Day 2: AI Code assistants

Hany Saad
hghiett@iti.gov.eg

Course: AI-powered Coding Assistant tools

Course Title: AI-powered Coding Assistant tools

Tracks: Development tracks only (Including UI dev, FE and mobile dev tracks)

Duration: 6 hours (Lecture + Lab)

Evaluation:

- Attendance + Lab.

Prerequisites:

- Intro to Generative AI and prompt engineering.
- Preferred to be in last month after core programming courses.

Outlines:

- Intro to AI-powered Coding Assistant tools.
- Benefits, limitations and security considerations.
- Exploring GitHub Copilot features: Code auto-completion, Copilot chat, Multi-file edits, Copilot Workspace.
- Overview on other alternatives: Cursor, TabNine,…
- Demo: on implementing full feature on existing app using GitHub Copilot software agents
- Lab: Installing and exploring GitHub Copilot (Free version)

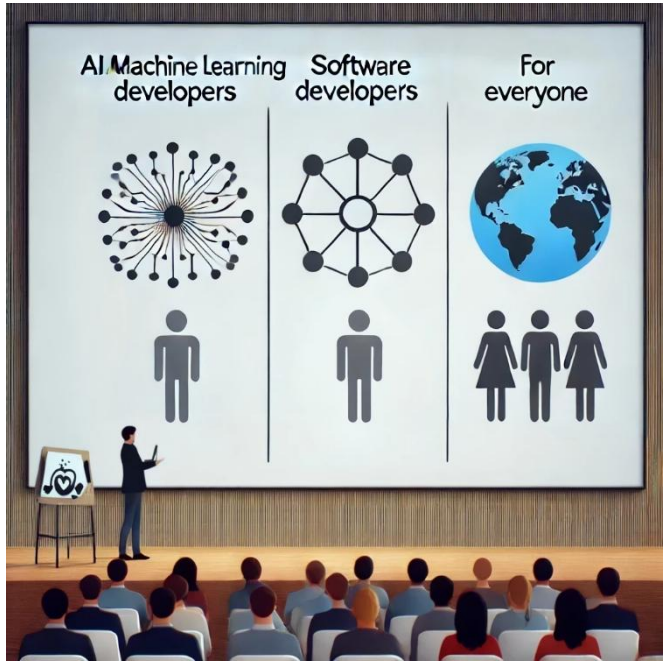
Why do we need to know about Gen AI?

Who should learn about Gen AI?

- **AI / Machine Learning Developers**:
  - Design and refine Generative AI models to create more intelligent and efficient AI-driven solutions.

- **Software Developers**:
  - Utilize AI tools to enhance productivity in development tasks.
  - Leverage Generative AI APIs to build AI-powered applications

- **Everyone**:
  - In today's AI-centric world, it's crucial for individuals across all professions to grasp the basics of Generative AI and incorporate AI tools into their workflows to boost productivity.

