KISSKI
KI-Servicezentrum für sensible
und kritische Infrastrukturen

Sascha Safenreider

KI-Methoden im akademischen Alltag: Programmieren mit KI

Part 1

Outline

1 Introduction to Vibe Coding

2 Principles of Vibe Coding

3 Conclusion

Experience

Do you have any Experience in (Vibe-)Coding?

Our Goal

- We developed a PRD
- We created an Application only using AI

What is Vibe Coding?

Vibe Coding is a new approach to programming that focuses on the creative and intuitive aspects of coding. It’s about capturing the essence of a project and bringing it to life through code.

Origin and Impact of Vibe Coding

Vibe Coding was first introduced by Andrej Karpathy, a researcher at OpenAI. It has since gained popularity and has been used in various projects, including the development of AI-powered games and simulations.

Origin and Impact of Vibe Coding

Andrej Karpathy @karpathy Feb 2, 2025

"There's a new kind of coding I call "vibe coding", where you fully give in to the vibes, embrace exponentials, and forget that the code even exists. It's possible because the LLMs (e.g. Cursor Composer w Sonnet) are getting too good. Also I just talk to Composer with SuperWhisper so I barely even touch the keyboard. I ask for the dumbest things like "decrease the padding on the sidebar by half" because I'm too lazy to find it. I "Accept All" always, I don't read the diffs anymore. When I get error messages I just copy paste them in with no comment, usually that fixes it. The code grows beyond my usual comprehension, I'd have to really read through it for a while. Sometimes the LLMs can't fix a bug so I just work around it or ask for random changes until it goes away. It's not too bad for throwaway weekend projects, but still quite amusing. I'm building a project or webapp, but it's not really coding - I just see stuff, say stuff, run stuff, and copy paste stuff, and it mostly works."

View on X

Vibe Coding Success Stories

- **René Turcios** – Won 200+ hackathons without writing a single line of code.
  - Uses only prompts with AI to build complete apps.
  - Former Yu-Gi-Oh! pro turned hackathon champion.
- **Lovable.dev Competition** – $40K prize, no coding required.
  - Built full apps in 24h with just natural language prompts.
  - Winning projects focused on creativity and clever prompt design.
  - The Output were over 18,000 Apps
- **Takeaway:** Prompting is the new programming — ideas win over syntax, but...

When Vibe Coding Goes Wrong

- **45%** of AI-generated code contains exploitable security vulnerabilities.
- Tested: 80 tasks in different programming languages.
- Model size has **no** impact on security.
- **Conclusion:** Human security review remains essential.
  - AI tools speed up development but do not replace critical security assessment.

Source

When Vibe Coding Goes Wrong

Examples

- Paying customers lose access or get wrong content
- Accounts hacked, data stolen
- Random things happening:
  - Maxed API usage
  - Subscription bypass
  - Random database writes
- App taken down

Quiz Question 1

What is the primary focus of Vibe Coding?

- Efficiency and productivity
- Creativity and intuition
- Technical skills and expertise
- Project management and planning

Quiz Question 1

What is the primary focus of Vibe Coding?

- **Efficiency and productivity**
- **Creativity and intuition**
- Technical skills and expertise
- Project management and planning

Outline

1 Introduction to Vibe Coding

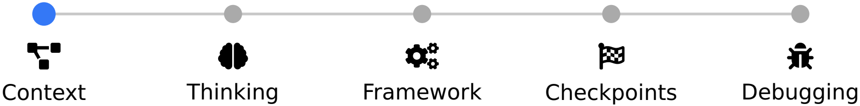
2 Principles of Vibe Coding

3 Conclusion

5 Fundamental Skills of Vibe Coding

- **Context:** Understanding the full picture before you start coding
- **Thinking:** Understanding the problem and identifying the key elements.
- **Framework:** Choosing the right framework and tools for the project.
- **Checkpoints:** Setting milestones and tracking progress.
- **Debugging:** Identifying and fixing errors.

