Input

Q: Is the following sentence plausible?
'Joao Moutinho was out at third.‘

Joao Moutinho is a soccer player.
Being out at third is part od baseball, not soccer.

A: No.

Q: Is the following sentence plausible?
'Derrik White backhanded a shot.'

### Output 👎

A: Yes, it is plausible. This sentence suggests that Derrik White, who is most likely a basketball or hockey player, made a backhanded shot attempt.

## c) Ours: Chain-of-Knowledge Prompting

### Input

Q: Is the following sentence plausible? 'Joao Moutinho was out at third.‘

Evidence triples:
1. (Joao, isA, soccer player)
2. (being out at third, is part of, baseball)

Explanation hints: Being out at third is part of baseball, yet, Joao Moutinho is a soccer player.

A: No.

Q: Is the following sentence plausible? 'Derrik White backhanded a shot.'

### Output ✓

Evidence triples:
1. (Derrick White, isA, basketball player)
2. (backhanded shot, is commonly used in, hockey or tennis)

Explanation hints: Backhanded shot is commonly used in hockey or tennis, but not in basketball.

A: No.

**„Use Chain-of-Knowledge: evidence triplets (subject, relation, object)“**

Inspired by: Wang et al. 2024, licensed CC-BY 4.0
https://aclanthology.org/2024.acl-long.271/

# Rephrase & Respond

**Idea:** Ask model to restate your question in different words.

**Why:** Forces the model to clarify intent before generating content.

**Paper:** Deng et al. (2023) [arxiv.org/2311.04205](https://arxiv.org/2311.04205)

# Rephrase & Respond

„Rephrase and expand the question and answer“

Original Question

Was {person} born in an even day?

Was {person} born in an even month?

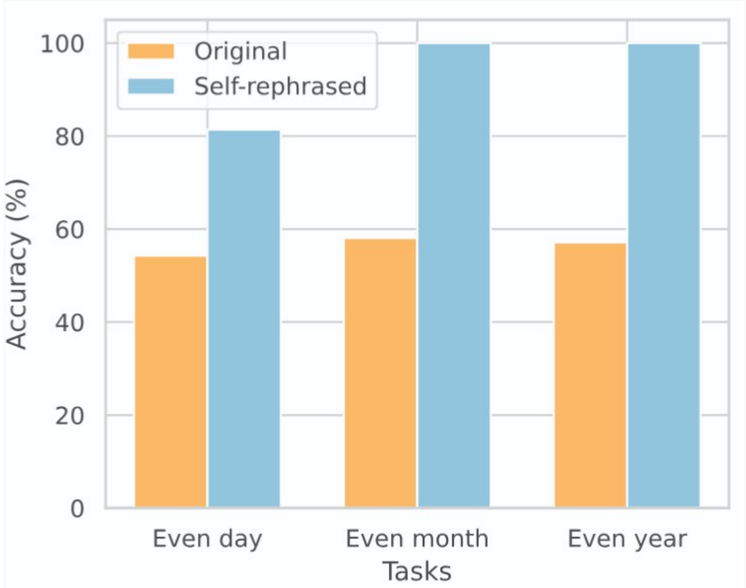
Was {person} born in an even year?

Self-rephrased question

Could you provide more information on wether the individual named {person} was born on a day that is an even number? This refers to dates such as the 2nd, 4th, 6th, 8th, and so on within a given month.

Can you provide the specific month of the year in wich {person} was born to determine if it falls into an even-numbered month such as February, April, June, August, October, or December?

What is the birth year of {person} and is it an even number?