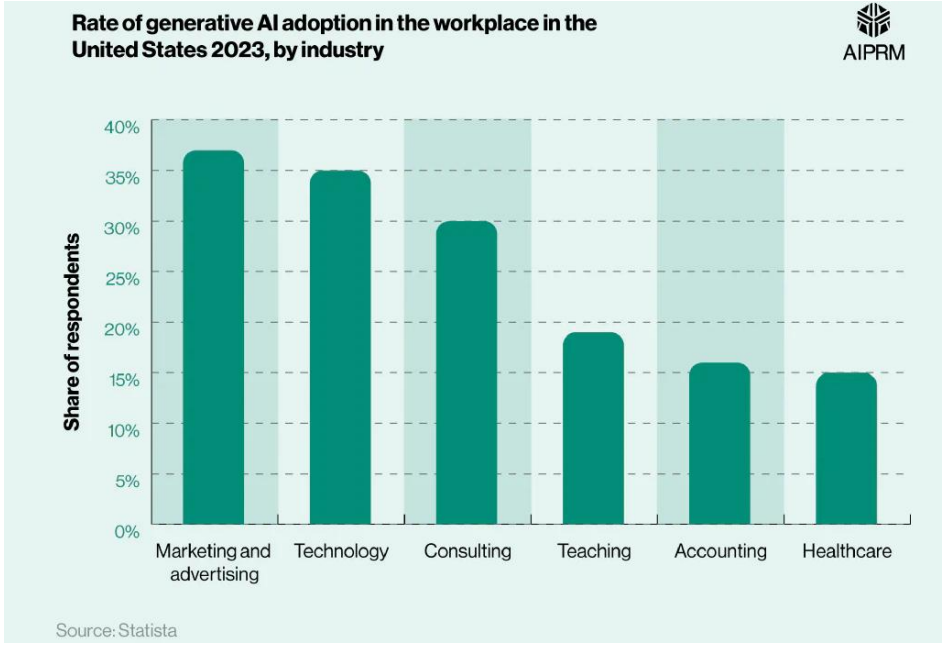


Will AI Replace Jobs / employees?

NEWS

As many as 41% of employers plan to use AI to replace roles—but it's not a 'jobs apocalypse,' experts say

By Ryan Johnston, CNBC • Published February 26, 2025 •



Sample GenAI Freelancing jobs

- **Upwork**
  - AI Image Generation and Deepfake Model Creation
  - AI developer (rag + llm) + fullstack
  - Next.js Developer Needed for SaaS Application Integration with AI Features
  - Developers for 5 AI agents
  - AI Engineer for Medical RAG-based Chatbot Development
  - GenAI engineer to build GenAI-powered modules of an astrology app
  - CReate AI Agent using Open AI or Google AI API or CrewAI
  - Full-Stack Developer with AI Experience (Replit Preferred)
  - Developer Needed to Build AI Agents for Data Analysis and Digital Advertising
  - Web App Development with AI for Meta Ads Management
  - … and more
- **Khamsat and Mostaql**
  - Khamsat Gen AI Services
  - Mostaql Gen AI Projects

Gen-AI based startups and real products

- **Use cases for real products using Gen AI:**
  - Loop-x (AI Agents for different roles): https://loop-x.co
  - Gemelo ai (Twin AI creator): https://gemelo.ai
  - Apriora AI (Recruitment and Interviews): https://www.apriora.ai
  - Velents AI (Recruitment and Interviews): : https://www.velents.com
  - Nancy AI (Recruitment and Interviews): : https://nancy-ai.com
  - Mariana AI (Medical): https://marianaai.com

How to use the AI tools responsibly?

- **As a Student/Learner:**
  - ✅ **Use AI for:** Asking for explanations, self-testing, researching, and similar activities.
  - ❌ **Don't use AI for:** Assignments, projects, tasks, or topics you're still learning.
  - **Build your skills** to work effectively both with and without AI.

- **In a Job or Freelancing:**
  - **Check first** if AI tool usage is permitted and which tools are allowed.
  - **Use AI to boost productivity** and improve your work.
  - **Always review** AI-generated content carefully.
  - **Don't rely on AI** for tasks you have no prior knowledge of.

AI Tools for developers

AI-Powered Coding Assistant Tools

What is an AI coding assistant?

AI coding assistants are software tools or IDE extensions that leverage artificial intelligence to assist developers in writing, understanding, and improving code.

Key Capabilities:

- **Code Suggestions and Auto-completion:** Offers real-time suggestions for completing code blocks, reducing manual typing.
- **Error Detection and Fixes:** Identifies potential errors or bugs and suggests solutions or fixes automatically.
- **Code Generation:** Can generate entire code snippets or functions based on prompts or comments.
- **Code Refactoring:** Helps improve the structure of existing code for better performance or readability.
- **Documentation Support:** Generates or improves comments and documentation based on the code context.

Top AI coding assist tools:

- GitHub Copilot
- Gemini Code Assist
- Cursor AI
- CodiumAI
- Tabnine
- … and more.

What’s GitHub Copilot?

