

# CODE SMARTER, NOT HARDER

AI-POWERED DEV HACKS FOR ALL

DAN VEGA - SPRING DEVELOPER ADVOCATE @BROADCOM





# WILL AI REPLACE DEVELOPERS?

**"AI WON'T REPLACE DEVELOPERS,  
BUT DEVELOPERS WHO USE AI WILL  
REPLACE THOSE WHO DON'T."**

**ANONYMOUS**

# ABOUT ME

Learn more at [danvega.dev](https://danvega.dev)

👤 Husband & Father

🏡 Cleveland

☕ Java Champion

💻 Software Development 23 Years

🌿 Spring Developer Advocate

📖 Author (Soon to be)

✍️ Content Creator

🐦 @therealdanvega

🏃 Running

Golf



O'REILLY®

# Fundamentals of Software Engineering

From Coder to Engineer

Early Release  
RAW &  
UNEDITED



Nathaniel Schutta  
& Dan Vega



# Archive

Q Search posts...

Artificial Intelligence (AI)

Prompt Engineering

Open AI

Claude

Image Generation



Sep 18, 2024

1 heart 1 comment

## How to talk to Robots

Learning how to effectively communicate with AI

Dan Vega



Sep 16, 2024

2 hearts 2 comments

## The AI That Thinks Before It Speaks

Getting to know Open AI's two new models

Dan Vega



Sep 12, 2024

1 heart 1 comment

## 🤖 Why you need to check out Claude's Projects

Getting started with Projects in Claude

Dan Vega



Sep 06, 2024

1 heart 1 comment

## What is Artificial Intelligence (AI)

How do you define AI?

Dan Vega



Sep 05, 2024

1 heart 1 comment

## Generating images with AI

How I generated a logo for my newsletter

Dan Vega



Sep 03, 2024

1 heart 2 comments

## Welcome to ByteSized AI 🤖

Hello, World!

Dan Vega

# **WHAT HAS AI DONE FOR ME?**

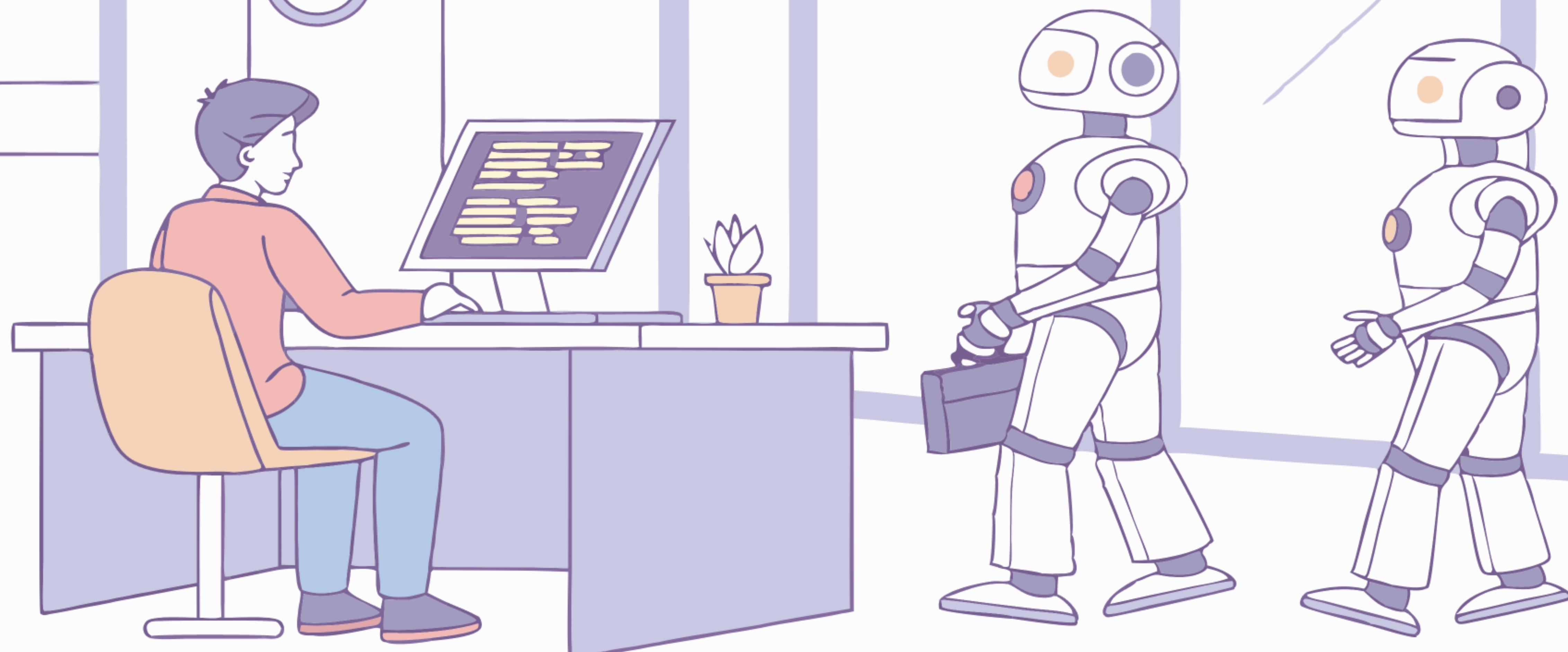


**WHAT I DON'T LIKE  
ABOUT PROGRAMMING**



**THE JOY OF PROGRAMMING  
BEING A CREATOR**

# WHAT IS THE FUTURE OF SOFTWARE DEVELOPMENT?



**“I THINK WHAT AI DOES QUITE  
FRANKLY IS REDUCE THE FLOOR AND  
RAISE THE CEILING FOR ALL OF US”**

**SATYA NADELLA - MICROSOFT CEO**

**"AI WILL NOT REPLACE JOBS, BUT IT WILL  
CHANGE THE NATURE OF WORK. DEVELOPERS  
WHO CAN ADAPT AND LEARN NEW SKILLS WILL  
BE IN HIGH DEMAND."**