Inspired by: Deng et al. (2023) licensed CC-BY 4.0 arxiv.org/2311.04205

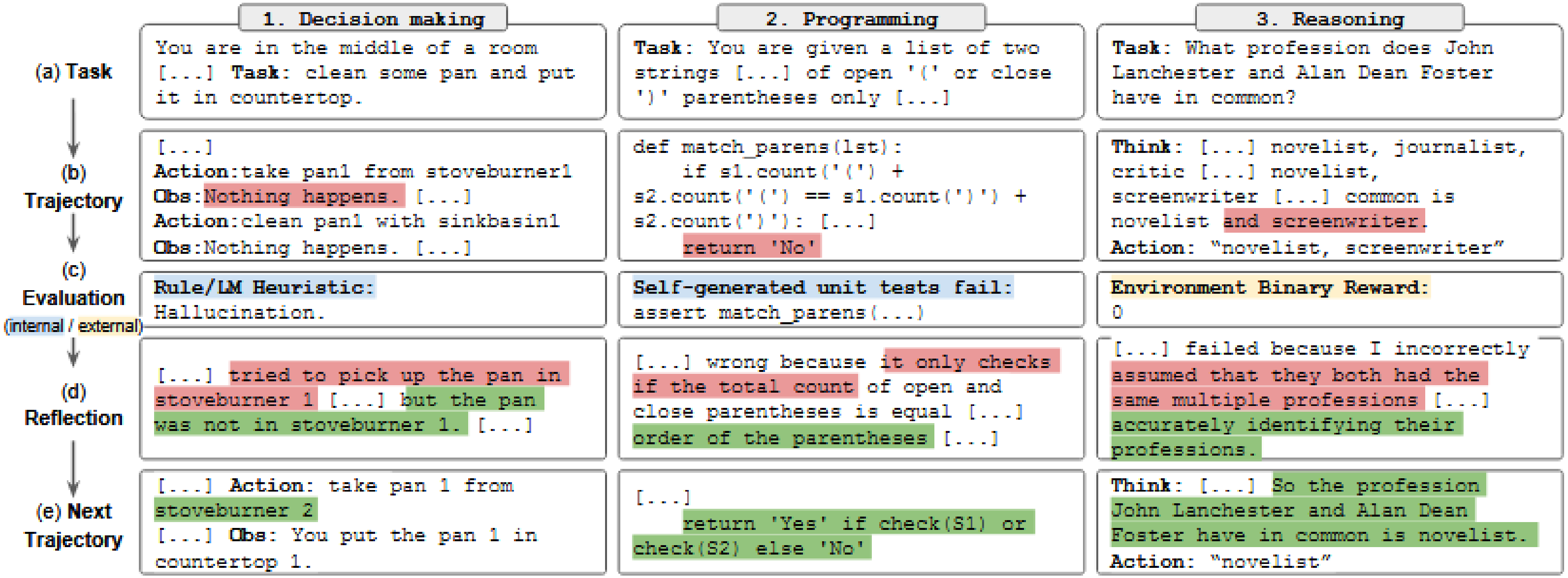
# Reflection Prompting

**Idea:** Ask the AI to review & improve its own output.

**Why:** Catches mistakes or gaps and improves quality by iterating over tasks/solutions

**Paper:** Shinn et al. (2023) https://arxiv.org/abs/2303.11366

# Reflection Prompting



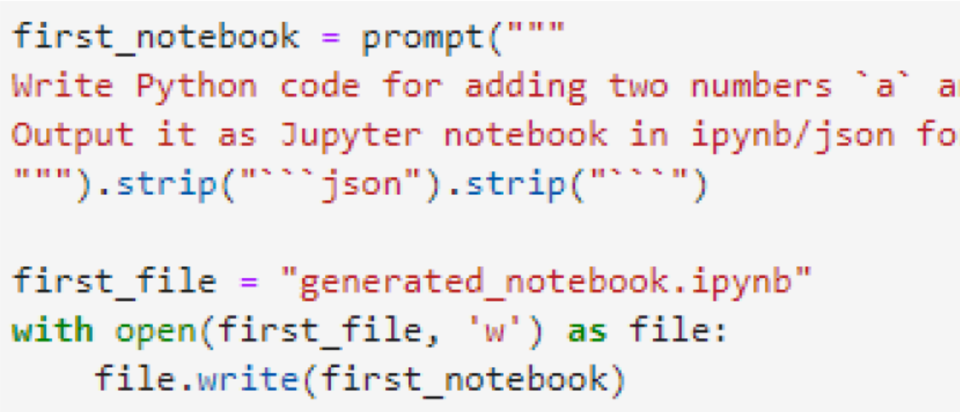
| | 1. Decision making | 2. Programming | 3. Reasoning |
|---|---|---|---|
| (a) Task | You are in the middle of a room [...] **Task**: clean some pan and put it in countertop. | **Task**: You are given a list of two strings [...] of open '(' or close ')' parentheses only [...] | **Task**: What profession does John Lanchester and Alan Dean Foster have in common? |
| (b) Trajectory | [...]<br>**Action**:take pan1 from stoveburner1<br>**Obs**:Nothing happens. [...]<br>**Action**:clean pan1 with sinkbasin1<br>**Obs**:Nothing happens. [...] | `def match_parens(lst):`<br>`    if s1.count('(') +`<br>`s2.count('(') == s1.count(')') +`<br>`s2.count(')'): [...]`<br>`    return 'No'` | **Think**: [...] novelist, journalist, critic [...] novelist, screenwriter [...] common is novelist and screenwriter.<br>**Action**: “novelist, screenwriter” |
| (c) Evaluation (internal / external) | **Rule/LM Heuristic:**<br>Hallucination. | **Self-generated unit tests fail:**<br>assert match_parens(...) | **Environment Binary Reward:**<br>0 |
| (d) Reflection | [...] tried to pick up the pan in stoveburner 1 [...] but the pan was not in stoveburner 1. [...] | [...] wrong because it only checks if the total count of open and close parentheses is equal [...] order of the parentheses [...] | [...] failed because I incorrectly assumed that they both had the same multiple professions [...] accurately identifying their professions. |
| (e) Next Trajectory | [...] **Action**: take pan 1 from stoveburner 2<br>[...] **Obs**: You put the pan 1 in countertop 1. | [...]<br>`    return 'Yes' if check(S1) or check(S2) else 'No'` | **Think**: [...] So the profession John Lanchester and Alan Dean Foster have in common is novelist.<br>**Action**: “novelist” |

Figure 1: Reflexion works on decision-making 4.1, programming 4.3, and reasoning 4.2 tasks.

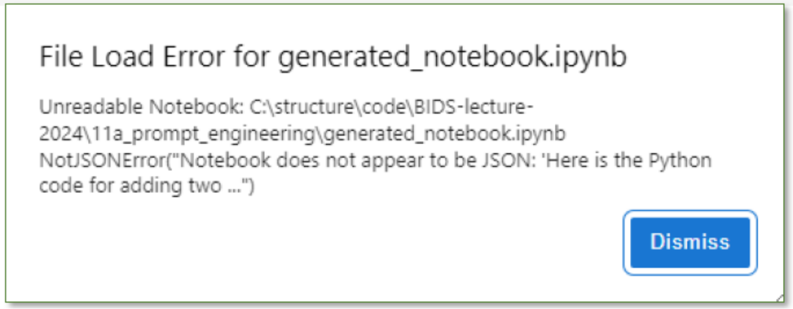
Cropped from Shinn et al. (2023) CC-BY 4.0
https://arxiv.org/abs/2303.11366

# Reflection Prompting

- Example task: Generate a Jupyter notebook

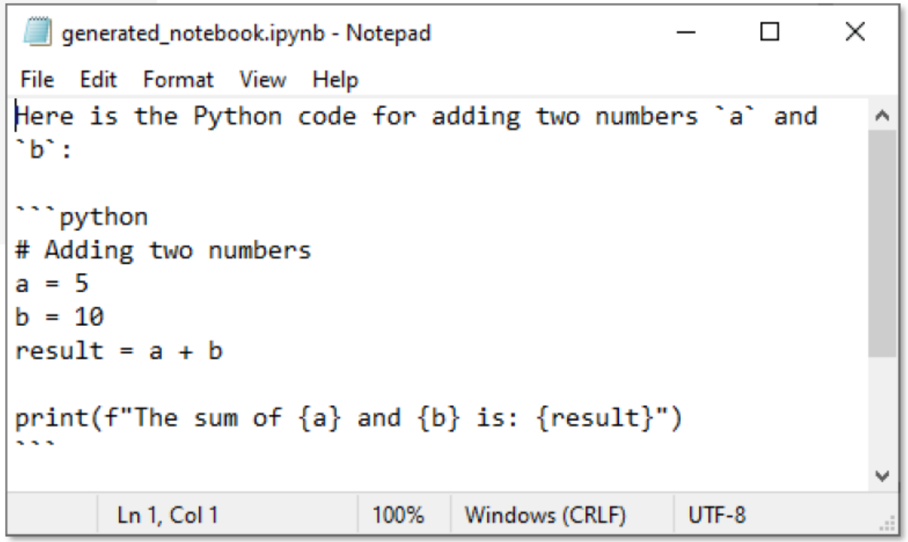


```
first_notebook = prompt("""
Write Python code for adding two numbers `a` and `b`.
Output it as Jupyter notebook in ipynb/json format.
""").strip("```json").strip("```")

first_file = "generated_notebook.ipynb"
with open(first_file, 'w') as file:
    file.write(first_notebook)
```