5 Fundamental Skills of Vibe Coding



Context

Thinking

Framework

Checkpoints

Debugging

5 Fundamental Skills of Vibe Coding: Context

Understanding the full picture before you start coding

- Context means knowing what you’re building, for whom, and why
- Good context gathering prevents building the wrong thing
- Tools: Mind maps and Product Requirements Documents (PRD)
- The clearer your context, the better your AI results

5 Fundamental Skills of Vibe Coding: Context

Next Step: Product Requirements Document (PRD)

A PRD is a document that outlines the requirements and specifications of a project.

Important Recommendation

Spend a lot of time on this part!

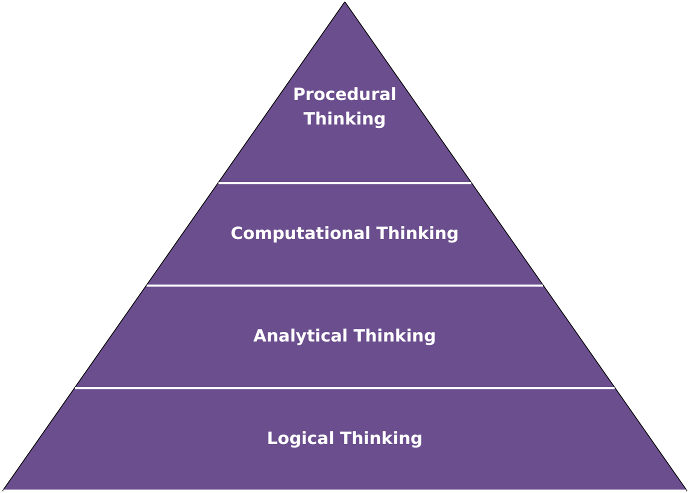
Take time to have a clear vision and figure out what you want.

- The PRD transforms your mind map into actionable requirements
- It’s essential to ensure that the project is well-defined and achievable
- Time invested here pays off in better AI results

5 Fundamental Skills of Vibe Coding: Context

Use Case: PRD - The Thinking Pyramid

How do we use *THINKING* for our PRD?



Procedural Thinking

Computational Thinking

Analytical Thinking

Logical Thinking

5 Fundamental Skills of Vibe Coding: Context

Use Case: PRD Step 1 - Logical Thinking

Project Overview (Logical Thinking)

- The goal of this project is to develop an AI-powered system that creates personalized weekly meal plans for casual households
- The system will take into account factors such as work schedule, cooking skills, dietary restrictions, food preferences, and grocery budget
- It will also consider available kitchen equipment and pantry inventory to ensure the meal plan is practical and efficient
- For users with limited cooking experience or literacy, the plan will include visual recipe cards to make preparation easier

5 Fundamental Skills of Vibe Coding: Context

Use Case: PRD Step 2 - Analytical Thinking

Skills Required (Analytical Thinking)

- Examples:
  - My programming language (e.g. Python)
  - Recipe and Nutrition Data Processing
  - API Key to a certain AI model
  - Image Processing (for visual recipe cards)
  - UI development
    - Maybe in addition: More specific Frontend/Backend stack

5 Fundamental Skills of Vibe Coding: Context

Use Case: PRD Step 3 - Computational Thinking

Key Features (Computational Thinking)

- **Milestone 1:** Personalized Meal Plan Engine
  - AI-Based Meal Plan Generation:
    - Generate weekly meal plans based on user preferences, dietary needs, available cooking time, and budget
    - Suggest ingredient substitutions based on pantry inventory and local grocery availability
- **Milestone 2:** Contextual Customization:
  - Adapt instructions based on user cooking skill level and literacy—provide visual recipe cards for beginners or busy users who prefer quick visual scanning
  - Recommend batch cooking or quick-prep options for users with especially busy weeks

5 Fundamental Skills of Vibe Coding: Context

Use Case: PRD Step 4 - Procedural Thinking

Detailed Context (Procedural Thinking)

