

AI Development Tools & Environments

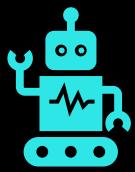
*Beginner-friendly guide to
essential tools and concepts for
getting started in AI
development.*



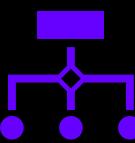
Agenda

-  **Local vs Cloud Development Environments** - Understanding where your code runs
-  **Jupyter Notebooks & Lab** - The foundation of AI development
-  **Python Libraries & Frameworks** - Building blocks for AI projects
-  **AI Code Assistants** - Modern tools that help you write better code faster
-  **Collaboration Platforms** - GitHub, Hugging Face, and Kaggle
-  **Integration Concepts** - How everything works together

How we teach Machines to Learn in AI



ML: Machines learn from examples (data) instead of explicit programming.



It's a subset of AI: Algorithms improve themselves based on past data.

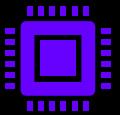


Example: Predicting the price of a house based on size and location.

Traditional Programming vs. ML vs. Deep Learning



Traditional Programming: Give the computer rules; it follows them exactly.



ML: Let the computer learn patterns from data.



Deep Learning: Use very large neural networks to learn complex patterns (like images or text).

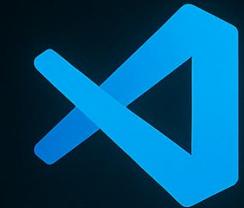
The learning Process

Data: Past examples (what the model learns from).

Algorithm: Math formulas to find patterns in data.

Model: The final product that can make predictions.

New Data: Model can give predictions or decisions for unseen data.



VS Code



AI



PyCharm



Integrated Development Environment (IDE)

- **IDE** is a software application that provides a comprehensive set of tools for software development in a single interface
- **Purpose:** Streamline development process, provide tools for debugging, visualization, and collaboration.
- **IDEs** can be used locally (on your own computer) or in the cloud (online).

Local vs Cloud Environments

- **Local Environment:**

- Runs on your personal computer (laptop/desktop)
- Requires installing Python, Jupyter, and libraries
- Full control over your setup and data
- Works offline once installed
- Limited by your hardware capabilities

- **Cloud Environment:**

- Runs on remote servers accessed through your browser
- No installation needed - just sign up and start coding
- Access to powerful hardware (GPUs, TPUs) for AI training
- Requires internet connection
- Easy sharing and collaboration



UNDERSTANDING JUPYTER NOTEBOOKS

QUESTION MARK **WHAT IS JUPYTER?**

- Interactive coding environment that combines code, text, and visualizations.
- Files saved as **.ipynb** (Jupyter Notebook format)
- Code runs in "cells" - small blocks you can execute individually
- Perfect for data exploration, experimentation, and learning

STAR **WHY JUPYTER FOR AI?**

- See results immediately as you code
- Mix explanations with code (great for learning)
- Easy to share your work with others
- Industry standard for data science and AI development



Jupyter Lab vs Jupyter Notebook

Jupyter Notebook:

- Simple, single-document interface
- Great for beginners and focused work
- Classic interface most tutorials use



Jupyter Lab:

- Advanced interface with multiple tabs and panels
- File browser, terminal, and multiple notebooks in one window
- More powerful for complex projects
- **Same underlying technology** - just a better interface

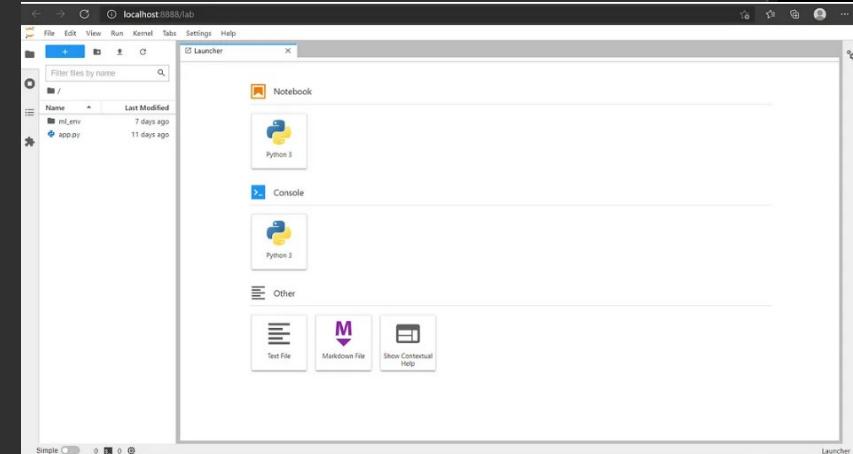
Local Jupyter Setup

□ INSTALLATION OPTIONS

- Anaconda (beginners friendly – Not for AI student's) - includes Python + Jupyter + common libraries
- `pip install jupyter` - if you already have Python

► RUNNING JUPYTER

- 1 Open terminal/command prompt
- 2 Type `jupyter notebook` or `jupyter lab`
- 3 Opens in your web browser at `localhost:8888`
- 4 Create new notebooks or open existing `.ipynb` files



□ **Important:** Even though it opens in a browser, it's running locally on your computer!