File Load Error for generated_notebook.ipynb

Unreadable Notebook: C:\structure\code\BIDS-lecture-2024\11a_prompt_engineering\generated_notebook.ipynb NotJSONError("Notebook does not appear to be JSON: 'Here is the Python code for adding two ...")

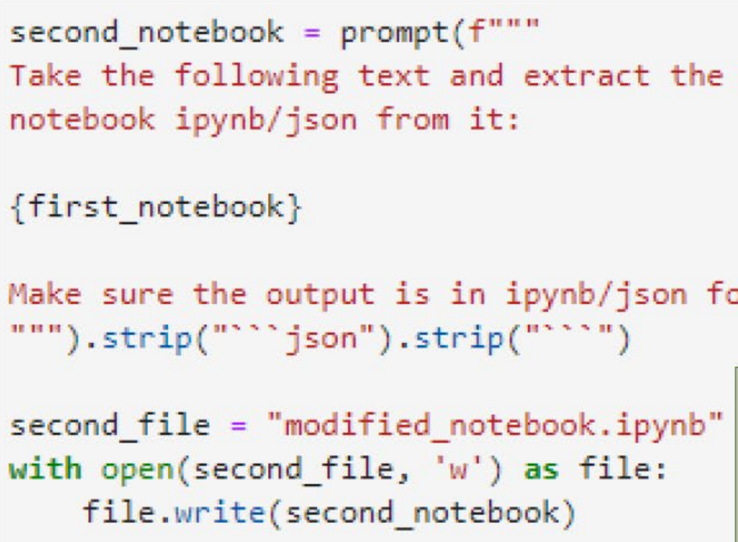


~~~
Here is the Python code for adding two numbers `a` and
`b`:

```python
# Adding two numbers
a = 5
b = 10
result = a + b

print(f"The sum of {a} and {b} is: {result}")
```
~~~

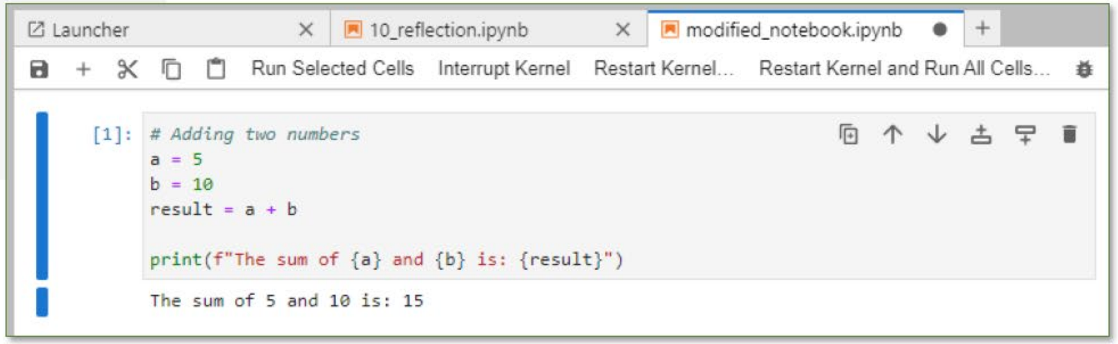
Cropped from Robert Haase, CC-BY 4.0 https://github.com/ScaDS/BIDS-lecture-2024/blob/main/11a_prompt_engineering/10_reflection.ipynb

# Reflection Prompting

- Example task: Generate a Jupyter notebook



```
second_notebook = prompt(f"""
Take the following text and extract the Jupyter
notebook ipynb/json from it:

{first_notebook}

Make sure the output is in ipynb/json format.
""").strip("```json").strip("```")

second_file = "modified_notebook.ipynb"
with open(second_file, 'w') as file:
    file.write(second_notebook)
```



```
# Adding two numbers
a = 5
b = 10
result = a + b

print(f"The sum of {a} and {b} is: {result}")
```

```
The sum of 5 and 10 is: 15
```

„Review and improve your own output“

Cropped from Robert Haase, CC-BY 4.0 https://github.com/ScaDS/BIDS-lecture-2024/blob/main/11a_prompt_engineering/10_reflection.ipynb

