



MyAIHelper

Your Personalised AI Setup & Training Guide

Complete step-by-step guide to achieve: I'd like to learn to code

Your Info

AI Experience

New to AI

User Type

Personal & Hobby

Specific Goal

I'd like to learn to code

Understanding your goal

Learning to code opens doors to countless creative and professional opportunities, from building your own apps to automating daily tasks. AI-powered coding tools can dramatically accelerate your learning journey by providing instant explanations, debugging help, and personalised guidance tailored to your pace and learning style. With the right AI assistants, you'll overcome common beginner frustrations and build confidence through hands-on practice with immediate feedback.

1. Set Up GitHub Copilot

GitHub Copilot acts like a coding mentor that suggests code completions and explanations in real-time, making it perfect for beginners who need guidance whilst writing their first programs.

Time: 25-35 minutes **Difficulty:** New to AI [Go to GitHub Copilot →](#)

Your Step-By-Step Set-up Instructions

- 1 Create a free GitHub account at github.com using your email address and choose a username that represents you professionally
- 2 Navigate to GitHub Copilot settings and start your free 30-day trial (students get it free with GitHub Student Pack)
- 3 Download and install Visual Studio Code from code.visualstudio.com - this will be your main coding environment
- 4 Open VS Code and click the Extensions icon (four squares) on the left sidebar, then search for 'GitHub Copilot' and install it
- 5 Install the 'GitHub Copilot Chat' extension as well - this allows you to ask coding questions directly in your editor
- 6 Sign in to GitHub when prompted by clicking 'Sign in to GitHub' and authorising VS Code to access your account
- 7 Create a new folder on your desktop called 'My Coding Journey' and open it in VS Code using File > Open Folder

- 8 Create your first file called 'hello.py' and type '# My first Python program' - watch Copilot suggest code completions
- 9 Test Copilot by typing 'print(' and see how it suggests completing your first print statement
- 10 Open Copilot Chat by pressing Ctrl+Shift+I (Cmd+Shift+I on Mac) and ask it to explain any code you don't understand
- 11 Configure Copilot settings by going to File > Preferences > Settings and searching for 'copilot' to adjust suggestion frequency
- 12 Practice accepting suggestions with Tab key and rejecting with Escape - this will become second nature as you code

Your Custom Prompts

For: Daily coding practice with explanations

```
I'm a complete beginner learning to code. Can you give me a simple coding exercise in Python that teaches [specific concept like variables, loops, functions] and then explain each line of code in plain English? Please start with something I can complete in 10-15 minutes and include the expected output.
```

For: Code debugging and learning from mistakes

```
I'm getting this error in my Python code: [paste your error message]. I'm a beginner, so please explain what went wrong in simple terms, show me the corrected code, and teach me how to avoid this mistake in future. Also explain any programming concepts I should understand to prevent similar errors.
```

For: Weekly progress review and next steps

```
I've been learning to code for [time period] and have practiced [list topics you've covered]. As a beginner, what should I focus on learning next? Please suggest 3 specific coding projects that would help me practice what I've learned and introduce one new concept. Include difficulty estimates and learning objectives for each project.
```

Setup Tips

Start with Python as your first language since Copilot works exceptionally well with it and it's beginner-friendly. Don't rely on Copilot suggestions blindly - always read and try to understand what it's suggesting before accepting. Create a 'practice' folder where you can experiment freely without worry. Enable Copilot's inline suggestions but keep the chat panel open for questions - this combination gives you both automated help and explanatory support when you're stuck.

Integrations

Use GitHub Copilot alongside the Python extension in VS Code for syntax highlighting and error detection. Connect it with the 'Code Runner' extension to easily run your code with a single click. Later, integrate with GitHub Desktop to save your coding projects and track your progress over time - this creates a portfolio of your learning journey.

