Vibe Programming

```
module = Module(
    code="ELEE1149",
    name="Software Engineering",
    credits=15,
    module_leader="Seb Blair BEng(H) PGCAP MIET MIHEEM FHEA"
)
```

UNIVERSITY OF GREENWICH

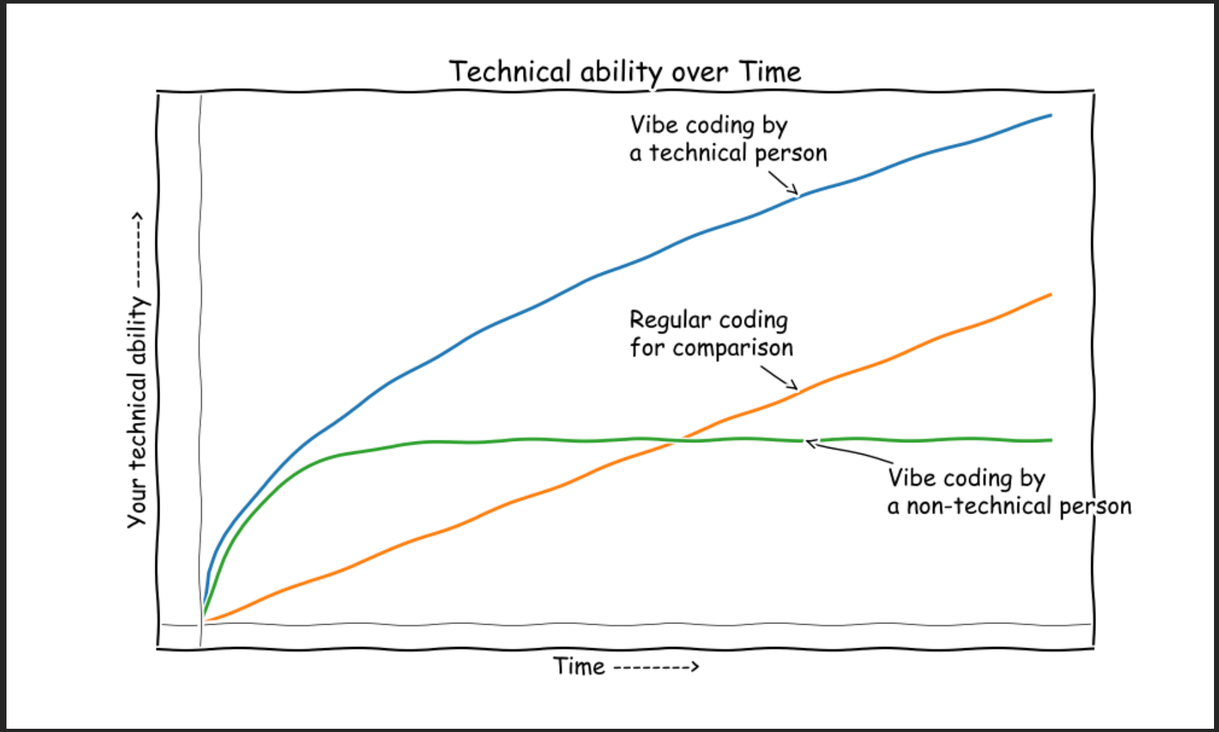
Download as a PDF

What is it?

> Vibe [programming] (OpenAI, Andrej Karpathy, 2025)
>
> - 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

- Vibe programming refers to AI-augmented or AI-dependent programming workflows where the user relies on natural language prompts, generated code snippets, and automated completions to develop software systems, often without fully understanding the underlying code mechanics.

Do I need to learn programming?