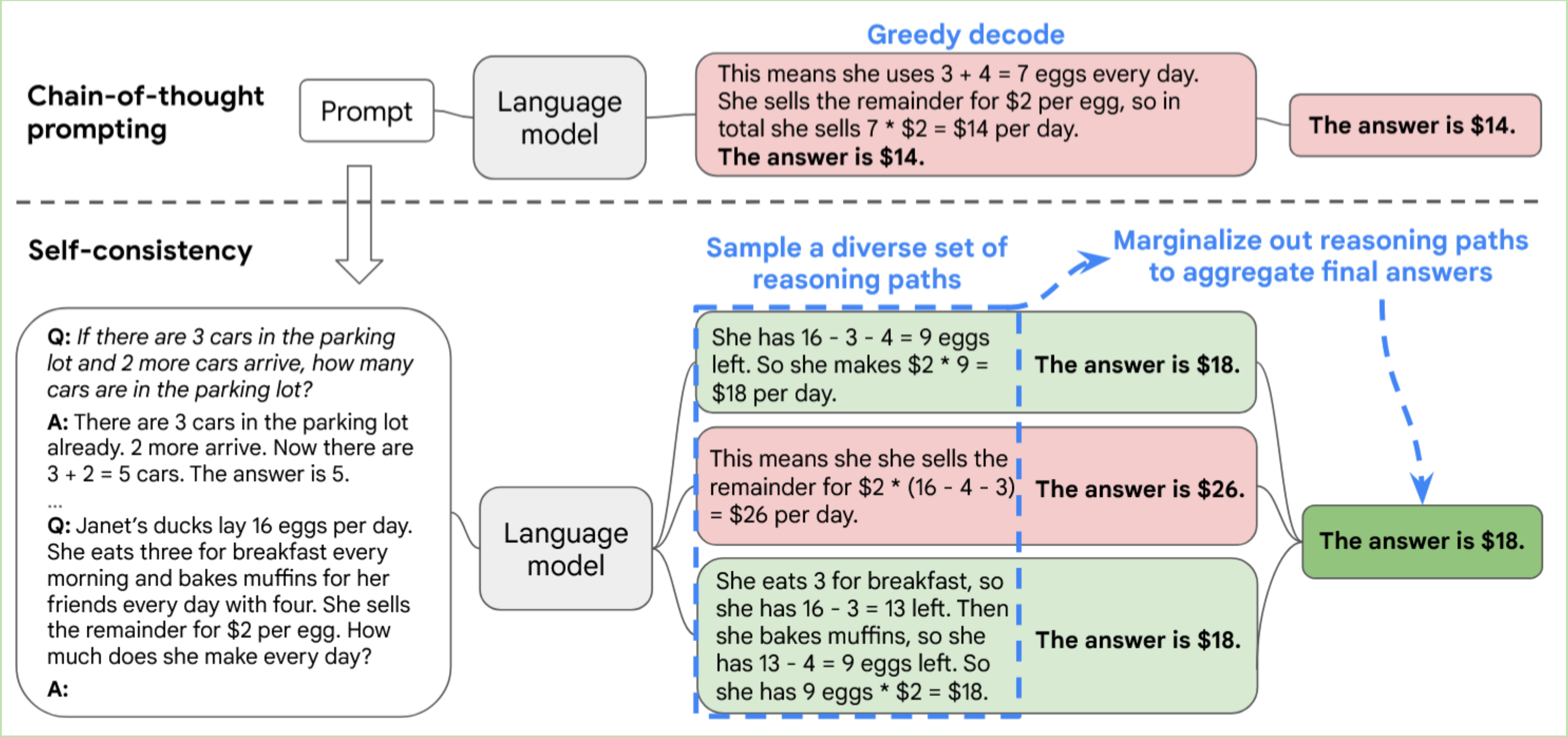
# Self-Consistency

**Idea**: Generate multiple answers & keep the most consistent result.

**Why**: Reduces random errors. Improves robustness by filtering out bad reasoning paths.

**Origin:** Wang et al. (2022) https://arxiv.org/abs/2203.11171

# Self Consistency



Greedy decode

Chain-of-thought prompting

Prompt

Language model

This means she uses 3 + 4 = 7 eggs every day. She sells the remainder for $2 per egg, so in total she sells 7 * $2 = $14 per day. **The answer is $14.**

**The answer is $14.**

Self-consistency

Sample a diverse set of reasoning paths

Marginalize out reasoning paths to aggregate final answers

**Q:** *If there are 3 cars in the parking lot and 2 more cars arrive, how many cars are in the parking lot?*
**A:** There are 3 cars in the parking lot already. 2 more arrive. Now there are 3 + 2 = 5 cars. The answer is 5.
...
**Q:** Janet's ducks lay 16 eggs per day. She eats three for breakfast every morning and bakes muffins for her friends every day with four. She sells the remainder for $2 per egg. How much does she make every day?
**A:**

Language model

She has 16 - 3 - 4 = 9 eggs left. So she makes $2 * 9 = $18 per day. | **The answer is $18.**

This means she she sells the remainder for $2 * (16 - 4 - 3) = $26 per day. | **The answer is $26.**

She eats 3 for breakfast, so she has 16 - 3 = 13 left. Then she bakes muffins, so she has 13 - 4 = 9 eggs left. So she has 9 eggs * $2 = $18. | **The answer is $18.**

**The answer is $18.**

„Generate multiple answers & keep the most consistent result“

Cropped from Zhang et al 2022,
licensed CC-BY 4.0 https://arxiv.org/abs/2203.11171