- User lifestyle data: work schedule, cooking skills, dietary restrictions, preferences, grocery budget, available equipment
- **Client Information:**
  - The client is a lifestyle-focused health coach based in Germany
  - They work primarily with urban professionals in high-paced careers who want to maintain a healthy diet without excessive time spent on meal planning or cooking
  - The client’s focus is on improving daily nutrition habits through simple, personalized, and practical solutions that fit real-world schedules

5 Fundamental Skills of Vibe Coding: Context

Key Insight

The clearer your vision is

↓

The clearer the PRD is

↓

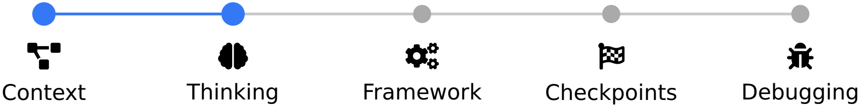
The better result you will get from the AI

5 Fundamental Skills of Vibe Coding: Context

Now we are ready to merge our PRD Steps into one document:

- **Project Overview (Logical Thinking)**
  - The goal of this project is to develop an AI-powered system that creates personalized weekly meal plans for casual households
  - The system will take into account factors such as work schedule, cooking skills, dietary restrictions, food preferences, and grocery budget
  - It will also consider available kitchen equipment and pantry inventory to ensure the meal plan is practical and efficient
  - For users with limited cooking experience or literacy, the plan will include visual recipe cards to make preparation easier
- **Skills Required (Analytical Thinking)**
  - Python
  - Recipe and Nutrition Data Processing
  - API Key
  - Image Processing (for visual recipe cards)
  - UI development (optionally: more specific Frontend/Backend stack)
- **Key Features (Computational Thinking)**
  - **Milestone 1:** Personalized Meal Plan Engine
    - Generate weekly meal plans based on user preferences, dietary needs, available cooking time, and budget
    - Suggest ingredient substitutions based on pantry inventory and local grocery availability
  - **Milestone 2:** Contextual Customization
    - Adapt instructions based on user cooking skill level and literacy
    - Provide visual recipe cards for beginners or busy users who prefer quick visual scanning
    - Recommend batch cooking or quick-prep options for users with especially busy weeks
- **Detailed Context (Procedural Thinking)**
  - User lifestyle data: work schedule, cooking skills, dietary restrictions, preferences, grocery budget, available equipment
  - **Client Information:**
    - The client is a lifestyle-focused health coach based in Germany
    - They work primarily with urban professionals in high-paced careers who want to maintain a healthy diet without excessive time spent on meal planning or cooking
    - The client’s focus is on improving daily nutrition habits through simple, personalized, and practical solutions that fit real-world schedules