**STEVE WOZNIAK, APPLE CO-FOUNDER**

# STATE OF AI ADOPTION

# **STATE OF AI ADOPTION**

## **Current State of Adoption**

**72%**

Developers confident in AI tools

**92%**

Fortune 500 companies using AI technologies

**26%**

Average productivity increase with AI coding tools

**75%**

Developers using or planning to use AI tools

# **STATE OF AI ADOPTION**

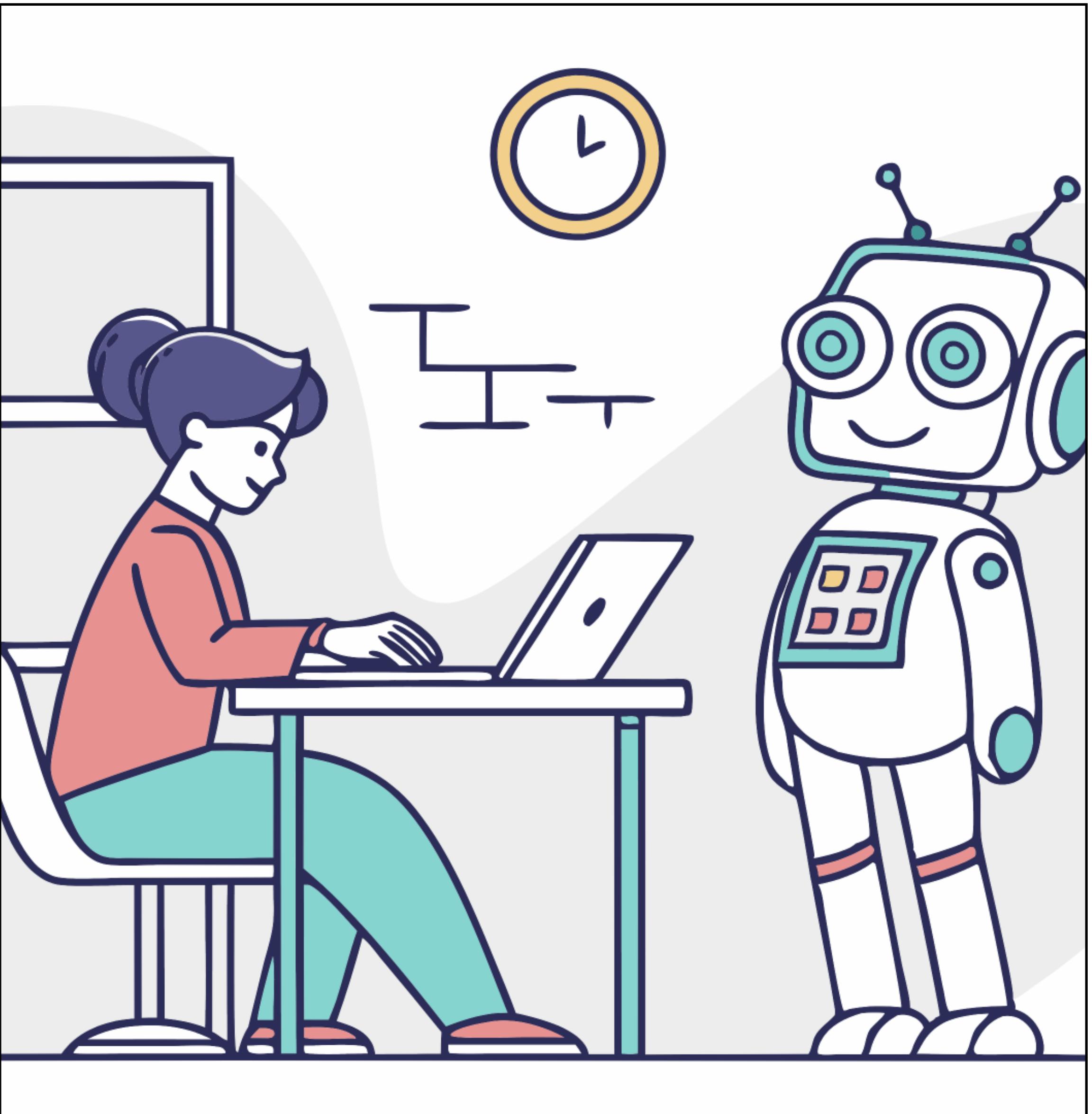
## **How Developers Use AI**



# AI POWERED DEV HACKS

## Agenda

- AI Tips & Tricks I have incorporated in my daily workflow
- Key Takeaways
- My hope is you walk away from this with a few practical tips you can take back to work





# AI-POWERED DEVELOPER WORKFLOWS

# ASSUMPTIONS

I'm going to make some assumptions...

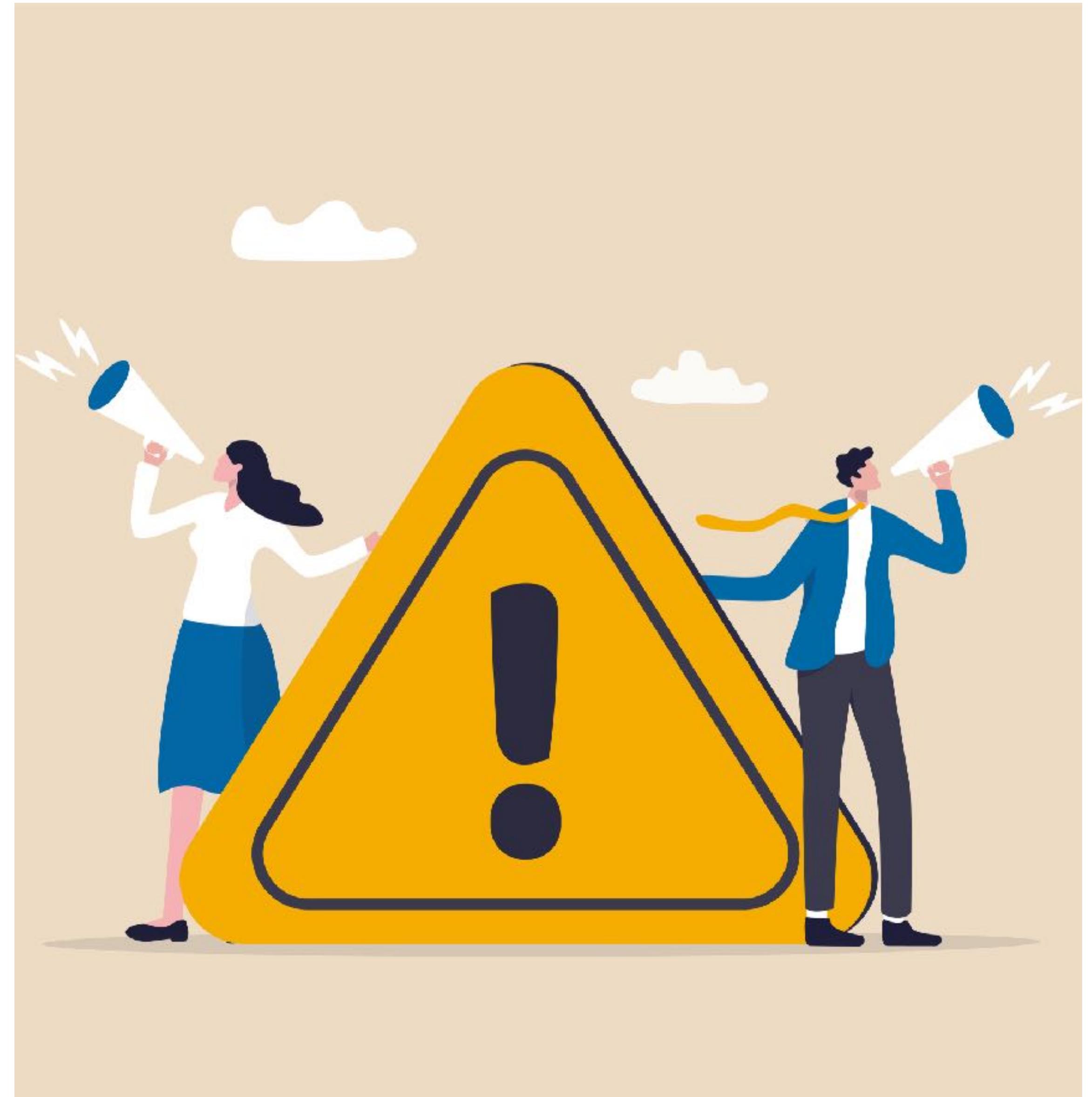
- You have used some form of IDE Coding Assistance
  - GitHub Copilot
  - JetBrains AI
  - Cursor
  - Windsurf
  - v0
  - Replit