Photo by Helena Forde on Unsplash

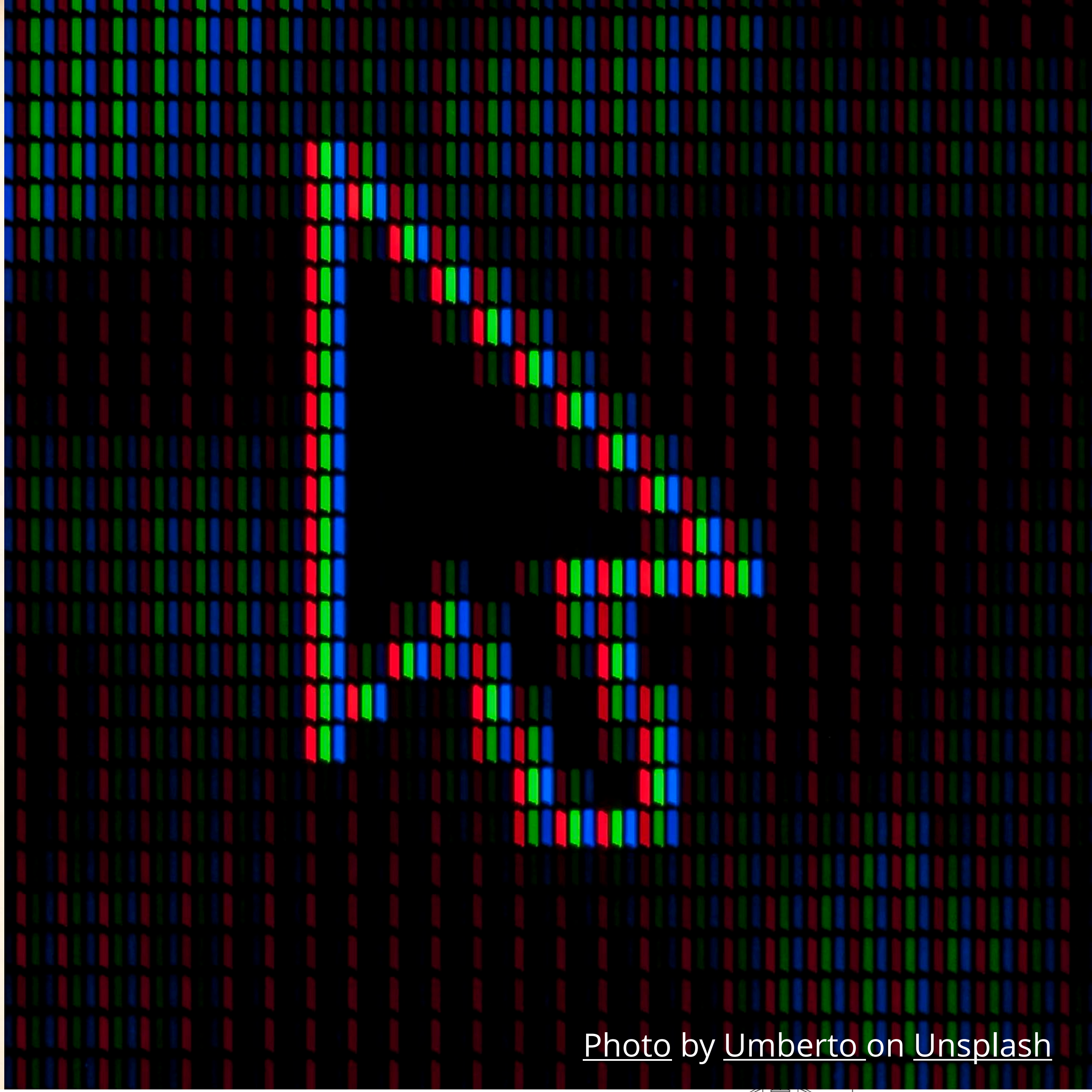
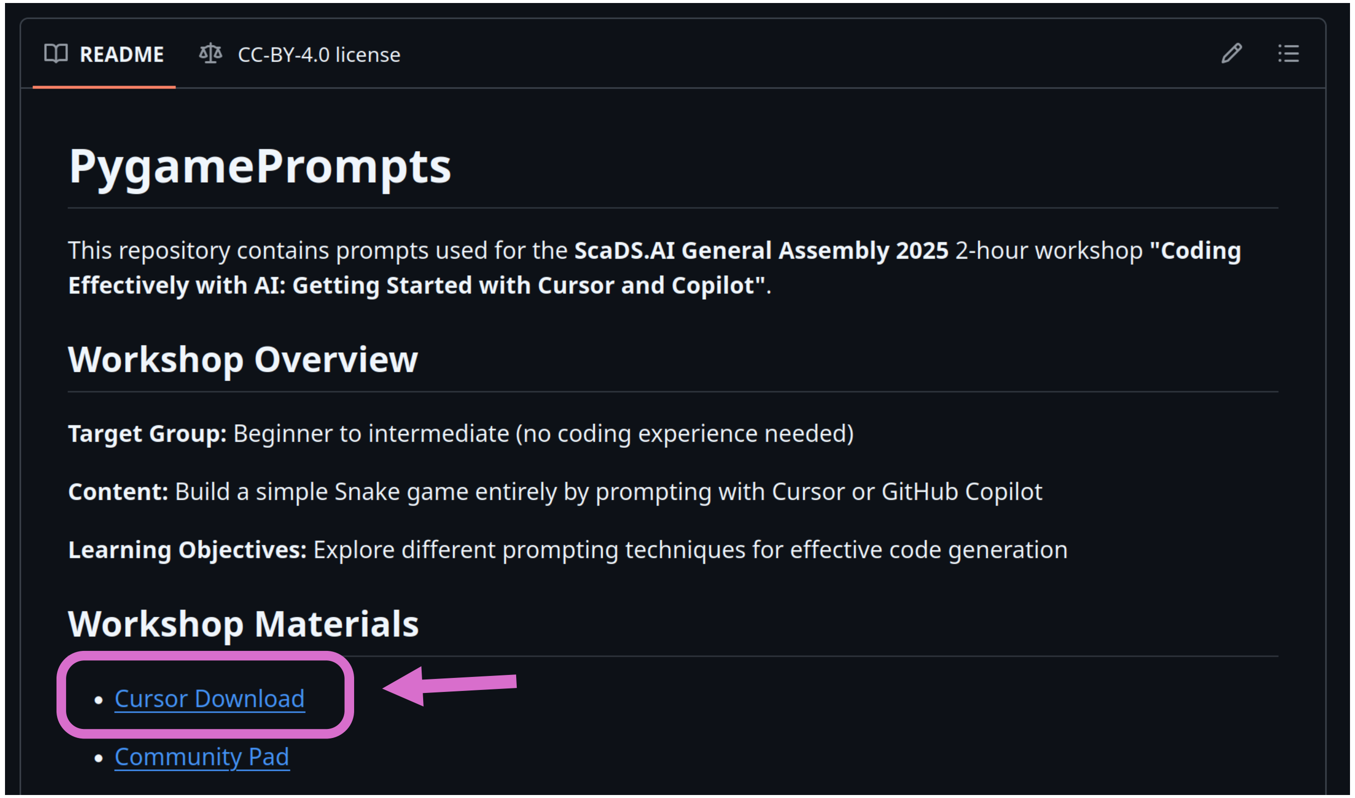


Photo by Umberto on Unsplash

# Go to Github...

https://github.com/kaabl/PygamePrompts



# PygamePrompts

This repository contains prompts used for the **ScaDS.AI General Assembly 2025** 2-hour workshop **"Coding Effectively with AI: Getting Started with Cursor and Copilot"**.