5 Fundamental Skills of Vibe Coding



Context

Thinking

Framework

Checkpoints

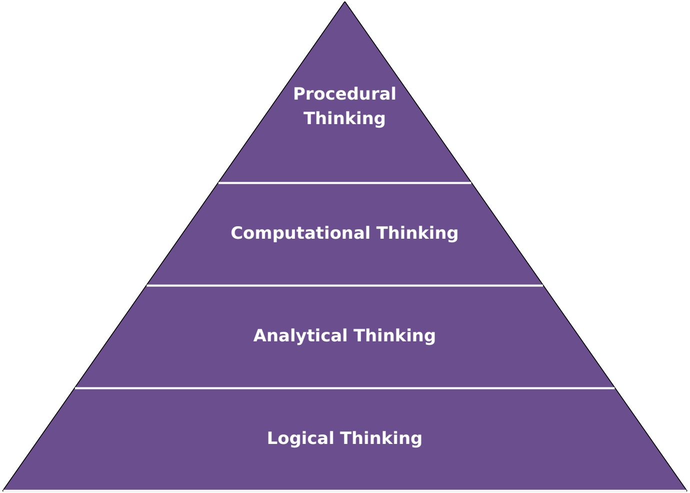
Debugging

5 Fundamental Skills of Vibe Coding: Thinking

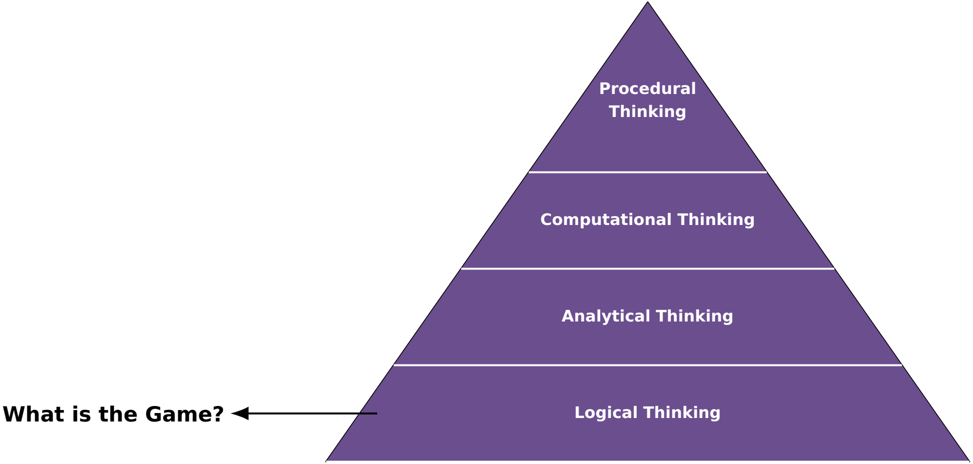
Now we are ready for our own App idea »

5 Fundamental Skills of Vibe Coding: Thinking

How do we think about, i.e., a Chess Game App?



5 Fundamental Skills of Vibe Coding: Thinking



Procedural Thinking

Computational Thinking

Analytical Thinking

Logical Thinking

What is the Game?

5 Fundamental Skills of Vibe Coding: Thinking

App Inspiration

- 🎮 **Gamified app** – XP, levels, challenges
- **Health app** – fitness, meals, mindfulness
- **Social app** – connect, share, challenge friends
- 🤖 **AI buddy** – chatbot, coach, creative helper
- 📅 **Productivity app** – habits, focus, reminders

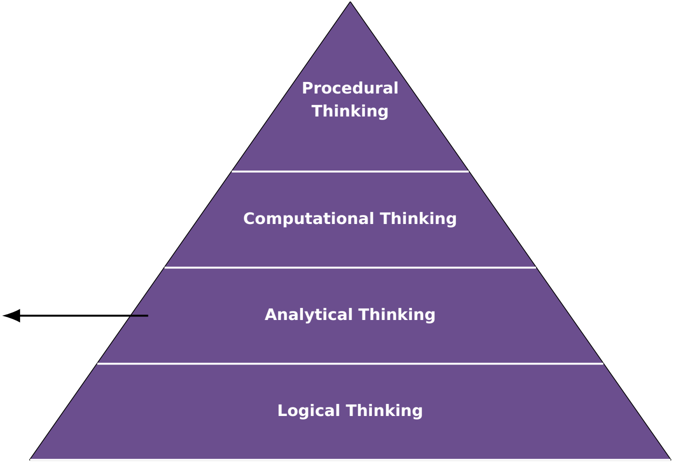
Interactive: Logical Thinking

Please answer these questions for your own project idea (🕒 15 Min):

- What is the main goal of your project?
- Who will use it?
- What core problem does it solve?
- What makes it valuable or unique?

5 Fundamental Skills of Vibe Coding: Thinking

Main objective and goal of this game
How do I play this game?