Impacts | Costs

Impact Area	Why It's Critical
Energy consumption	Datacenter electricity use is skyrocketing; clean grid only partly mitigates the impact
Hardware emissions	Manufacturing GPUs, memory, and specialized hardware generates huge embodied emissions
Water + resources	Cooling, semiconductor fab, and supply chains intensify water and raw material demands
Systemic rebound	Efficiency alone leads to more demand, not less environmental impact
Governance gap	Lack of unified regulation allows unchecked, unsustainable expansion

Year	Global Datacenter Electricity Use (TWh/ year)	Datacenter Power Capacity (MW)
2010	~194 TWh	
2018	~204 TWh (despite ~6× increase in capacity)	2,688 MW
2022	~460 TWh	5,341 MW (nearly doubled)
2026 (proj.)	620–1,050 TWh (base ~800 TWh)	+12,000 MW (global)

Google reported in 2021 that its U.S. datacenters consumed 12.7 billion liters of water*

*On average person in the UK uses 147 Literes per day, a population of circa 69 million uses 10.9 Billion liters of water.

https://mit-genai.pubpub.org/pub/8ulgrckc/release/2

Model	Approximate Training Energy Use
GPT-3	~1,287 MWh (≈ energy use of ~120 U.S. homes/year)
GPT-4 (likely)	Higher (exact not disclosed, but >2× GPT-3 likely)

Unit	CO_2 Emissions (Estimate)
Single GPT query (simple)	~2–4 g CO_2
Single GPT query (complex, multi-turn)	~10–50 g CO_2

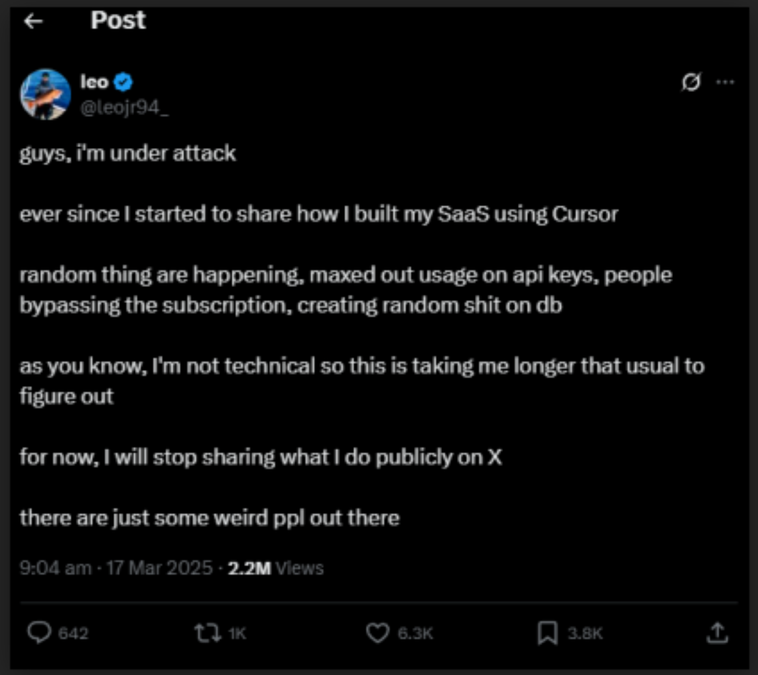
Costly Errors

Last week, we asked Devin to make a change.
It added an event on the banner component mount, which caused 6.6M posthog events in one week, which will cost us $733.
Devin costs $500 + $733 = $1273

Lesson - Review AI-generated code multiple times

Anurag Bhagsain (2025)