VS Code + Jupyter Integration



Visual Studio Code (VS Code):

- Free, powerful code editor by Microsoft
- Can open and run .ipynb files directly
- No need to launch separate Jupyter server
- Integrated terminal, debugging, and extensions

Benefits:

- All-in-one development environment
- IntelliSense code completion
- Git integration for version control
- Extensions for Python, AI libraries, and more

Key Point:

VS Code runs Jupyter notebooks using the same technology - just different interface!



CLOUD JUPYTER ENVIRONMENTS

GOOGLE COLAB

- Free Jupyter notebooks in the cloud
- Free access to GPUs and TPUs for AI training
- Saves to Google Drive automatically
- Great for learning and small projects

AWS SAGEMAKER STUDIO LAB

- Amazon's free cloud Jupyter environment
- More storage and compute than Colab
- Good for larger datasets and longer training

AZURE NOTEBOOKS

- Microsoft's cloud Jupyter service
- Integrated with Azure ML services

GOOGLE COLAB DEEP DIVE

WHY COLAB IS POPULAR

- **Zero setup** - just open in browser and start coding
- **Free GPU access** - essential for deep learning
- **Easy sharing** - share notebooks like Google Docs
- **Pre-installed libraries** - most AI libraries already available



PERFECT FOR

- Learning AI and machine learning
- Experimenting with models
- Sharing tutorials and examples
- Quick prototyping

Access: colab.research.google.com

DON'T GET CONFUSED - THEY'RE ALL JUPYTER!

🔑 IMPORTANT CONCEPT

All these environments use the same core technology

📄 Jupyter notebooks (.ipynb files)



🔢 Python kernel to run code

❓ THE DIFFERENCES ARE

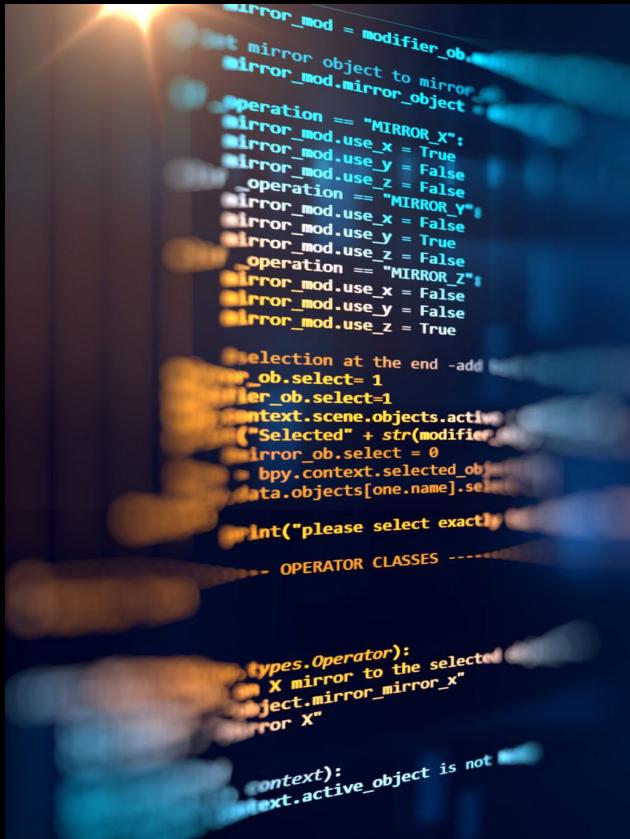
- 💻 Where it runs (local computer vs cloud servers)
- 💻 User interface (classic Jupyter vs VS Code vs Colab)
- ➕ Extra features (AI assistants, collaboration, hardware access)



Bottom line: Learn Jupyter concepts once, use them everywhere!

AI Code Assistant





Benefits of AI code Assistants

- **Increased Productivity:** Automate repetitive coding tasks and provide instant code suggestions.
- **Reduced Errors:** Assist in identifying and correcting coding errors.
- **Improved Learning:** Help new developers understand and write code by providing contextual code snippets and explanations.
- **Enhanced Collaboration:** Facilitate better code documentation and sharing among team members.

Modern AI Code Assistants

Can be web-based, locally installed, or integrated within an IDE, whether hosted in the cloud or on a local machine.

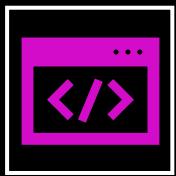


What are AI Code Assistants?