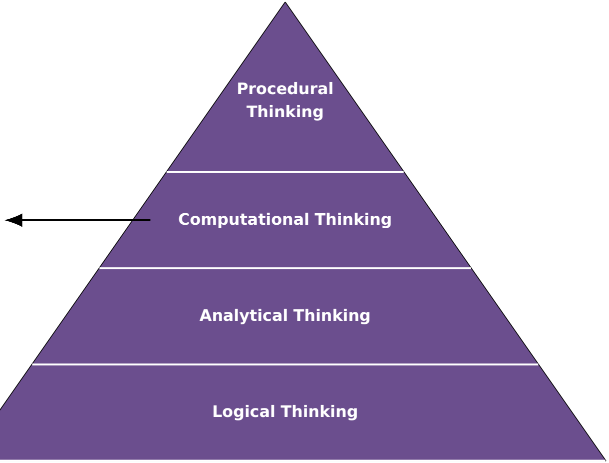
Interactive: Analytical Thinking

Identify the skills, technologies, and resources your project will need (🕒 15 Min):

- Which programming languages or tools are required?
- Which AI/ML models or APIs might be used?
- Do you need data processing, image handling, or other specialized skills?
- Which skills are in your team, and which need to be learned or outsourced?

5 Fundamental Skills of Vibe Coding: Thinking

How to fit the logic of the game into a complicated set of problems?



Procedural Thinking

Computational Thinking

Analytical Thinking

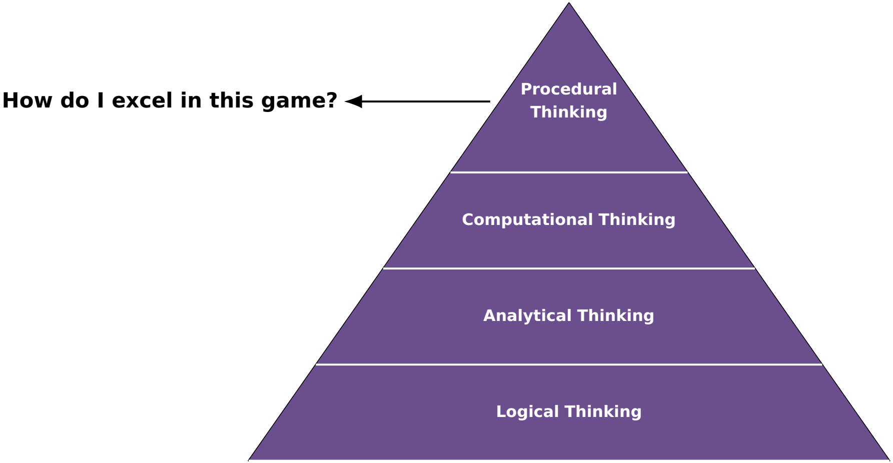
Logical Thinking

Interactive: Computational Thinking

Break down your project into key milestones and features (🕒 15 Min):

- What are the main functional components?
- How will the system process input and generate output?
- What would be Milestone 1 and Milestone 2?
- Which features are “must-haves” vs. “nice-to-haves”?

5 Fundamental Skills of Vibe Coding: Thinking



How do I excel in this game?

Procedural Thinking

Computational Thinking

Analytical Thinking

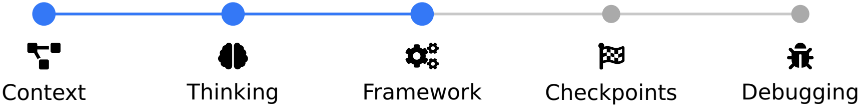
Logical Thinking

Interactive: Procedural Thinking

Add context and constraints for realistic planning (🕓 15 Min):

- Who is the client or target audience?
- What is their daily reality (time, budget, skills)?
- What equipment, data, or infrastructure is available?
- Are there legal, cultural, or accessibility requirements?

5 Fundamental Skills of Vibe Coding



Context

Thinking

Framework

Checkpoints

Debugging

5 Fundamental Skills of Vibe Coding: Framework

Guide the AI to the right tools

Why?

If you clearly tell the AI *what* tools to use, it can give you better fitting examples and instructions.