Troubleshooting

If Copilot isn't showing suggestions, check your internet connection and ensure you're signed into GitHub. If suggestions seem too advanced, add comments like '# beginner level' or '# simple version' to get more appropriate suggestions. If you're overwhelmed by suggestions, disable inline suggestions temporarily in settings and rely on the chat feature until you're more comfortable. Remember that Copilot learns from your coding style, so the more you practice, the better its suggestions become.

2. Set Up ChatGPT for Coding Education

ChatGPT serves as your personal coding tutor, explaining programming concepts in plain English and helping you understand the 'why' behind coding practices, not just the 'how'.

Time: 15-25 minutes **Difficulty:** New to AI [Go to ChatGPT →](#)

Your Step-By-Step Set-up Instructions

- 1 Visit chat.openai.com and create an account using your email address - the free version is perfect for learning to code
- 2 Complete the account verification process and familiarise yourself with the chat interface
- 3 Start a new conversation and introduce yourself as a coding beginner to set the context for future interactions
- 4 Create your first custom instruction by clicking your profile icon and selecting 'Custom instructions'
- 5 In the 'What would you like ChatGPT to know about you?' section, write: 'I'm a complete beginner learning to code. Please explain concepts in simple terms and provide step-by-step guidance.'
- 6 In the 'How would you like ChatGPT to respond?' section, add: 'Use beginner-friendly language, provide code examples, and explain the reasoning behind coding decisions.'
- 7 Test your setup by asking 'Explain what programming is and why someone would want to learn it'
- 8 Create a bookmark folder called 'Coding Learning' and save the ChatGPT link for quick access
- 9 Practice using the conversation history feature to refer back to previous coding lessons and explanations
- 10 Learn to use the copy code button that appears when ChatGPT provides code examples
- 11 Set up a simple note-taking system where you copy important explanations and code examples for later review
- 12 Experiment with follow-up questions like 'Can you explain that differently?' or 'Can you give me a simpler example?' to get explanations that match your understanding level

Your Custom Prompts

For: Learning new programming concepts step-by-step

```
I want to learn about [specific programming concept like variables, functions, loops]. I'm a complete beginner, so please: 1) Explain what it is in simple terms with a real-world analogy, 2) Show me a basic code example with comments explaining each part, 3) Give me a small practice exercise I can try, and 4) Explain common mistakes beginners make with this concept and how to avoid them.
```

For: Code review and improvement suggestions

```
I've written this code as a beginner: [paste your code]. Please review it and provide feedback in this format: 1) What the code does well, 2) What could be improved and why, 3) Suggestions for making it more readable or efficient, 4) One new concept I could learn to enhance this code further. Please keep explanations beginner-friendly and encouraging.
```

For: Creating a personalised learning roadmap

```
I'm a beginner who wants to learn programming for [specific goal like web development, data analysis, game development]. Please create a 12-week learning roadmap that includes: 1) Weekly topics to focus on, 2) Practical projects to build each week, 3) Key concepts to master before moving forward, 4) Resources or practice exercises for each week. Make it realistic for someone spending 5-10 hours per week learning.
```

Setup Tips

Always start your coding questions with context about your skill level to get appropriately tailored responses. Use ChatGPT to explain error messages you encounter - it's excellent at breaking down technical jargon into understandable language. Create separate conversations for different programming topics (one for Python basics, one for web development, etc.) to keep your learning organised. Don't hesitate to ask for multiple explanations of the same concept if the first one doesn't click - ChatGPT can explain things in various ways until you find one that makes sense.

Integrations

Use ChatGPT alongside your VS Code setup with GitHub Copilot - when Copilot suggests code you don't understand, copy it to ChatGPT for a detailed explanation. Combine it with online coding platforms like Codecademy or freeCodeCamp by asking ChatGPT to explain exercises you're stuck on. Create a workflow where you plan your coding projects in ChatGPT, implement them in VS Code with Copilot's help, then return to ChatGPT for code review and improvement suggestions.