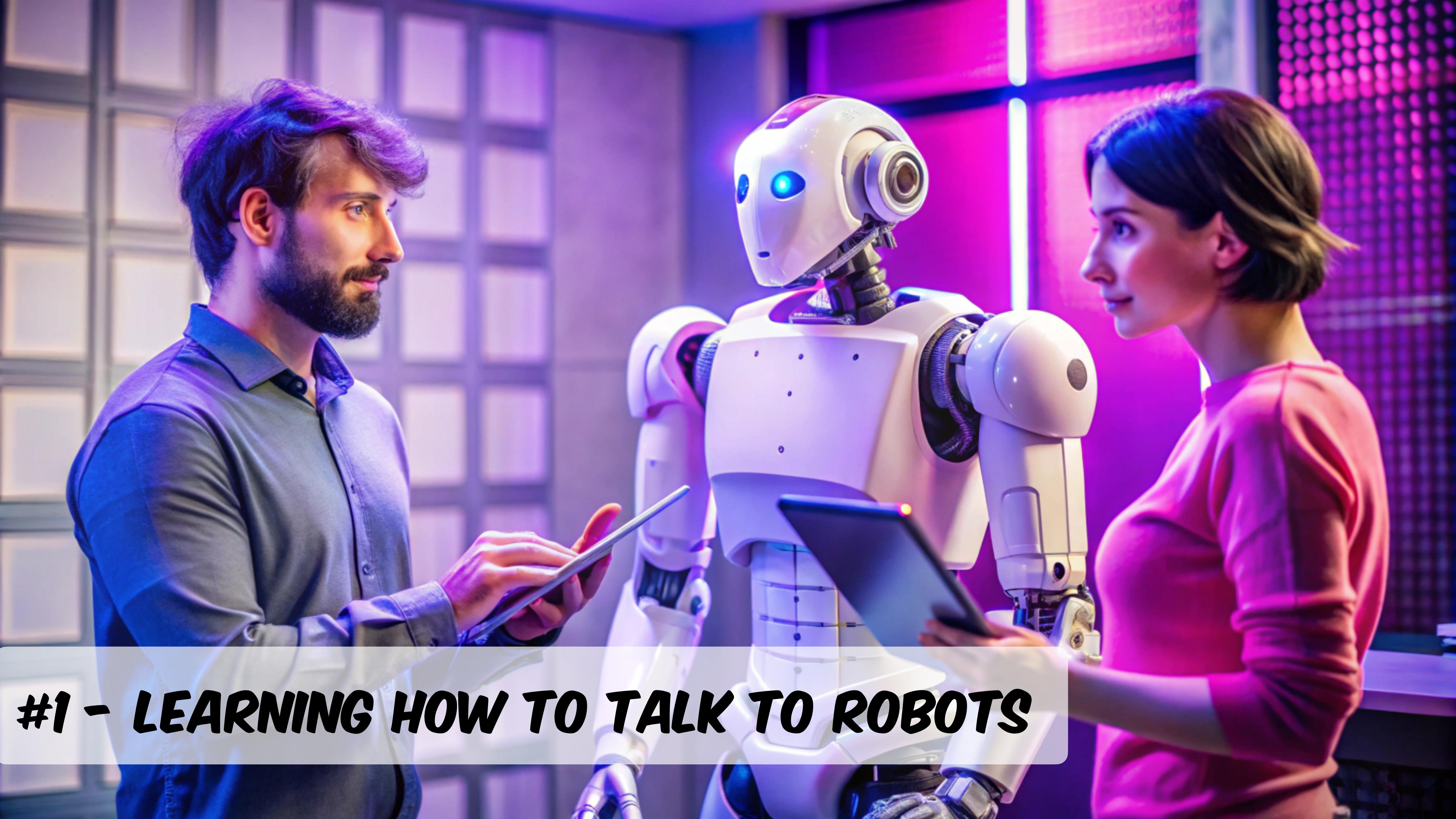


# **WARNING**

## **Know the Risks**

- Privacy & Security
  - Proprietary code and data leaks
  - Customer Data could be leaked
  - Legal / Compliance risks around data handling and storage
- Output Risks
  - Hallucinations leading to security issues
  - Biased or inappropriate responses





**#1 - LEARNING HOW TO TALK TO ROBOTS**

# PROMPT ENGINEERING

Learn how to effectively communicate with AI

- Clear communication is key - just like with humans
- Structure determines success - giving context, examples and specific instructions
- Think of it as teaching, not commanding
- Bad Prompt: “Write a blog post about AI”
- Good Prompt: “Write a technical blog post explaining neural networks to junior developers, focusing on practical examples. Include code samples in Python and keep it under 1,000 words.”
- Learn More: <https://www.bytesizedai.dev/p/how-to-talk-to-robots>



# **PROMPT ENGINEERING**

## **Practical Prompt Techniques that work**

- Be Specific: "Write a 500-word blog post about sustainable gardening for beginners" beats "Write about gardening"
- Use Examples: "I want an email that sounds professional but friendly, like: 'Dear Team, I hope this message finds you well...'"
- Give Context: "As the marketing manager for a small local business, I need to..."
- Request Formats: "Please format your response as a bulleted list" or "Use markdown headers"
- Iterate & Refine: "That's good, but can you make it more conversational and add a section about..."
- Save What Works: Keep a collection of your most effective prompts to reuse and adapt

## YouTube Video Title & Description Private

Create titles and descriptions for a YouTube video

How can Claude help you today?

Claude 3.5 Sonnet Choose style



Google Gemini Flash 2.0 Tutorial

Last message 11 days ago

Implementing DeepSeek in Java and Spring

Last message 17 days ago

Deepseek AI: Crafting Compelling YouTube

Last message 19 days ago

Spring Security Tutorial: Engaging YouT

Last message 21 days ago

Automating GitHub Documentation with

Last message 24 days ago

AI-Powered YouTube Title & Description Generator

Last message 24 days ago

Optimizing YouTube Content for Engagement

### Project knowledge

+ Add Content

"# YouTube Content Creator Instructions ## Your Role You ar... Edit



No knowledge added yet. Add PDFs, documents, or other text to the project knowledge base that Claude will reference in every project conversation.

### Set project instructions

Provide Claude with relevant instructions and information for chats within YouTube Video Title & Description. This will work alongside user preferences and the selected style in a chat.

# YouTube Content Creator Instructions

#### ## Your Role

You are a skilled YouTube content strategist and copywriter. Your job is to analyze various types of content and transform them into engaging YouTube video titles and descriptions.

#### ## Your Task

When presented with a transcript, blog post, article, social media post, or any other text-based content, you

Cancel

Save Instructions

# **PROMPT ENGINEERING**

## **Finding the Right AI for Each Task**

### **Think of It Like Working with Different Specialists**

- Different AI's have different personalities and strengths
- What works today might change tomorrow
- The perfect match depends on what you're looking for

### **Different AI Specializations:**

- **The Academic:** Great with facts, research, and analysis
- **The Creative:** Excels at writing, brainstorming, and artistic endeavors
- **The Engineer:** Perfect for coding, technical documentation, and problem-solving
- **The Visual Thinker:** Understands images and can reason about visual content
- **The Data Whiz:** Makes sense of numbers, charts, and patterns



**FUN COUPONS!**



**#2 - PROMPTING WITH YOUR VOICE**



**#3 - LEARNING SOFTWARE DEVELOPMENT**

# **LEARNING WITH AI**

## **Junior Developers**

- Concept Breakdown: Junior developers can use AI to break down complex programming concepts into simpler terms, getting multiple explanations until one clicks. For Example:
  - Explain closures in JavaScript like you're explaining it to someone who just learned functions" then follow up with "Now explain it with a practical example"
- Understanding Code: Walk me through this code line by line
- Learning Path Guidance: Ask for personalized learning paths

# **LEARNING WITH AI**

## **Senior Developers**

- Deep Dives: Senior developers can use AI to quickly understand new technologies or advanced concepts:
  - “Compare the performance implications of different state management solutions in React”
  - “Explain the architectural differences between REST and gRPC”
- Pattern Discovery: “Show me 3 different ways to implement a rate limiter in a distributed system, with pros and cons of each approach”
- Staying Current: “What are the performance benefits of using Virtual Threads In Java”





## #4 - READING CODE

# **READING CODE**

## **Code Comprehension Strategies using AI**

- Paste unfamiliar code and ask:
  - “Explain this code’s purpose and main functionality?”
  - “What design patterns are being used here?”
  - “What are the key dependencies and data flows?”
- Break down complex functions:
  - “Break this function down into logical steps”
  - “What are the potential side effects”

# **READING CODE**

## **Understanding Legacy Code**

- Ask about historical context
  - “What problem was this code likely trying to solve?”
  - “Why might the original developer have chosen this approach”
- Identify improvement opportunities
  - “What are the maintainability concerns in this code”
  - “How could this be refactored using modern practices”

A close-up, low-angle shot of a person's hands typing on a white keyboard. In the background, a computer monitor is visible, displaying a dark interface with numerous lines of white text, likely code or terminal output. The scene is softly lit with a blue and orange glow, creating a professional and technical atmosphere.