## Workshop Overview

**Target Group:** Beginner to intermediate (no coding experience needed)

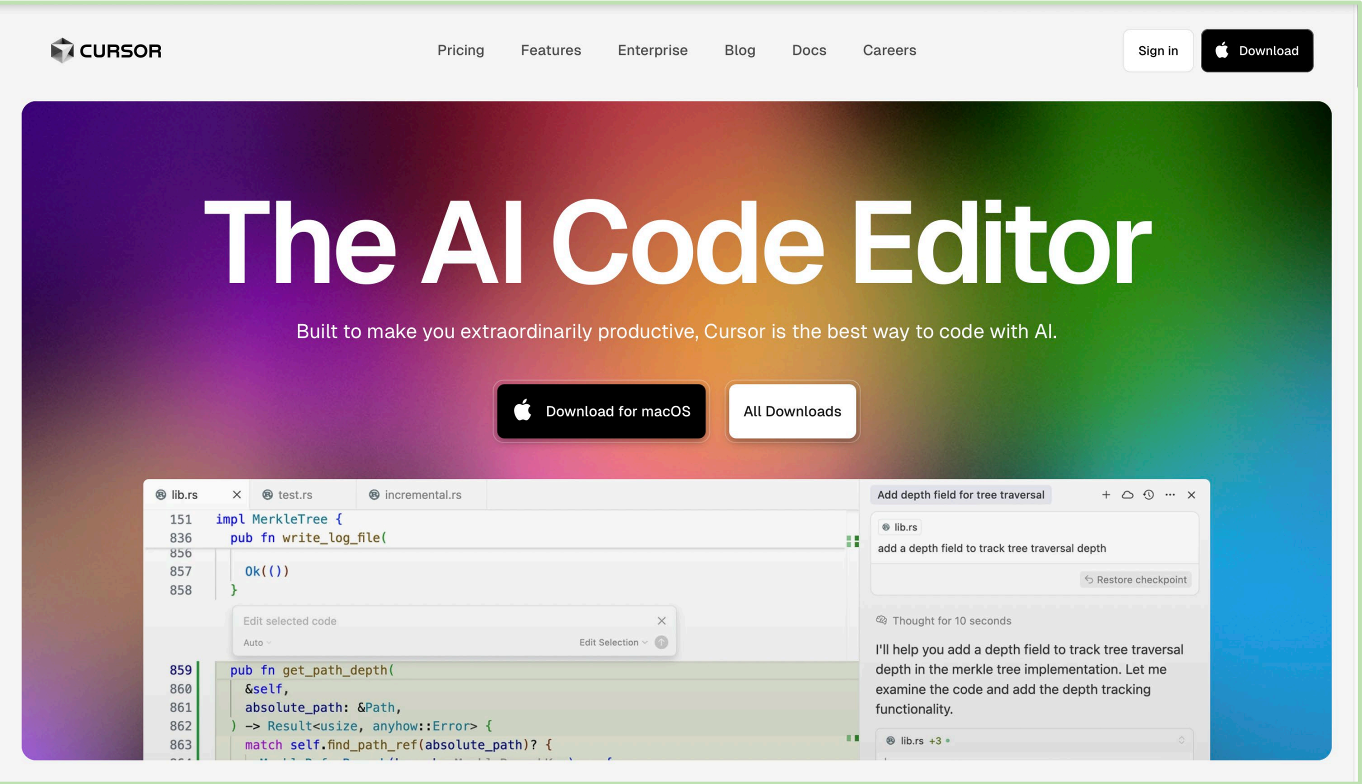
**Content:** Build a simple Snake game entirely by prompting with Cursor or GitHub Copilot

**Learning Objectives:** Explore different prompting techniques for effective code generation

## Workshop Materials

- Cursor Download
- Community Pad

# Cursor Installation (Mac)



The AI Code Editor

Built to make you extraordinarily productive, Cursor is the best way to code with AI.

```
impl MerkleTree {
  pub fn write_log_file(

    Ok(())
  }
  pub fn get_path_depth(
    &self,
    absolute_path: &Path,
  ) -> Result<usize, anyhow::Error> {
    match self.find_path_ref(absolute_path)? {
```

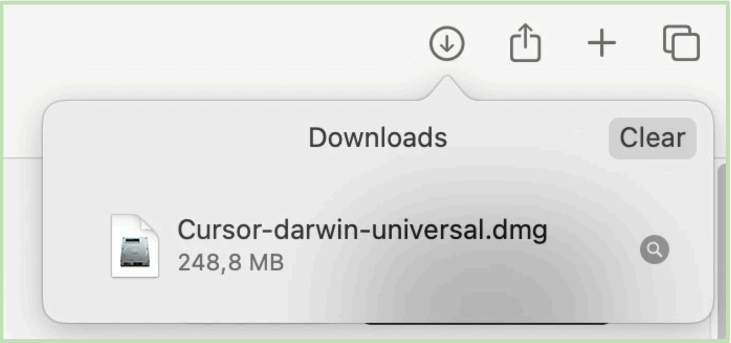
Add depth field for tree traversal

lib.rs

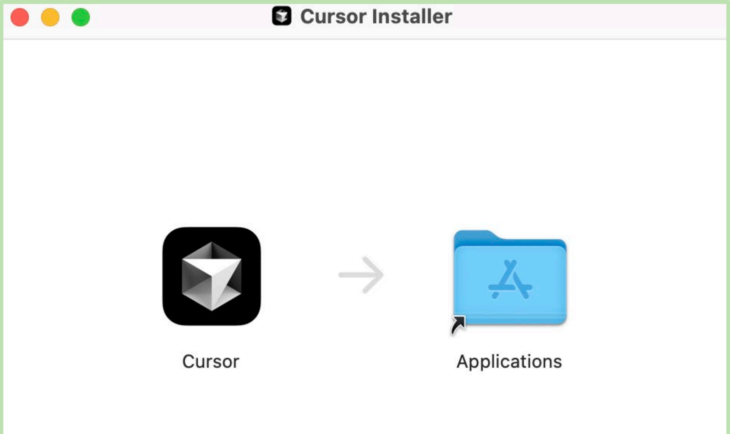
add a depth field to track tree traversal depth

Thought for 10 seconds

I'll help you add a depth field to track tree traversal depth in the merkle tree implementation. Let me examine the code and add the depth tracking functionality.