Troubleshooting

If ChatGPT's explanations are too technical, explicitly ask for 'simpler terms' or 'explain like I'm 10 years old'. If code examples don't work, mention your operating system and the programming language version you're using. When ChatGPT provides long responses, ask it to 'break this down into smaller steps' or 'focus on the most important parts first'. Remember that ChatGPT can't run code, so always test examples in your own coding environment and report back any issues for clarification.

3. Set Up Replit for Hands-On Practice

Replit provides an instant coding environment in your browser where you can practice programming without installing anything, making it perfect for experimenting and learning through trial and error.

Time: 20-30 minutes **Difficulty:** New to AI [Go to Replit →](#)

Your Step-By-Step Set-up Instructions

- 1 Visit [replit.com](#) and sign up for a free account using your email or GitHub account (using GitHub connects nicely with your Copilot setup)
- 2 Complete the welcome tutorial that introduces you to the Replit interface and basic features
- 3 Click 'Create Repl' and select 'Python' as your first programming language to start with
- 4 Name your first repl 'Learning Python Basics' and explore the three-panel interface: file explorer, code editor, and console
- 5 Write your first program by typing `print("Hello, World!")` and click the green 'Run' button to see the output
- 6 Explore the 'Templates' section and create a new repl using the 'Python with Turtle' template to try graphical programming
- 7 Set up your workspace by creating folders for different topics: 'Week 1 Basics', 'Practice Exercises', 'Mini Projects'
- 8 Install the Replit mobile app on your phone so you can practice coding during spare moments throughout the day
- 9 Join the Replit community by exploring the 'Community' tab and following some beginner-friendly programmers
- 10 Enable the AI assistant feature (Ghostwriter) in your account settings to get coding suggestions similar to Copilot
- 11 Create a 'Daily Practice' repl where you'll solve one small coding challenge each day
- 12 Set up version history by making small changes to your code and using the 'History' tab to see how your code evolves over time

Your Custom Prompts

For: Daily coding challenges for skill building

Give me a beginner-friendly Python coding challenge that I can complete in 15-20 minutes on Replit. The challenge should practice [specific skill like loops, functions, lists] and include: 1) A clear problem description, 2) Example input and expected output, 3) Hints to guide me if I get stuck, 4) A brief explanation of what programming concept this challenge helps me practice. Make it fun and engaging!

For: Project-based learning with step-by-step guidance

I want to build a simple [type of project like calculator, guessing game, to-do list] in Python using Replit. Please provide: 1) A project outline with main features to implement, 2) Step-by-step development plan starting with the simplest version, 3) Code structure suggestions for organising my files, 4) Ideas for how to expand the project once I complete the basic version. Keep it suitable for a beginner who's been coding for [time period].

For: Debugging and code improvement in Replit environment

I'm working on this Python code in Replit: [paste your code]. It's not working as expected and I'm getting [describe the problem or error]. As a beginner, I need help with: 1) Identifying what's wrong with my code, 2) Understanding why this error occurs, 3) Step-by-step instructions to fix the issue.

ions to fix it, 4) Tips for testing my code in Replit to make sure the fix works. Please explain everything in simple terms.

Setup Tips

Use Replit for quick experiments and ChatGPT/Copilot for explanations - this combination gives you both a place to practice and mentors to guide you. Start each coding session by creating a new repl for that day's practice to keep your work organised. Take advantage of Replit's instant sharing feature to get feedback from other learners in coding communities. Use the multiplayer feature to code alongside friends or mentors who can help you learn. Always save your work by giving your repls descriptive names and adding comments to remember what you were learning.

Integrations

Connect Replit with your GitHub account to automatically save your projects and create a portfolio of your coding journey. Use Replit for quick prototyping and testing ideas you discuss with ChatGPT, then move more serious projects to VS Code with Copilot for advanced development. Share your Replit projects in coding communities and social media to get feedback and connect with other learners. Use Replit's database features when you're ready to build projects that save and retrieve information.