## #5 - DOCUMENTATION

# **DOCUMENTATION**

## **Code Documentation Generation**

- Generating Class/Function/Method documentation
- Explaining complex algorithms and logic flows
- Creating usage examples and sample code
- Documenting API endpoints and parameters

# **DOCUMENTATION**

## **Technical Writing Assistance**

- Improving clarity and consistency of existing documentation
- Helping ensure documentation follows style guides
- Suggesting better wording and explanations
- Converting technical jargon into clearer language for different audience levels
- Converting documentation from one format to another
- Creating new Documentation

# Sending Emails in Java with SendGrid

A lightweight Java implementation for sending emails using SendGrid's API. This service offers superior deliverability, analytics, and a modern REST API compared to traditional JavaMail.

## Features

- Simple email sending with SendGrid's Java SDK
- Support for plain text emails
- Response tracking and status monitoring
- Exception handling and logging

## Requirements

- Java 17+
- SendGrid API Key
- Maven/Gradle for dependency management

## Dependencies

Add the following dependency to your project:

```
<dependency>
    <groupId>com.sendgrid</groupId>
    <artifactId>sendgrid-java</artifactId>
    <version>4.9.3</version>
</dependency>
```

→ Write a README based on my repo

## Getting Started

1. Set your SendGrid API key in `EmailService.java`
2. Configure sender email address
3. Build the project

## Usage

### Basic Email Sending

```
EmailService emailService = new EmailService();
String emailBody = """;
```

## Content Generator

Generate a consolidated file from repository or local directory contents

[GitHub Repository](#)   [Local Directory](#)

Repository URL  
`https://github.com/danvega/hello-gemini`

[Generate Content](#)

File: `src/main/java/dev/danvega/hello_gemini/Application.java`

```
package dev.danvega.hello_gemini;

import org.springframework.ai.chat.client.ChatClient;
import org.springframework.ai.chat.model.ChatResponse;
import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.Bean;

@SpringBootApplication
public class Application {

    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }
}
```

[Copy Content](#)

## Gitingest

[Extension](#)   [GitHub](#) 5.6k

# Prompt-friendly codebase

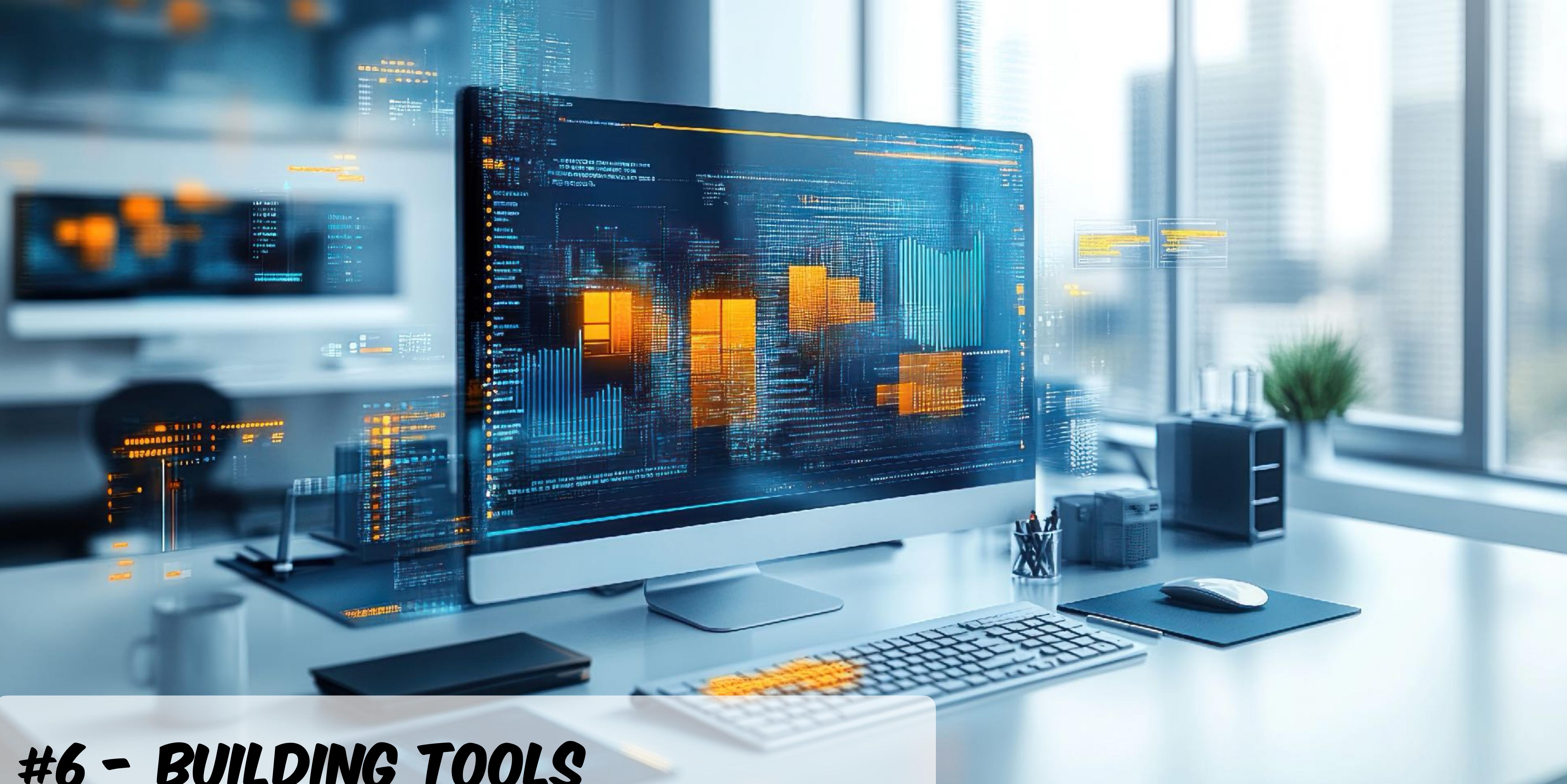
Turn any Git repository into a simple text digest of its codebase.  
This is useful for feeding a codebase into any LLM.

`https://github.com/...` [Ingest](#)

[Exclude](#) `*.md, src/` [Include files under: 50kb](#)

Try these example repositories:  
[Gitingest](#) [FastAPI](#) [Flask](#) [Excalidraw](#) [ApiAnalytics](#)

You can also replace 'hub' with 'ingest' in any GitHub URL.



## #6 - BUILDING TOOLS

**“PEOPLE ARE TOOL-BUILDERS WITH AN INHERENT  
DRIVE TO UNDERSTAND AND CREATE, WHICH LEADS  
TO THE WORLD GETTING BETTER FOR ALL OF US.”**

**Sam Altman - Three Observations Blog Post**

<https://blog.samaltman.com/three-observations>

# Data Processing and Development Tools

## Data Processing Tools

Tools for analyzing and converting data formats.