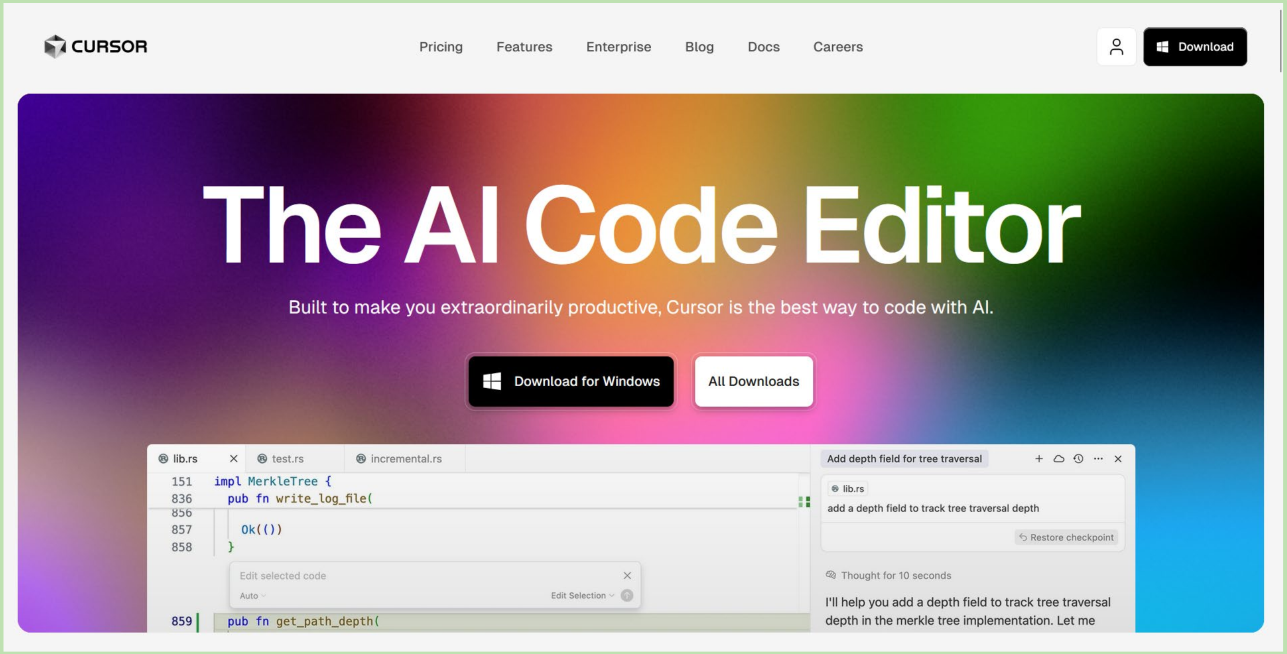


Cursor-darwin-universal.dmg
248,8 MB



Drag Cursor to Applications

# Cursor Installation (Windows)



The AI Code Editor

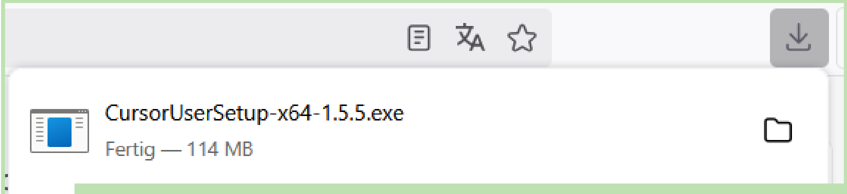
Built to make you extraordinarily productive, Cursor is the best way to code with AI.

```
impl MerkleTree {
  pub fn write_log_file(

    Ok(())
  }
```

```
  pub fn get_path_depth(
```

add a depth field to track tree traversal depth

I'll help you add a depth field to track tree traversal depth in the merkle tree implementation. Let me