https://x.com/abhagsain/status/1876362355870994538

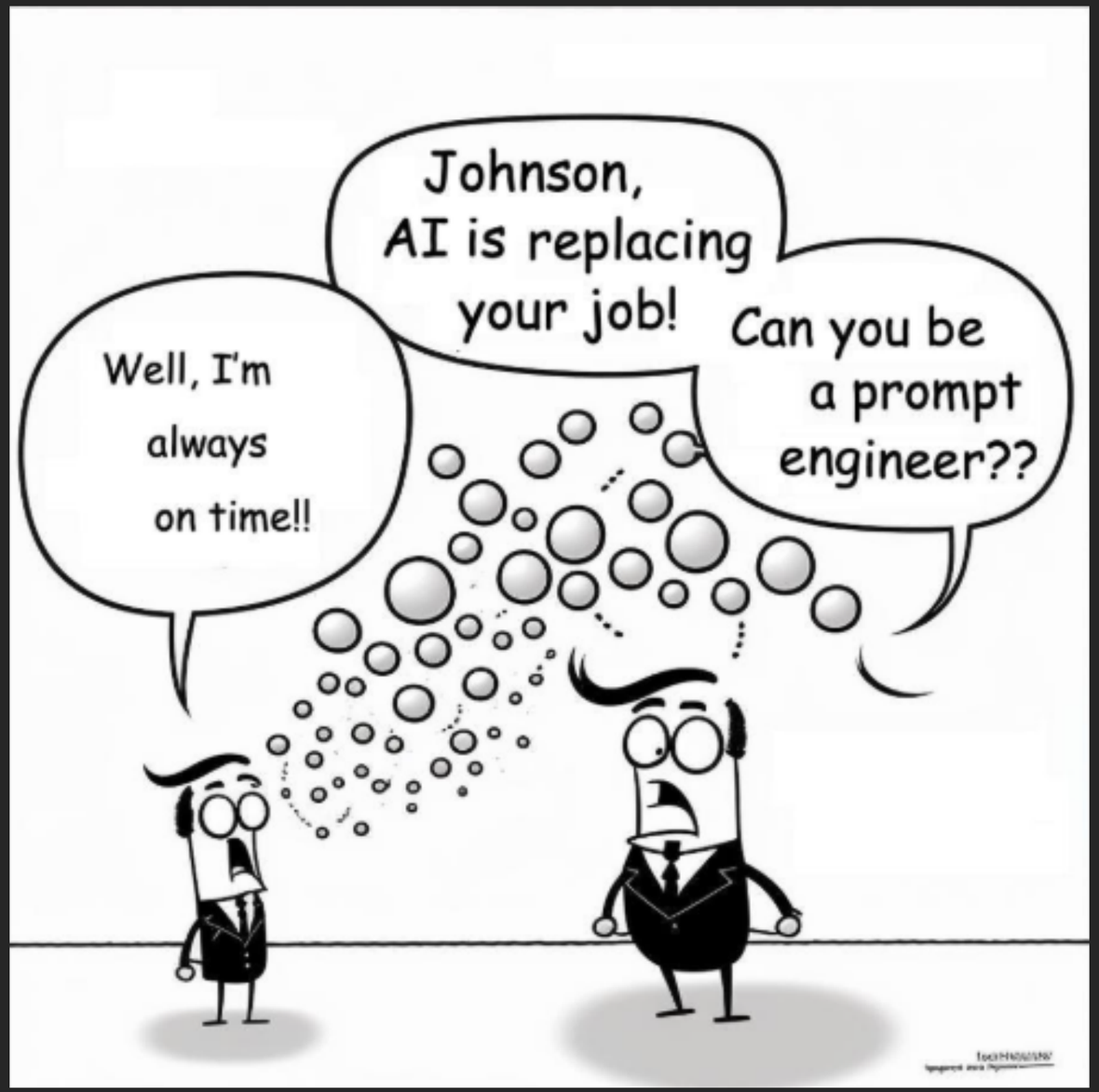


https://x.com/leojr94_/status/1901560276488511759

Limits

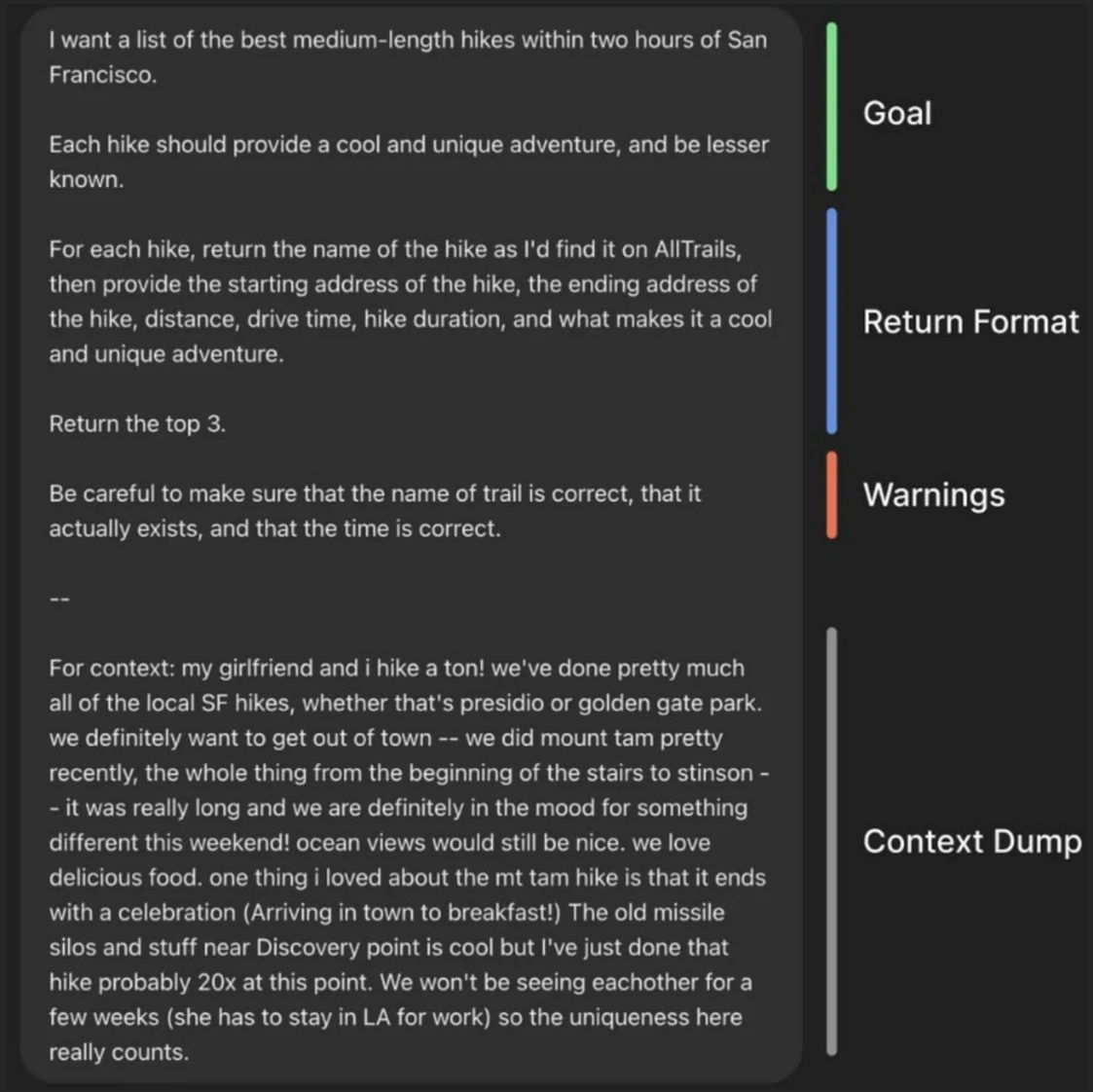
- Generated code ≠ correct code.
- The models don’t reason — they pattern-match statistically.
- Warning: AI-generated code must be reviewed, tested, and integrated responsibly. Blind trust leads to failure.

Prompt Engineering



The Anatomy of an o1 Prompt

- Set clear context
- Give specific instructions
- Define the output format
- Include warnings (what to avoid)
- Use a context dump when needed



https://x.com/gdb/status/1878489681702310392

- **Business & Marketing**
  - Old Prompt: “Give me some marketing ideas.”
  - o1 Prompt: “You are a marketing consultant for a tech startup launching a new fitness app. Provide three innovative campaign ideas with target demographics, a brief budget estimate, and potential ROI. Format your answer in bullet points.”

- **Education & E-Learning**
  - Old Prompt: “Explain photosynthesis.”
  - o1 Prompt: “Act as a middle-school science teacher preparing a 10-minute lesson on photosynthesis. Include a simple experiment students can do at home, three key vocabulary words, and a 5-question quiz at the end. Keep the tone friendly and age-appropriate.”

- **Creative Writing**
  - Old Prompt: “Write a short story about dragons.”
  - o1 Prompt: “Write a 300-word short story set in a futuristic city, featuring dragons that coexist with humans. Focus on the emotional tension between humans and dragons. Conclude with a cliffhanger.”