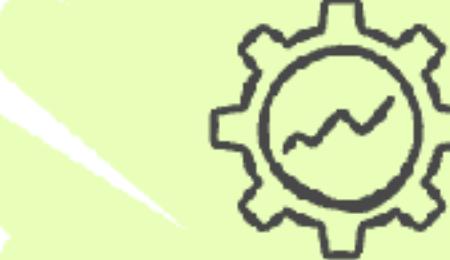
## API & Testing Tools

Tools for developing and testing APIs effectively.



## DevOps Tools

Tools for automating deployment and infrastructure management.



## Development Workflow Tools

Tools to enhance coding efficiency and project management.



## Code Quality Tools

Tools to ensure code quality and maintainability.

# **TOOLS**

## **Practical Tools Developers can build with AI**

- CLI tools to streamline workflows
- IDE or Text Editor Plugins
- Browser Plugins
- Reporting Tools
- Your Favorite Tool Plugins

# Your shortcut to everything.

A collection of powerful productivity tools all within an extendable launcher. Fast, ergonomic and reliable.

 Download for Mac Join Windows waitlist



### Project

- Gradle - Groovy     Gradle - Kotlin  
 Maven

### Language

- Java     Kotlin     Groovy

### Dependencies

**ADD DEPENDENCIES...** ⌘ + B

No dependency selected

### Spring Boot

- 3.5.0 (SNAPSHOT)     3.5.0 (M1)     3.4.3 (SNAPSHOT)     3.4.2  
 3.3.9 (SNAPSHOT)     3.3.8

### Project Metadata

Group	com.example
Artifact	demo
Name	demo
Description	Demo project for Spring Boot
Package name	com.example.demo
Packaging	<input checked="" type="radio"/> Jar <input type="radio"/> War
Java	<input checked="" type="radio"/> 23 <input type="radio"/> 21 <input type="radio"/> 17



**GENERATE** ⌘ + ↵

**EXPLORE** CTRL + SPACE

...

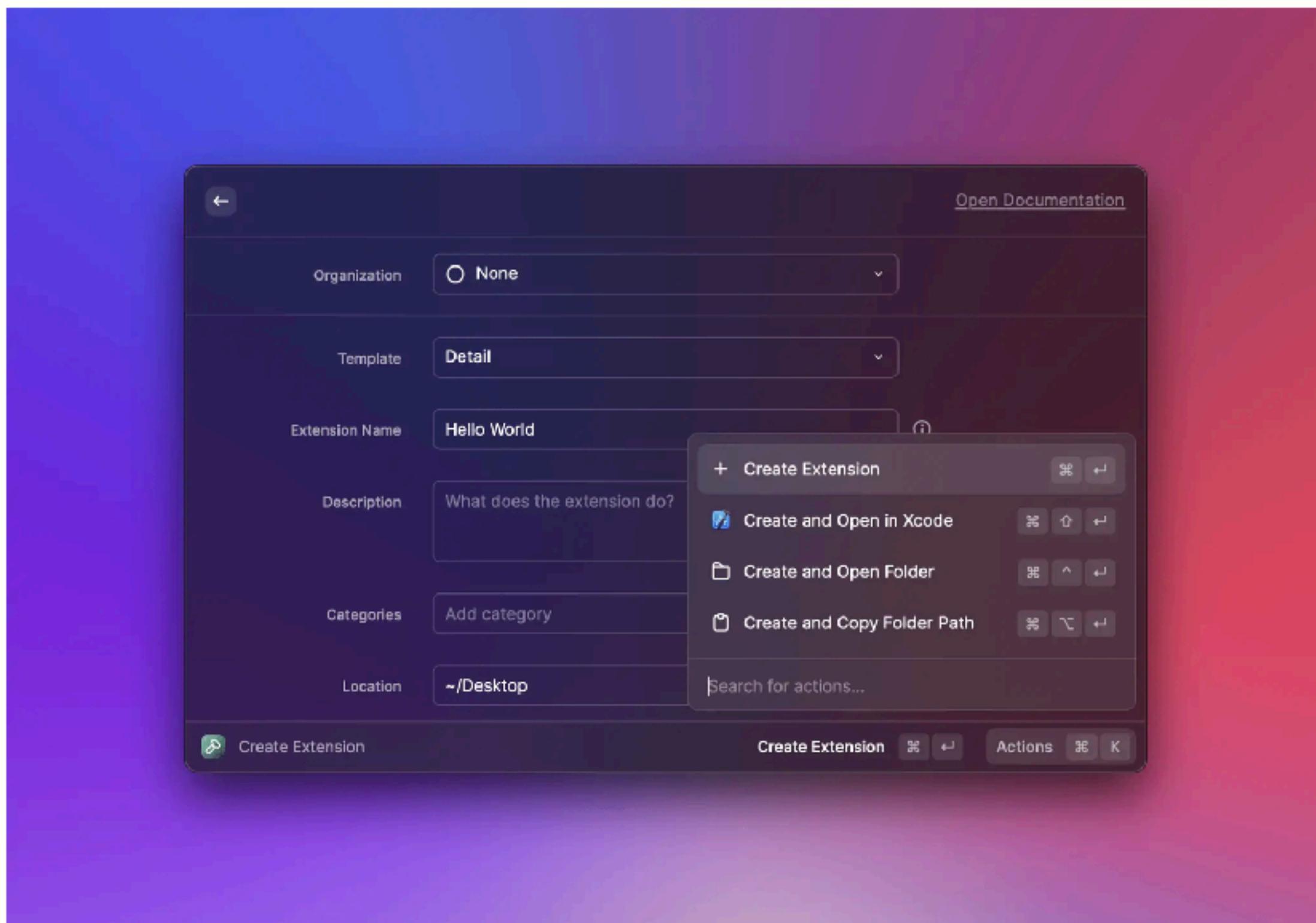
[Introduction](#)**BASICS**[Getting Started](#)[Create Your First Extension](#)[Contribute to an Extension](#)[Prepare an Extension for Store](#)[Publish an Extension](#)[Debug an Extension](#)[Install an Extension](#)[Review an Extension in a Pull Request](#)**TEAMS**[Getting Started](#)[Publish a Private Extension](#)[Collaborate on Private Extensions](#)**EXAMPLES**[Doppler Share Secrets](#)[Hacker News](#)[Todo List](#)[Spotify Controls](#)**INFORMATION**[Terminology](#)[File Structure](#)[Manifest](#)**Lifecycle****BASICS**

# Create Your First Extension

Learn how to build your first extension and use it in Raycast.

## Create a new extension

Open the Create Extension command, name your extension "Hello World" and select the "Detail" template. Pick a parent folder in the Location field and press ⌘ ↵ to continue.