- **What am I building?** (e.g., a simple web app)
- **Which tools?** (e.g., React, HTML/CSS/JavaScript)
- Optional: **Any extras?** (e.g., animations, special styling)

Example prompt

```
I want to create a small web app with React.
Use HTML, CSS, and JavaScript as the base, plus Tailwind CSS for styling.
Start with a homepage that shows a title and a "Start" button.
```

5 Fundamental Skills of Vibe Coding: Framework

Name frameworks and packages

- **Frontend:** HTML, CSS, JavaScript
- **UI framework:** React
- **Styling:** Tailwind CSS
- **(Optional) Animation/3D:** Three.js

Example prompt

```
Help me set up a simple one-page app.
Use React for the interface and Tailwind CSS for quick styling.
Make the page show a welcome message in large text.
```

5 Fundamental Skills of Vibe Coding: Framework

Name frameworks and packages

- If animation is needed: name the tool (e.g., **Three.js**)
- Describe exactly what should move or change

Example prompt

```
Add a simple animation using Three.js:
Show a spinning cube in the center of the page.
Explain each step so I can follow along.
```

5 Fundamental Skills of Vibe Coding: Framework

Not sure which tool to use? Drag & Drop as example

How to ask the AI

Ask for options, short explanations, and a small working example.

- Common React libraries for drag & drop:
  - `@dnd-kit/core` — flexible and modern
  - `react-beautiful-dnd` — great for lists/boards
  - `react-dropzone` — for file upload areas
- Try one option and see how it works, then choose whatever you prefer

Example prompt

```
I want to add drag-and-drop to my React app.
Suggest a good library and explain why it’s a good choice.
Show me a simple example with two lists where I can move items.
```

5 Fundamental Skills of Vibe Coding: Framework

Minimum web app knowledge

- **Frontend:** What the user sees and clicks (e.g., React)
- **Backend:** The “behind-the-scenes” part (logic and data)
- **Communication:** The frontend sends questions and gets answers from the backend
- **Popular tools:** React (UI), Tailwind CSS (styling)

Example prompt

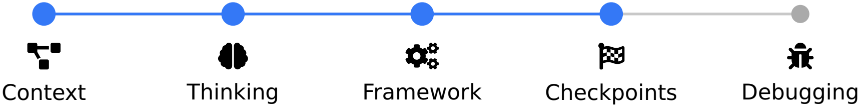
```
Explain frontend, backend, and API in simple terms.
Then show a small React example that loads data from a free API
and displays it as a list on the page.
```

5 Fundamental Skills of Vibe Coding: Framework

Build and learn with the AI

- Build small features one at a time
- Ask the AI to explain things simply while showing examples
- Test, adjust, then move to the next feature
- This way, you learn while your project grows

5 Fundamental Skills of Vibe Coding



Context

Thinking

Framework

Checkpoints

Debugging

5 Fundamental Skills of Vibe Coding: Checkpoints

Why checkpoints and version control matter

Key idea

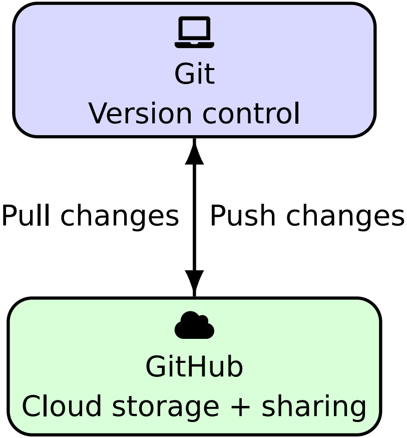
Things will break — and that’s normal. Checkpoints let you save working versions of your project so you can always go back.

- You can save your project’s current state
- If today’s changes break things, you can restore yesterday’s version
- Helps you try new ideas without fear

5 Fundamental Skills of Vibe Coding: Checkpoints

Version Control: Git vs GitHub

- 💻 **Git** — Tool on your computer to track changes
- ☁ **GitHub** — Online place to store and share your Git projects



5 Fundamental Skills of Vibe Coding: Checkpoints

Using Git: First steps

- Download Git: `https://git-scm.com/downloads`

Basic commands

`git init` — Start version control in this folder

`git add .` — Mark all files to save

`git commit -m ”initial commit”` — Save the changes with a message

`git push` — Send to GitHub (needs setup)

5 Fundamental Skills of Vibe Coding: Checkpoints

Ask the AI to help

- Many IDEs (like VS Code) have AI assistants
- You can ask the AI to:
  - Commit your changes
  - Push them to GitHub
  - Explain Git errors
- Saves time and avoids typos

Example prompt

```
Please commit my changes with the message "added login page"
and push them to the main branch on GitHub.
```

5 Fundamental Skills of Vibe Coding: Checkpoints

Quiz Question 2

Why are checkpoints (like Git commits) important in Vibe Coding?