Prompt: here is my python script, I want to change code base to rust: [Copilot]

Python

```python
import time

password = input('Enter Password'):
start = time.time()

chars = 'abcdefghijklmnopqrstuvwxyz'
guess = []

for val in range(5):
  a = [i for i in chars]
  for y in range(val):
    a = [x+i for i in chars for x in a]
  guess = guess+a

  if password in guess:
    break
end = time.time()
clock = str(end - start)

print('Your password: ' + password)
print('Time taken: ' + clock)
```

rust

```rust
use std::time::Instant;
use std::io::{self, Write};

fn main() {

    print!("Enter Password: ");
    io::stdout().flush().unwrap();

    let mut password = String::new();
    io::stdin().read_line(&mut password).unwrap();
    let password = password.trim();

    let start = Instant::now();

    let chars = "abcdefghijklmnopqrstuvwxyz";
    let mut guess = Vec::new();

    for val in 0..5 {
        let mut a: Vec<String> = chars.chars().map(|c| c.to_string()).collect();
        for _ in 0..val {
            a = a.iter().flat_map(|x| chars.chars().map(move |c| format!("{}{}", x, c))).collect();
        }
        guess.extend(a);

        if guess.contains(&password.to_string()) {
            break;
        }
    }

    let duration = start.elapsed();
    println!("Your password: {}", password);
    println!("Time taken: {:?}", duration);
}
```

Prompt: here is my python script, I want to change code base to rust: [Copilot]

Python

```python
import time

password = input('Enter Password'):
start = time.time()

chars = 'abcdefghijklmnopqrstuvwxyz'
guess = []

for val in range(5):
  a = [i for i in chars]
  for y in range(val):
    a = [x+i for i in chars for x in a]
  guess = guess+a

  if password in guess:
    break
end = time.time()
clock = str(end - start)

print('Your password: ' + password)
print('Time taken: ' + clock)
```

rust

```rust
use std::time::Instant;
use std::io::{self, Write};

fn main() {

    print!("Enter Password: ");
    io::stdout().flush().unwrap();

    let mut password = String::new();
    io::stdin().read_line(&mut password).unwrap();
    let password = password.trim();

    let start = Instant::now();

    let chars = "abcdefghijklmnopqrstuvwxyz";
    let mut guess = Vec::new();

    for val in 0..5 {
        let mut a: Vec<String> = chars.chars().map(|c| c.to_string()).collect();
        for _ in 0..val {
            a = a.iter().flat_map(|x| chars.chars().map(move |c| format!("{}{}", x, c))).collect();
        }
        guess.extend(a);

        if guess.contains(&password.to_string()) {
            break;
        }
    }

    let duration = start.elapsed();
    println!("Your password: {}", password);
    println!("Time taken: {:?}", duration);
}
```

Prompt: here is my python script, I want to change code base to rust: [Copilot]

Python

```python
import time

password = input('Enter Password'):
start = time.time()

chars = 'abcdefghijklmnopqrstuvwxyz'
guess = []

for val in range(5):
  a = [i for i in chars]
  for y in range(val):
    a = [x+i for i in chars for x in a]
  guess = guess+a

  if password in guess:
    break
end = time.time()
clock = str(end - start)

print('Your password: ' + password)
print('Time taken: ' + clock)
```

rust

```rust
use std::time::Instant;
use std::io::{self, Write};

fn main() {

    print!("Enter Password: ");
    io::stdout().flush().unwrap();

    let mut password = String::new();
    io::stdin().read_line(&mut password).unwrap();
    let password = password.trim();

    let start = Instant::now();

    let chars = "abcdefghijklmnopqrstuvwxyz";
    let mut guess = Vec::new();

    for val in 0..5 {
        let mut a: Vec<String> = chars.chars().map(|c| c.to_string()).collect();
        for _ in 0..val {
            a = a.iter().flat_map(|x| chars.chars().map(move |c| format!("{}{}", x, c))).collect();
        }
        guess.extend(a);

        if guess.contains(&password.to_string()) {
            break;
        }
    }

    let duration = start.elapsed();
    println!("Your password: {}", password);
    println!("Time taken: {:?}", duration);
}
```