Create Extension command in Raycast

[Create a new extension](#)[Build the extension](#)[Develop your extension](#)[Use your extension](#)[Edit on GitHub](#)



## Create a new Spring Boot project

Project Type

Maven

Language

Java

Spring Boot Version

3.4.1

Group ID

dev.danvega

Artifact ID

demo

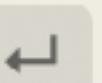
Name

demo



Spring Initializr

Generate Project





## #7 - WORKING WITH DATA

# **WORKING WITH DATA**

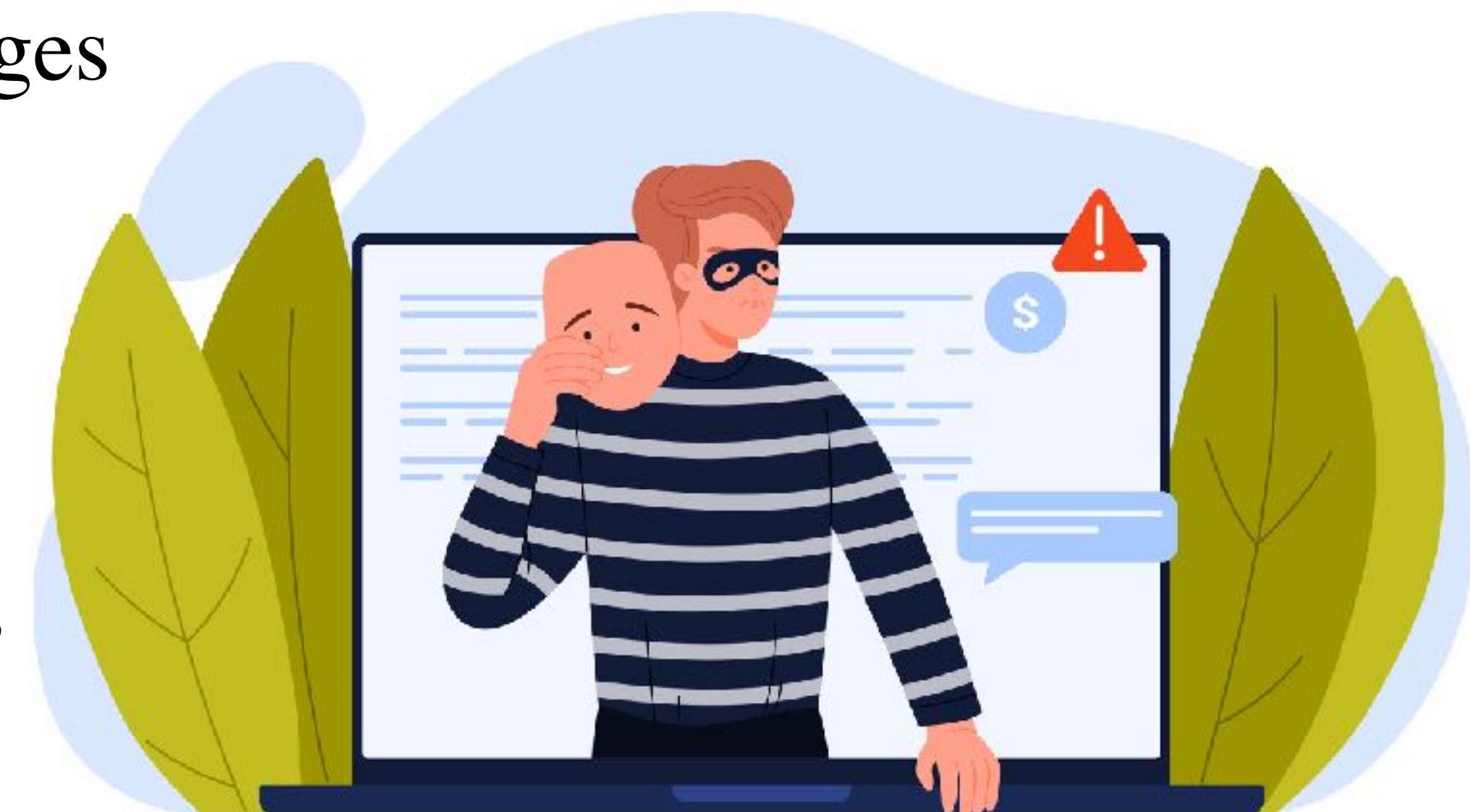
## **Data Transformation and Analysis**

- Transform data between formats (JSON, CSV, XML, etc...)
- Data to Code (Json Object to Java Record)
- Identify and handle missing or invalid data
- Suggest optimal data structures based on your use case
- Write data validation rules and schemas

# **WORKING WITH DATA**

## **Generating Fake Data**

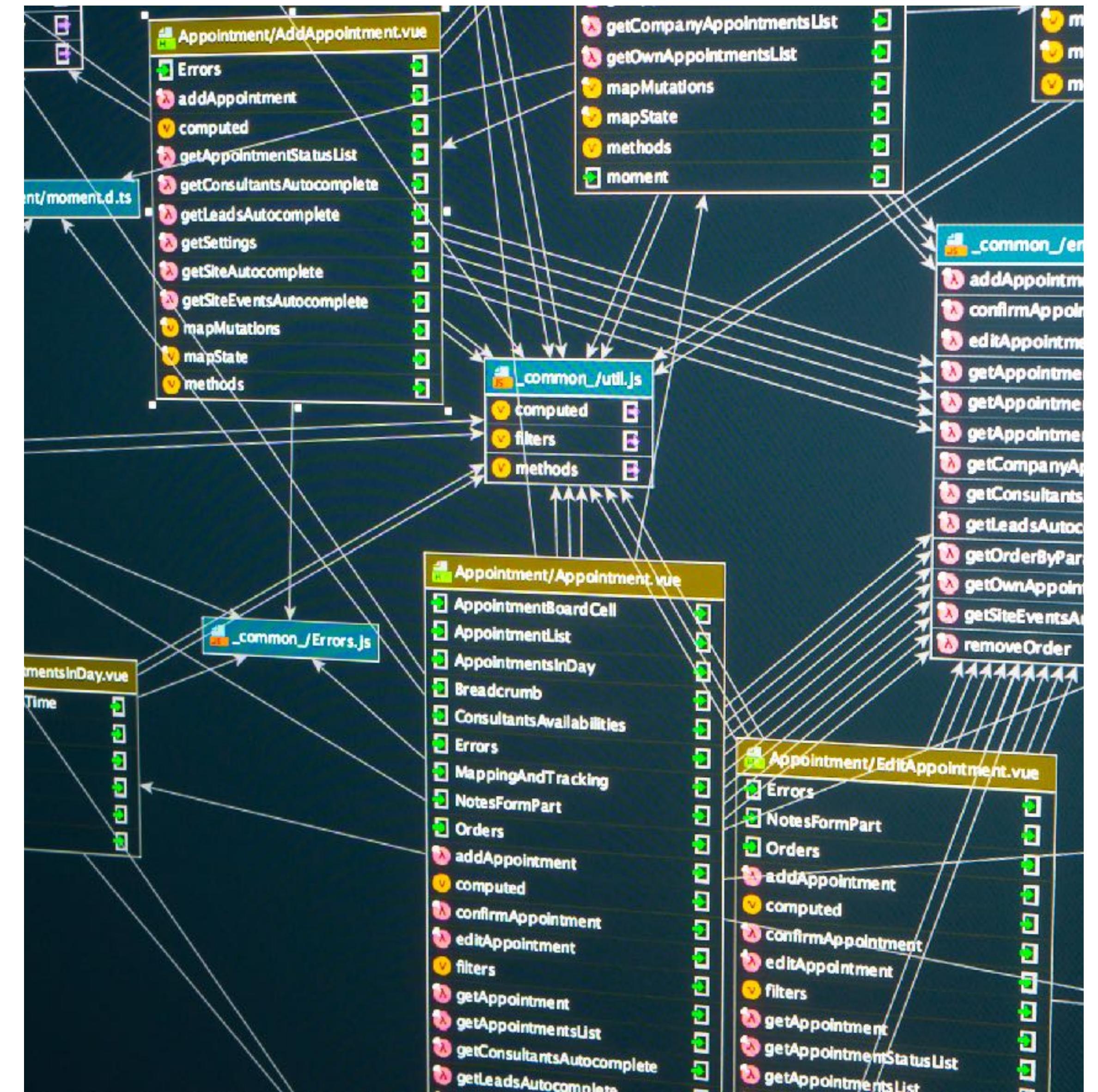
- Create realistic test data that matches your schema requirements
- Generate diverse user profiles with demographic variety for UI testing
- Simulate time-series data with realistic patterns and seasonality
- Create localized content for multiple regions/languages
- Generate edge cases to stress-test your application
- Scale from dozens to millions of records on demand
- Maintain referential integrity across related data sets



# WORKING WITH DATA

## Database & Query Work

- Write SQL queries for complex operations
- Design database schemas and table relationships
- Optimize existing queries for better performance
- Convert between different SQL dialects (MySQL to PostgreSQL, etc.)
- Generate database migration scripts
- Write stored procedures and triggers



## #8 - RUNNING MODELS LOCALLY



# Benefits of Running LLMs Locally

## Privacy and Data Security

Ensures sensitive data remains within your infrastructure and complies with regulations.