GitHub Copilot is a cloud-based AI pair programming tool developed by GitHub and OpenAI based on OpenAI Codex model to autocomplete and suggest code and entire functions for faster development, It could also generate code from scratch from a well documented code comments.

GitHub Copilot integrates directly into your editor including **Neovim, JetBrains IDEs, Visual Studio, and Visual Studio Code.**

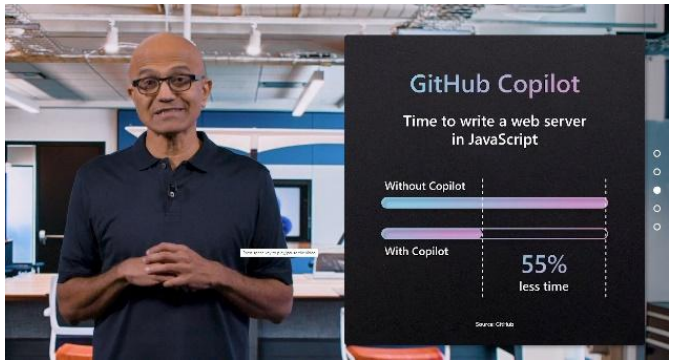
GitHub Copilot provides suggestions for numerous languages and a wide variety of frameworks, but works especially well for **Python, JavaScript, TypeScript, Ruby, Go, C# and C++**.

GitHub Copilot History:

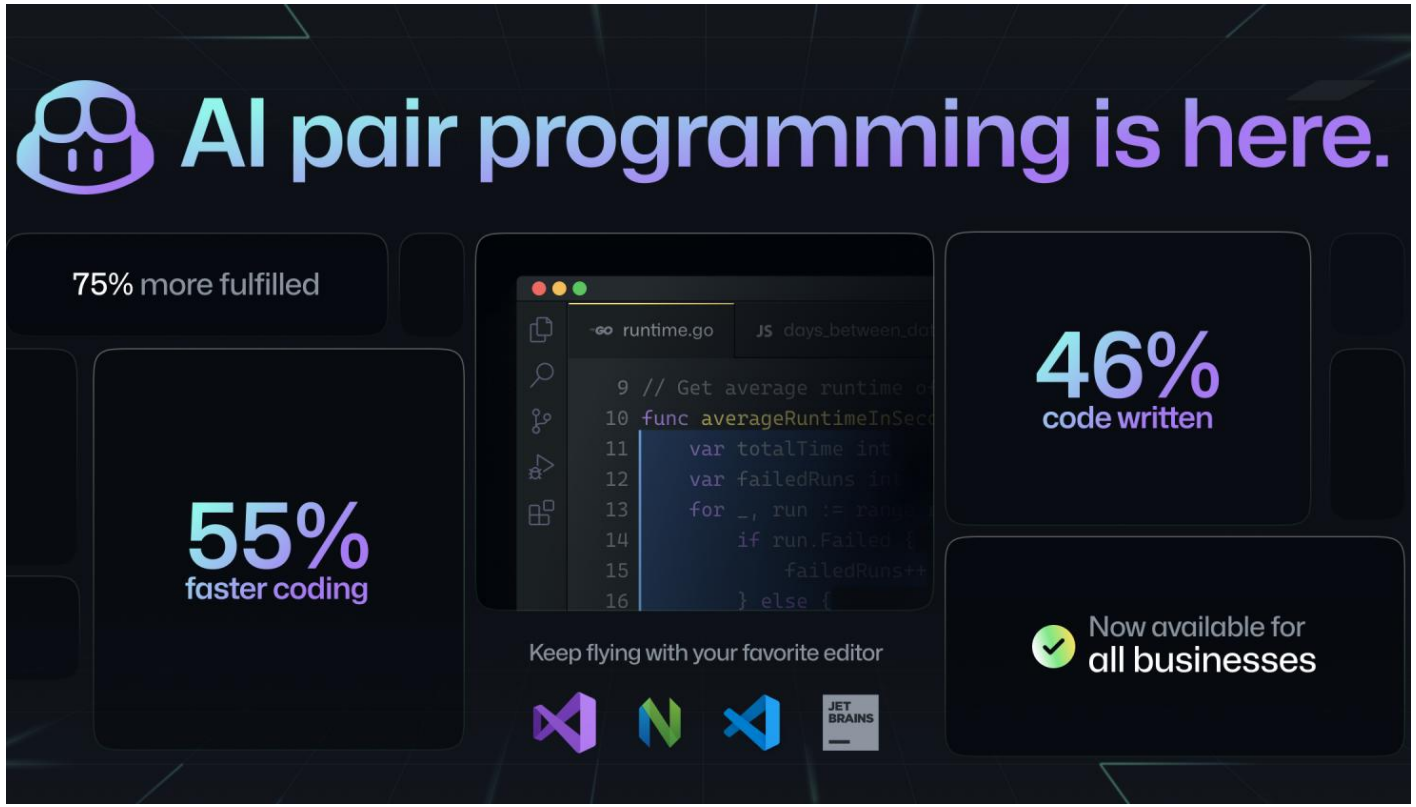
On June 29, 2021, GitHub announced GitHub Copilot for technical preview in the Visual Studio Code development environment.