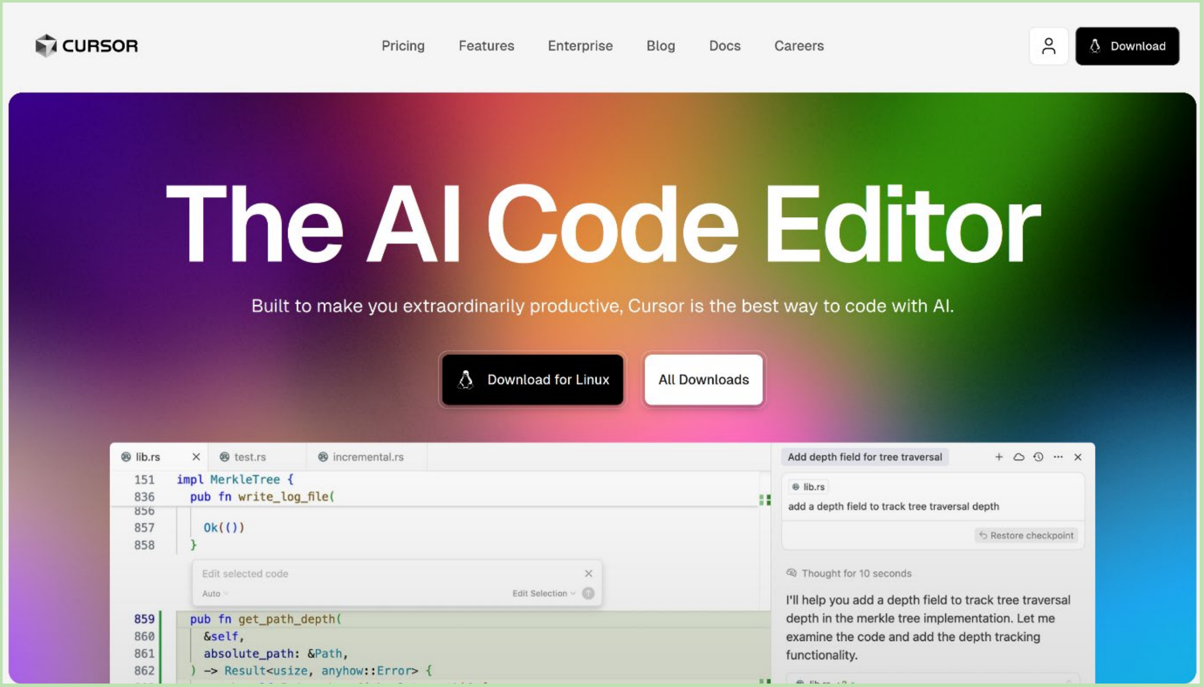


CursorUserSetup-x64-1.5.5.exe

Fertig — 114 MB

Follow instructions in the setup window

# Cursor Installation (Linux)



The AI Code Editor

Built to make you extraordinarily productive, Cursor is the best way to code with AI.

```
impl MerkleTree {
  pub fn write_log_file(

    Ok(())
  }

  pub fn get_path_depth(
    &self,
    absolute_path: &Path,
  ) -> Result<usize, anyhow::Error> {
```

Add depth field for tree traversal

lib.rs
add a depth field to track tree traversal depth

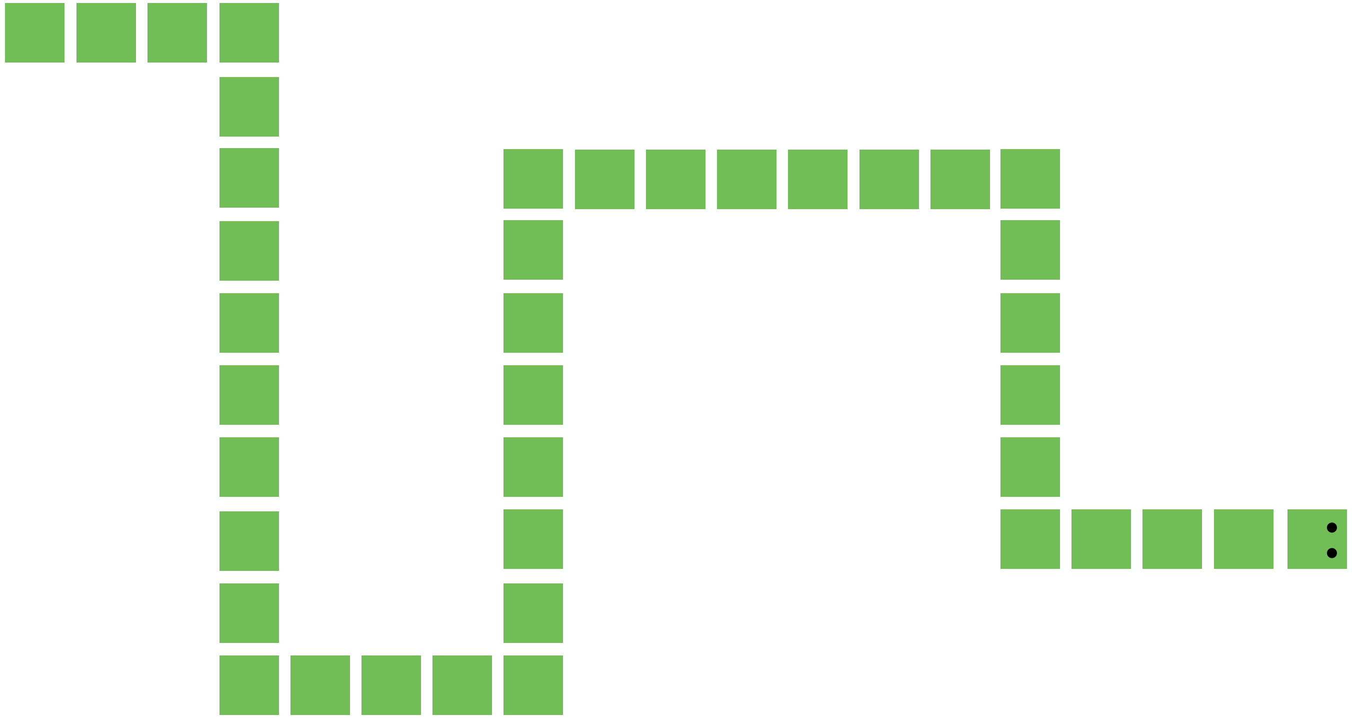
Thought for 10 seconds

I'll help you add a depth field to track tree traversal depth in the merkle tree implementation. Let me examine the code and add the depth tracking functionality.

Follow instructions in the GitHub repo

https://github.com/kaabl/PygamePrompts

# Practical



Lets build a snake game using cursor:

https://github.com/kaabl/PygamePrompts

# Pitfalls and Limitations

| Ethical & Legal Risks | Quality & Reliability | Bias & Integrity | Practical Limitations |
|---|---|---|---|
| **Data Privacy**<br>Entering information into AI tools can create risks of data leakage or misuse | **Hallucinations**<br>AI can produce code that looks valid but is incorrect, nonfunctional, or based on nonexistent functions/libraries | **Model Bias**<br>AI systems may reinforce stereotypes or reflect biases present in their training data | **Versioning & Compatibility**<br>Suggestions may include outdated or deprecated libraries that don't align with the current tech stack |
| **Licensing & Copyright**<br>AI-generated code may contain copyrighted snippets or unclear licensing, leading to potential legal issues | **Verification Needed**<br>Generated code should be reviewed, tested, and understood by the developer to ensure correctness and security | **Academic & Scientific Integrity**<br>Outputs may lack rigor, reproducibility, or proper citation unless these are deliberately enforced | **Overreliance on AI**<br>Depending too heavily on AI can hinder critical thinking, debugging skills, and long-term learning |

# Acknowledgements

ScaDS.AI DRESDEN LEIPZIG

TECHNISCHE UNIVERSITÄT DRESDEN

UNIVERSITÄT LEIPZIG

come2 data

Gefördert durch:

Bundesministerium
für Forschung, Technologie
und Raumfahrt

Finanziert von der
Europäischen Union
NextGenerationEU

NFDI4BIOIMAGE