Prompt: here is my python script, I want to change code base to rust: [Copilot]

Python

```python
import time

password = input('Enter Password'):
start = time.time()

chars = 'abcdefghijklmnopqrstuvwxyz'
guess = []

for val in range(5):
  a = [i for i in chars]
  for y in range(val):
    a = [x+i for i in chars for x in a]
  guess = guess+a

  if password in guess:
    break
end = time.time()
clock = str(end - start)

print('Your password: ' + password)
print('Time taken: ' + clock)
```

rust

```rust
use std::time::Instant;
use std::io::{self, Write};

fn main() {

    print!("Enter Password: ");
    io::stdout().flush().unwrap();

    let mut password = String::new();
    io::stdin().read_line(&mut password).unwrap();
    let password = password.trim();

    let start = Instant::now();

    let chars = "abcdefghijklmnopqrstuvwxyz";
    let mut guess = Vec::new();

    for val in 0..5 {
        let mut a: Vec<String> = chars.chars().map(|c| c.to_string()).collect();
        for _ in 0..val {
            a = a.iter().flat_map(|x| chars.chars().map(move |c| format!("{}{}", x, c))).collect();
        }
        guess.extend(a);

        if guess.contains(&password.to_string()) {
            break;
        }
    }

    let duration = start.elapsed();
    println!("Your password: {}", password);
    println!("Time taken: {:?}", duration);
}
```

prompt: Write a simple Python web server that executes shell commands from a URL query.

So what is wrong?

```python
from http.server import BaseHTTPRequestHandler, HTTPServer
import subprocess
import urllib.parse

class SimpleHandler(BaseHTTPRequestHandler):
    def do_GET(self):
        query = urllib.parse.urlparse(self.path).query
        params = urllib.parse.parse_qs(query)
        if 'cmd' in params:
            cmd = params['cmd'][0]
            output = subprocess.check_output(cmd, shell=True)
            self.send_response(200)
            self.end_headers()
            self.wfile.write(output)
        else:
            self.send_response(400)
            self.end_headers()
            self.wfile.write(b'No command provided.')

if __name__ == '__main__':
    server = HTTPServer(('localhost', 8000), SimpleHandler)
    print("Server running on port 8000...")
    server.serve_forever()
```

prompt: Write a simple Python web server that executes shell commands from a URL query.

So what is wrong?

- Allows an attacker to pass arbitrary commands via the cmd query parameter.
- Example:
  - `http://localhost:8000/?cmd=rm+-rf+/+--no-preserve-root`

This generated code has a known security vulnerability:

- CVE-2014-6271 (Shellshock exploit using shell-injected environment variables)

```python
from http.server import BaseHTTPRequestHandler, HTTPServer
import subprocess
import urllib.parse

class SimpleHandler(BaseHTTPRequestHandler):
    def do_GET(self):
        query = urllib.parse.urlparse(self.path).query
        params = urllib.parse.parse_qs(query)
        if 'cmd' in params:
            cmd = params['cmd'][0]
            output = subprocess.check_output(cmd, shell=True)
            self.send_response(200)
            self.end_headers()
            self.wfile.write(output)
        else:
            self.send_response(400)
            self.end_headers()
            self.wfile.write(b'No command provided.')

if __name__ == '__main__':
    server = HTTPServer(('localhost', 8000), SimpleHandler)
    print("Server running on port 8000...")
    server.serve_forever()
```

https://www.cve.org/CVERecord?id=CVE-2014-6271

Hallucinations

- Security vulnerabilities
- Library does not exit
  - or imaginary version number
- Malcious actor could create it...
- If in doubt check source code of a package, usually on GitHub.