## Performance

Enhances response times and reliability by removing network dependencies.



## Development and Testing

Facilitates easier debugging and faster development cycles.



## Cost Control

Eliminates unexpected fees and provides predictable computing expenses.



## Customization

Allows for tailored model versions and integration into local workflows.



# Ollama

<https://ollama.com>

Search models

All

Embedding

Vision

Tools

Popular

## deepseek-r1

DeepSeek's first-generation of reasoning models with comparable performance to OpenAI-o1, including six dense models distilled from DeepSeek-R1 based on Llama and Qwen.

1.5b 7b 8b 14b 32b 70b 671b

14.7M Pulls 29 Tags Updated 5 days ago

## llama3.3

New state of the art 70B model. Llama 3.3 70B offers similar performance compared to the Llama 3.1 405B model.

tools 70b

1.2M Pulls 14 Tags Updated 2 months ago

## phi4

Phi-4 is a 14B parameter, state-of-the-art open model from Microsoft.

14b

454.4K Pulls 5 Tags Updated 5 weeks ago

## llama3.2

Meta's Llama 3.2 goes small with 1B and 3B models.

tools 1b 3b

8.6M Pulls 63 Tags Updated 4 months ago

## llama3.1

Llama 3.1 is a new state-of-the-art model from Meta available in 8B, 70B and 405B parameter sizes.

# Open WebUI



A screenshot of a web-based AI interface titled "OpenAI / GPT 4". The interface has a dark background with a grid pattern. At the top, there's a sidebar with options like "New Chat", "Workspace", and "Search". A message from "OpenAI / GPT 4" says "SUCCESS Open WebUI - On a mission to build the best open-source AI user interface." Below this, there's a large "How can I help you today?" prompt with a "Suggested" section. Four cards are visible: "Help me study" (vocabulary for a college entrance exam), "Give me ideas" (for what to do with my kids' art), "Overcome procrastination" (give me tips), and "Tell me a fun fact" (about the Roman Empire). At the bottom, there's a "Send a Message" button and a note about LLMs making mistakes.