GitHub announced Copilot's availability for the Visual Studio 2022 IDE on March 29, 2022.

On June 21, 2022, GitHub announced that Copilot was out of "technical preview" and is available as a subscription-based service for individual developers.



GitHub Copilot – stats



GitHub Copilot Use cases

GitHub Copilot use cases:

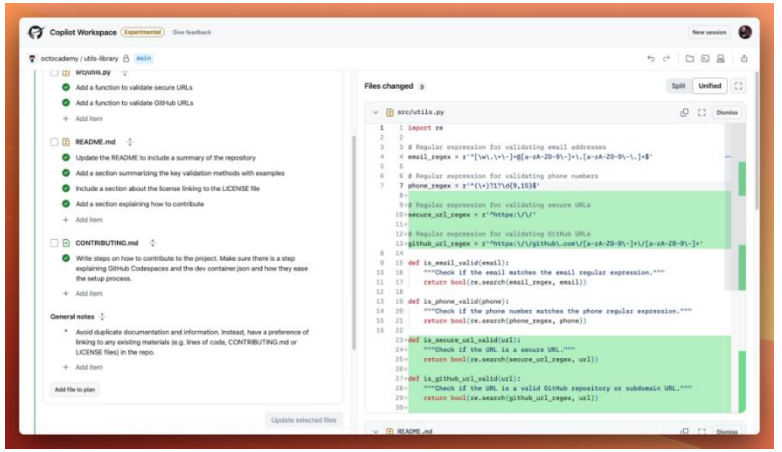
- Code generation and suggestions
- Unit test generation
- Code reviews and refactoring
- Bug fixing
- Generating code documentation and explaining code
- Generating SQL queries
- GitHub Copilot Chat
- Cloud deployment guidance and troubleshooting (e.g., Azure, AWS)
- Automating CI/CD pipeline tasks
- Optimizing code performance
- Command-line tool assistance
- Pull request summaries
- Managing and interacting with cloud resources (e.g., Azure services)
- Translating code between programming languages
- Enhancing security by identifying vulnerabilities
- Auto-generating comments and annotations
- Writing configuration files (e.g., YAML, JSON, Dockerfiles)
- Automating repetitive tasks (e.g., regex creation, cron jobs)
- Real-time feedback on coding best practices
- Knowledge base querying for organization-specific documentation