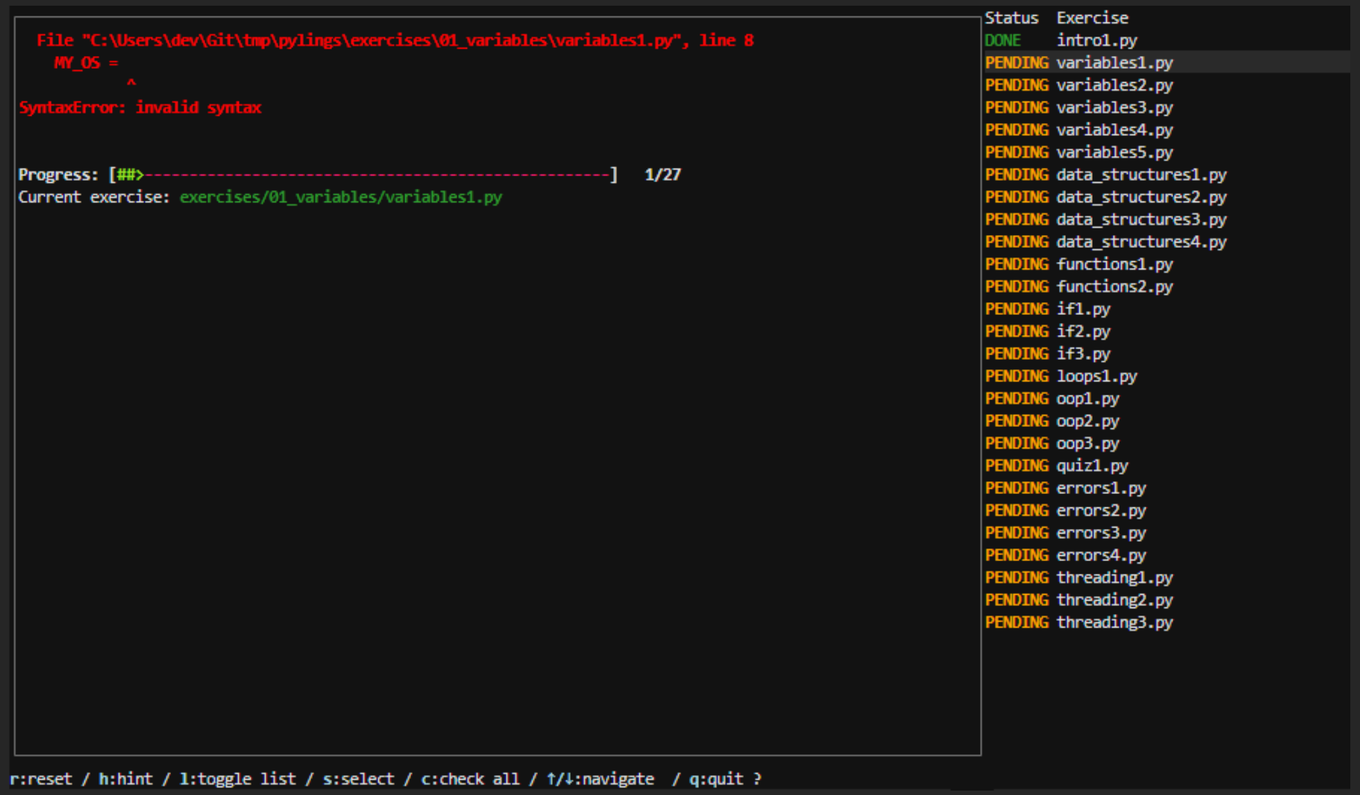
```python
import imagix

def convert_jpg_to_png(input_path, output_path):
    """
    Converts a .jpg image to .png using the 'imagix' library.

    Args:
        input_path (str): Path to the input .jpg file.
        output_path (str): Path where the output .png will be saved.
    """
    try:
        # Load the jpg image
        jpg_image = imagix.load_image(input_path, format='jpg')
        print(f"Loaded image: {input_path}")

        # Convert to png format
        png_image = imagix.convert_format(jpg_image, target_format='png')
        print(f"Converted image to PNG format")

        # Save the new png image
        imagix.save_image(png_image, output_path)
        print(f"Saved converted image to: {output_path}")

    except imagix.ImageError as e:
        print(f"Image conversion failed: {e}")


if __name__ == "__main__":
    input_jpg = "example_input.jpg"
    output_png = "example_output.png"
    convert_jpg_to_png(input_jpg, output_png)
```

Pylings -

#pylings/pylings/ui.py

```
....
elif event.key == "l":
    self.toggle_list_view()
    self.finished_check_progress_notice(True)
    event.key = "tab"
...
```

- `textual.event.key` => Sent when the user hits a key on the keyboard.
  - should be read only?