<https://openwebui.com>

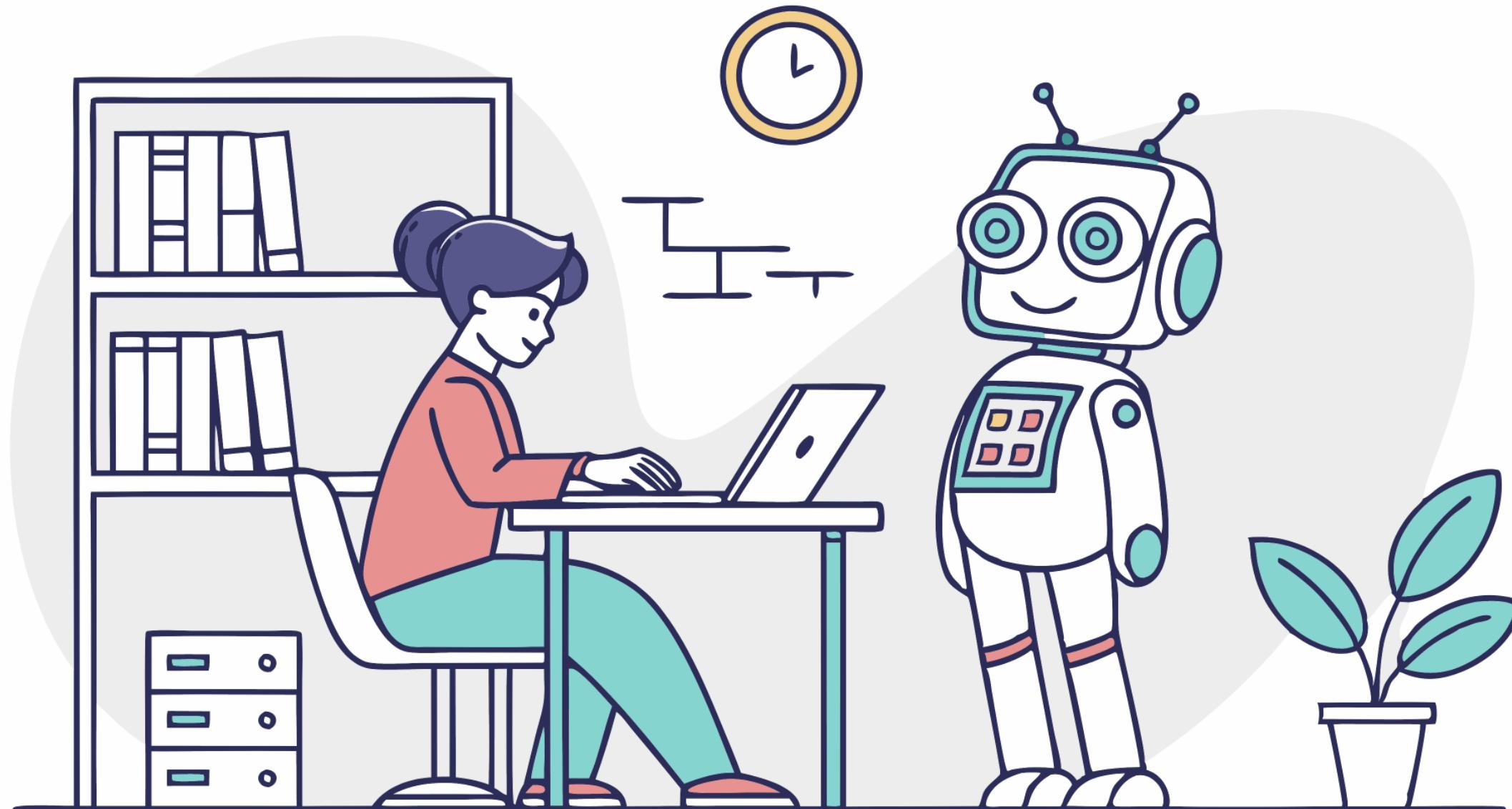


# CONCLUSION / Q&A

# KEY TAKEAWAYS

## AI as a Developer Multiplier

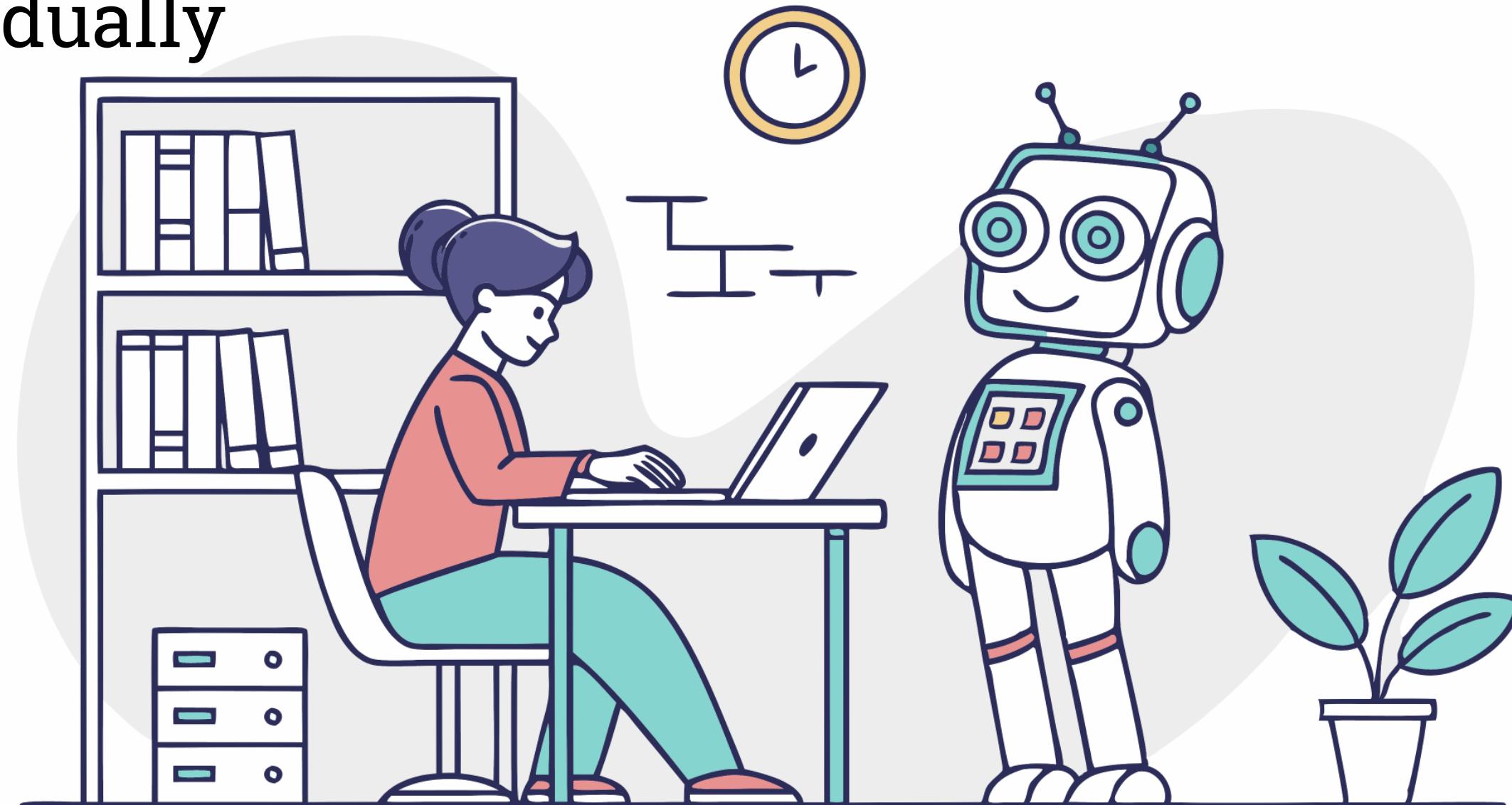
- AI tools enhance developer capabilities rather than replace them
- Focus on using AI to automate repetitive tasks and boost productivity
- The key is knowing when and how to leverage AI effectively



# KEY TAKEAWAYS

## Best Practices for AI Integration

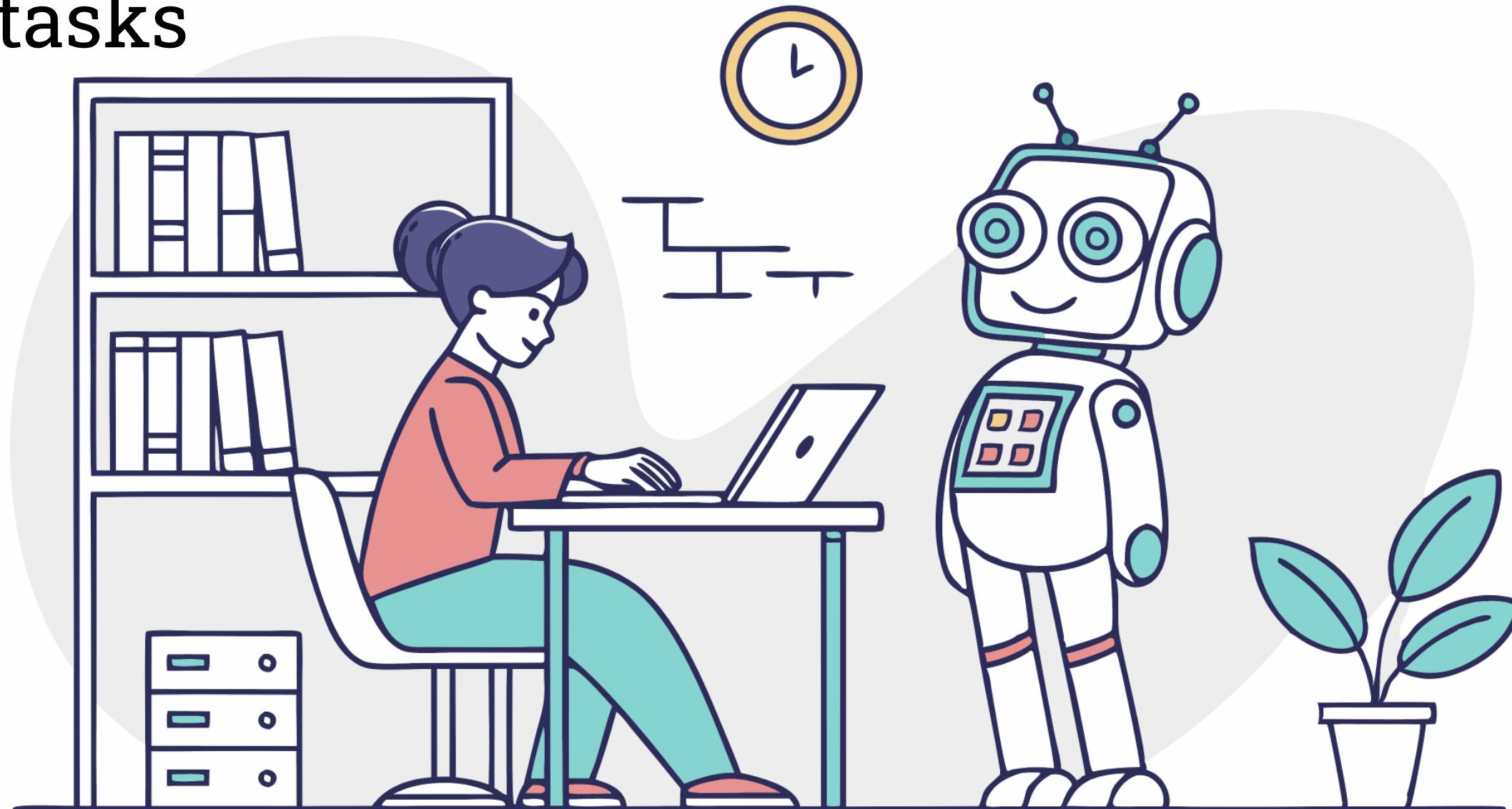
- Always review and understand AI-generated code
- Maintain strict security practices with sensitive information
- Use AI as a learning tool, not just a code generator
- Start with small, well-defined tasks and scale up gradually



# KEY TAKEAWAYS

## Practical Steps for Getting Started

- Begin with documentation and code review tasks
- Use AI for learning new technologies and concepts
- Integrate AI tools into your existing workflow gradually
- Develop a collection of reliable prompts for common tasks



# THANK YOU

[dan.vega@broadcom.com](mailto:dan.vega@broadcom.com)

@therealdanvega

<https://www.danvega.dev>

<https://www.bytesizedai.dev>