GitHub Copilot Features

- **Code Completion**: GitHub Copilot provides context-aware, autocomplete-style code suggestions in popular IDEs such as Visual Studio Code, Visual Studio, JetBrains IDEs, Azure Data Studio, and Vim/Neovim (The GitHub Blog) (GitHub Docs).
- **Copilot Chat**: This feature allows you to interact with Copilot via a chat interface. It is available in supported IDEs (Visual Studio Code, Visual Studio, and JetBrains IDEs), on GitHub.com, and in GitHub Mobile. It supports natural language queries for code explanations, debugging help, translating between programming languages, and more. Copilot Chat is powered by GPT-4 and can assist with context-specific coding scenarios (GitHub Docs) (The GitHub Blog) (GitHub Docs).
- **Copilot for the Command Line (CLI)**: This feature extends Copilot's assistance to the terminal. You can use commands like *gh copilot suggest* to get command recommendations or *gh copilot explain* to get explanations of commands. This feature also supports creating command aliases and executing suggested commands directly from the terminal (The GitHub Blog) (GitHub Docs) (GitHub).
- **GitHub Copilot Extensions (Beta):** provide developers with enhanced AI-assisted capabilities tailored to specific workflows or environments. These extensions integrate directly into popular development platforms and tools, offering greater flexibility and customization for coding tasks (The GitHub Blog).
- **GitHub Copilot for Azure (Beta):** is a specialized extension that integrates GitHub Copilot's AI-powered coding assistance with Azure services. It acts as a cloud coding companion that helps developers interact with Azure resources directly from their IDE, enhancing productivity and reducing the complexity of cloud development (TECHCOMMUNITY.MICROSOFT.COM).

GitHub Copilot Features (Cont.)

- **Copilot in GitHub.com:** Integrates directly into the platform for users subscribed to Copilot Individual or Business plans. This allows developers to harness AI directly in their browser without needing to switch to an IDE.
- **Pull Request Summaries** (Enterprise Only): Copilot can automatically generate summaries of pull requests, highlighting changes, their impact on files, and areas of interest for reviewers. This feature saves time during code reviews (The GitHub Blog) (The GitHub Blog).
- **GitHub Copilot workspace:** development environment that helps developers plan, write, and test code using natural language. It supports multi-file changes, integrates with GitHub Codespaces, and allows real-time editing and debugging for seamless collaboration
- **Text Completion** (Enterprise Only): Copilot also assists with natural language tasks, such as generating text descriptions for pull requests and other documentation (GitHub Docs).
- **Knowledge Bases** (Enterprise Only): This feature allows users to create collections of documentation that can be used as a reference for Copilot Chat, enabling more informed answers (GitHub Docs).



GitHub Copilot Features (Cont.)

1. **Slash Commands**
   - Take advantage of Copilot's slash commands such as /fix, /explain, /doc, and /tests for common development tasks.
   - These commands allow you to quickly fix code, generate explanations, create documentation, and write unit tests.
   - Examples:
     - /fix: Proposes a fix for selected code.
     - /explain: Provides a detailed explanation of the selected code.
     - /doc: Generates documentation for the code.
     - /tests: Creates unit tests for the selected code (DEV Community) (VS Code) (VS Code).
2. **Chat Participants (@ Operator)**
   - Use @workspace to instruct Copilot to consider your entire project when responding. This ensures Copilot leverages all open files and configurations in the workspace.
   - Other participants like @terminal help Copilot assist with terminal-related tasks, including shell commands (DEV Community) (VS Code).
3. **File and Section References (# Operator)**
   - Reference specific files, classes, or methods using the #file or #method syntax. This helps Copilot focus on particular sections of your code and improves accuracy.
   - Examples:
     - #BasketService.cs: Points Copilot to a specific file.
     - #AddItemToBasket: Directs Copilot to a particular method (VS Code) (C# Corner).
4. **Inline Chat**
   - Press CMD+I (or CTRL+I on Windows) to bring up inline chat directly within your code.
   - Ask questions, get explanations, or generate code snippets without leaving your current workflow. Inline chat provides context-aware suggestions based on the code you're working on (VS Code) (C# Corner).

More Resources:

https://www.youtube.com/playlist?list=PL0lo9MOBetEHEHi9h0k_lPn0XZdEeYZDS

https://www.youtube.com/playlist?list=PLj6YeMhvp2S5_hvBl2SE-7YCHYlLQ0bPt