- They help you roll back to working versions
- They automatically debug all errors
- They generate new prompts for the AI
- They make the app run faster

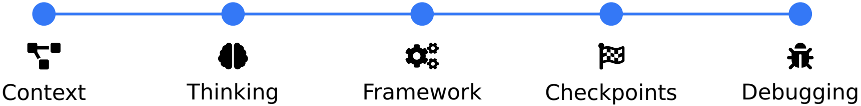
5 Fundamental Skills of Vibe Coding: Checkpoints

Quiz Question 2

Why are checkpoints (like Git commits) important in Vibe Coding?

- **They help you roll back to working versions**
- They automatically debug all errors
- They generate new prompts for the AI
- They make the app run faster

5 Fundamental Skills of Vibe Coding



Context

Thinking

Framework

Checkpoints

Debugging

5 Fundamental Skills of Vibe Coding: Debugging

Finding and fixing problems

Key idea

🔍 Debugging is about finding out what’s wrong and fixing it step by step.

- Identify:
  - 🔍 Where the problem is
  - ⚠ What the problem is
- Try different solutions until it works 🔧

Example prompt

```
My app crashes when I click the login button.
Help me find where the error is and fix it.
```

5 Fundamental Skills of Vibe Coding: Debugging

Using error messages

- Error messages are clues — read them carefully
- Don’t get stuck on the same error message
- If one fix doesn’t work, try a different approach

Example prompt

```
Here’s the error I get: ”Cannot read property ’map’ of undefined”.
What does this mean and how can I fix it?
```

5 Fundamental Skills of Vibe Coding: Debugging

Providing more context

- If prompts or PRDs aren’t enough, give more details
- Share:
  - Mock-ups
  - Additional data
  - Descriptions of what you expect vs. what happens

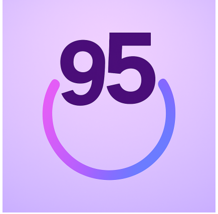
Example prompt

```
The app should show a blue header but it’s white.
Here’s a screenshot of what I mean, and my CSS file.
```

5 Fundamental Skills of Vibe Coding: Debugging

Screenshots help

- A picture makes the problem much clearer
- Highlight exactly what’s wrong
- Example: Overlapping elements



Example prompt

```
The percentage and the Loading Circle are overlapping – how can I fix it?
Here’s my HTML and CSS.
```

Outline

1 Introduction to Vibe Coding

2 Principles of Vibe Coding

3 Conclusion

Conclusion Part 1

- We learned some Success Stories, but also Risks of **Vibe Coding**
- Learned the **5 fundamental skills**:
  → Context, Thinking, Framework, Checkpoints, Debugging
- Practiced with interactive sessions and built first PRD drafts
- Take Home Message: **Clear vision** → **Clear PRD** → **Better AI results**

Outlook Part 2

- Deep dive into **PRD creation and refinement**
- Hands-on: **Build and test your own Vibe Coding projects**
- Explore and evaluate **different AI coding tools** together!
- Share insights: Which vibes, tools, and approaches really work?

Please save your today’s work! :)
→ We will need it for the next session!

More Content:

Simple introduction

https://www.youtube.com/watch?v=-LFB8D9WV-g

Comprehensive explanation

https://www.youtube.com/watch?v=mTE2BFXfQnw

Slightly more advanced

https://www.youtube.com/watch?v=c3CVVXYN0MU