Emoji Support Test

Headings with Emoji

Mixed text and emoji

Hello world! This paragraph mixes regular text with emoji characters.

Here is a **bold section with emoji** and *italic with stars*.

- First item with emoji
- Second item with emoji
- Plain item with trailing emoji

1. Gold
2. Silver
3. Bronze

Feature	Status
Emoji in headings	✅
Emoji in paragraphs	✅
Emoji in tables	✅
Emoji in lists	✅

> *A blockquote with emoji: Wise words here.*

Some flags:

ZWJ sequences:

Skin tones:

Code Blocks

`const x = 42;` and `console.log("hello")`.

TypeScript

```
import { readFile } from 'fs/promises';

interface Config {
  name: string;
  port: number;
  debug: boolean;
}

async function loadConfig(path: string): Promise<Config> {
  const raw = await readFile(path, 'utf-8');
  const config: Config = JSON.parse(raw);
  // Validate required fields
  if (!config.name || config.port <= 0) {
    throw new Error(`Invalid config: ${config.name}`);
  }
  return config;
}

const DEFAULT_PORT = 3000;
export { loadConfig, DEFAULT_PORT };
```

JavaScript

```javascript
class EventEmitter {
  constructor() {
    this.listeners = new Map();
  }

  on(event, callback) {
    if (!this.listeners.has(event)) {
      this.listeners.set(event, []);
    }
    this.listeners.get(event).push(callback);
    return this;
  }

  emit(event, ...args) {
    const handlers = this.listeners.get(event) || [];
    for (const handler of handlers) {
      handler(...args);
    }
    return handlers.length > 0;
  }
}

// Usage
const emitter = new EventEmitter();
emitter.on('data', (msg) => console.log(`Received: ${msg}`));
emitter.emit('data', 'Hello World!');
```

Python

```python
from dataclasses import dataclass
from typing import Optional

@dataclass
class User:
    name: str
    email: str
    age: Optional[int] = None

    def greet(self) -> str:
        """Return a greeting message."""
        return f"Hello, {self.name}!"

    @property
    def is_adult(self) -> bool:
        return self.age is not None and self.age >= 18

# Create and use
users = [User("Alice", "alice@example.com", 30), User("Bob", "bob@example.com")]
active = [u for u in users if u.is_adult]
print(f"Found {len(active)} adult users")
```