Troubleshooting

If your code isn't running, check the console panel for error messages and copy them to ChatGPT for explanation. If Replit feels slow, try using a simpler template or clearing your browser cache. If you can't find your saved repls, check both 'My Repls' and 'Recently Viewed' sections. When sharing repls for help, make sure they're set to public and include clear comments explaining what you're trying to achieve. If the AI assistant gives suggestions you don't understand, disable it temporarily and focus on writing code yourself until you build more confidence.

Curated Learning Resources

YouTube Tutorials

- **Python Programming for Absolute Beginners with AI Tools:** Complete beginner's guide showing how to use ChatGPT and GitHub Copilot while learning Python from scratch, with hands-on examples and common beginner mistakes to avoid
- **Coding with AI: ChatGPT and Copilot for New Programmers:** Step-by-step tutorials showing how new programmers can leverage AI tools effectively without becoming dependent on them, focusing on learning fundamentals
- **Replit Tutorial: Code Anywhere for Beginners:** Comprehensive guide to using Replit for learning programming, including how to set up projects, use templates, and collaborate with other learners

Online Courses

- **Complete Python Bootcamp: From Zero to Hero with AI Tools on Udemy:** Comprehensive programming course designed for absolute beginners, incorporating modern AI tools like ChatGPT and Copilot into the learning process
- **Programming Foundations: Learning to Code with AI Assistance on Coursera:** Structured course that teaches programming fundamentals whilst showing how to effectively use AI tools as learning aids rather than crutches

Advanced Tips & Next Steps

Workflow Automation for Learning to Code

- Set up a daily coding routine where you spend 15 minutes in Replit practicing, 15 minutes reading ChatGPT explanations of concepts, and 30 minutes building projects in VS Code with Copilot assistance
- Create automated reminders to practice coding at the same time each day, and use GitHub to automatically track your coding streak and progress over time
- Use ChatGPT to generate weekly learning goals and practice exercises, then implement them in Replit for quick testing before building larger versions in VS Code

- Set up a feedback loop where you build something in Replit, get it reviewed by ChatGPT, improve it with Copilot's help in VS Code, then share the final version for community feedback

Optimisation Strategies for New to AI Users

- Start with one tool at a time - master ChatGPT for explanations first, then add Copilot for coding assistance, and finally use Replit for experimentation to avoid feeling overwhelmed
- Create a simple learning log where you record new concepts learned each day and review them weekly to reinforce your understanding and track progress
- Use the 'explain this code' feature in all three tools to build understanding rather than just copying solutions - this builds genuine programming knowledge
- Join beginner-friendly coding communities where you can share your AI-assisted learning journey and get encouragement from others on similar paths

Measuring Success in Learning to Code

- Track your daily coding streak using GitHub's contribution graph and celebrate milestones like 7 days, 30 days, and 100 days of consistent practice
- Measure understanding by explaining code concepts to ChatGPT in your own words and asking it to verify your explanations are correct
- Build progressively complex projects each month - start with simple calculators, move to games, then web applications to demonstrate growing skills
- Document your learning journey by writing weekly reflections about what you've learned, what challenged you, and what you want to tackle next

Key Terms Glossary

AI Workflow Automation

Automated processes that use AI to handle repetitive personal coding tasks, specifically designed to help achieve learning to code more efficiently by providing instant feedback and guidance.

Custom Prompt Engineering

Creating specific instructions for AI tools that are tailored to your learning to code objectives and New to AI experience level, ensuring you get beginner-appropriate responses.

Integration Workflow

Connected systems where multiple AI coding tools work together seamlessly to support your learning to code process, from explanation to practice to implementation.

Context-Aware AI

AI systems that understand the specific context of your personal coding learning journey and provide relevant assistance for building programming skills at your pace.

Performance Metrics

Measurable indicators that track your progress toward learning to code effectively and the impact of your AI tool implementation on skill development and confidence building.