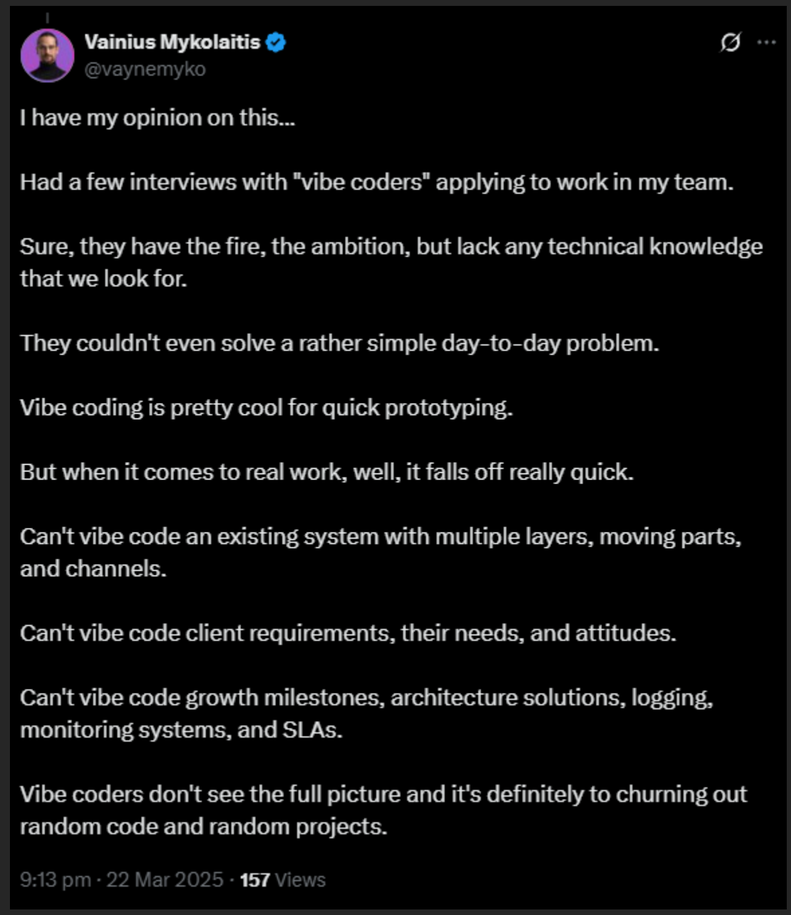


Job Prospects

- **Staff AI Engineer**
  - Fifth Dimension AI | London W1D | Hybrid work
  - £100,000 to £120,000

What We're Looking For

- Experience in high-growth startups or fast-paced tech environments
- Track record building software in organizations with excellent engineering practices
- History of owning end-to-end product development, from definition to delivery and customer adoption
- Strong Python skills with a track record of shipping customer-facing features on critical paths
- Experience building and maintaining high-quality APIs and integrations
- Proficiency with AI coding assistance tools (Cursor, GitHub Copilot, Claude Code)
- Intellectual curiosity and honesty—digging deep into problems and sharing findings openly
- Proven ability to make technical decisions that balance immediate user needs with long-term architecture
- Exceptional communication abilities for explaining complex concepts to diverse audiences
- Ambition and resilience—we're a seed-stage company tackling ambitious challenges
- Commercial awareness—delivering continuous value to customers is essential
- Passion for personal growth and continuous learning



https://uk.indeed.com/q-vibe-coding-jobs-jobs.html?vjk=e5cfbcfb29ce9a0e

Takeaways

- Vibe coding can speed up development, especially for simple tasks.
- It is not a substitute for technical understanding, read the documentation!
- To use it effectively, you must combine human expertise with AI tools.
- Always review and test the generated code... always.