AI-powered tools that help you write code faster and with fewer errors

Suggest code completions, explain errors, and generate functions

Learn from millions of code examples to provide intelligent suggestions

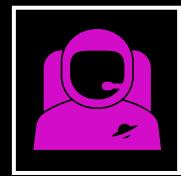


Popular Options:

GitHub Copilot - Integrated into VS Code and other editors

Cursor - AI-first code editor (VS Code fork)

Windsurf - New AI coding environment with advanced features



Latest Development (2025)

OpenAI Codex now available in ChatGPT Plus for \$20/month

Gemini 2.5 powers Google's Code Assist with advanced reasoning

New agentic coding tools that can work autonomously across multiple files

GITHUB COPILOT - THE ESTABLISHED LEADER

-  **WHAT IT DOES**

- Suggests code as you type (like autocomplete on steroids) Can generate entire functions from comments
- Helps explain code and fix errors
- Works in VS Code, JetBrains IDEs, and more

-  **RECENT UPDATES (2025)**

- **Next Edit Suggestions** - suggests changes across multiple files Enhanced code review features
- Integration with GPT-4o for better suggestions

\$ **Pricing:** Free for students, \$10/month for individuals



OPENAI CODEX - NOW ACCESSIBLE TO EVERYONE

★ MAJOR 2025 UPDATE

- Now available in **ChatGPT Plus** for just \$20/month (previously \$200/month)
- Internet access during task execution
- Can install dependencies and run tests automatically

📝 WHAT MAKES IT SPECIAL

- **Agentic behavior**- works autonomously on complex tasks
- Multi-file project understanding
- Draft multiple pull requests in parallel
- Stateless design - requires clear, complete prompts



Perfect for: Complex coding projects requiring autonomous assist



GEMINI 2.5 CODE ASSIST - GOOGLE'S LATEST

POWERED BY GEMINI 2.5 (2025)

- 2.5x better success rate on development tasks
- 2 million token context window for massive codebases
- Advanced reasoning for complex coding challenges

KEY FEATURES

- Chat history and threads - resume where you left off
- Custom rules - "always add unit tests"
- Custom commands - automate repetitive tasks
- Multi-file code suggestions with improved review process



Integration: VS Code, JetBrains IDEs, GitHub app



Free tier available, paid versions for teams

★ CURSOR VS WINDSURF - THE NEW GENERATION

.Cursor - AI-FIRST DEVELOPMENT

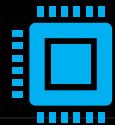
- Built specifically for AI-assisted coding
- **Supercomplete** - Advanced code completion powered by Supermaven
- **Composer** - Generate entire features from descriptions
- Multiple AI models (GPT-4, Claude, etc.)

\$ Free tier, Pro at \$20/month

.Windsurf - RECENTLY ACQUIRED BY OPENAI

- \$3 billion acquisition by OpenAI in 2025
- **Cascade feature** for multi-step code generation
- Can run AI models locally for privacy
- Agent-like behavior for complex tasks

\$ Free tier, Pro at \$15/month



CPU VS GPU - UNDERSTANDING YOUR HARDWARE

CPU (CENTRAL PROCESSING UNIT)

- "The Brain" - Handles general computing tasks
- Few cores (typically 4-16) but very fast
- Sequential processing - does one complex task at a time
- **Great for:** General programming, data preprocessing, small models

GPU (GRAPHICS PROCESSING UNIT)

- "The Muscle" - Originally designed for graphics/gaming
- Thousands of cores but individually slower than CPU cores
- Parallel processing - does many simple tasks simultaneously
- **Perfect for AI:** Matrix operations, neural network training

python™ - THE LANGUAGE OF AI

- Simple, readable syntax - great for beginners
- Massive ecosystem of AI/ML libraries
- Strong community and learning resources
- Industry standard for data science and AI

YOU'LL USE PYTHON IN



Local Jupyter notebooks



VS Code with AI assistants



Google Colab



All cloud platforms



Key Point: The language stays the same regardless of environment!

Introduction to Libraries and Frameworks



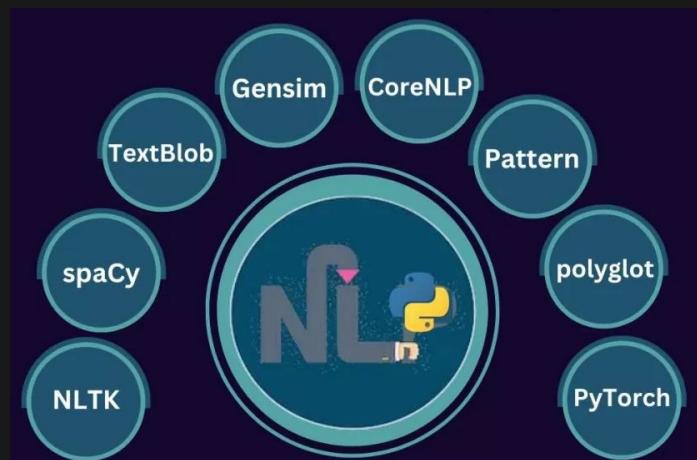
Why are they Important ?

- Accelerate Development: Provides pre-built components for rapid prototyping
- Improve Accuracy: Access to state-of-the-art algorithms and models
- Enhance Collaboration: Standardized tools and environments facilitate teamwork
- Ensure Scalability: Designed to handle growing data and model complexities

UNDERSTANDING LIBRARIES VS FRAMEWORKS

LIBRARIES

- Collections of pre-written functions you can use
- You **call** library functions in your code
- **Examples:** NumPy, Pandas, Matplotlib



FRAMEWORKS

- Provide structure for building entire applications
- You **write code** that fits into the framework's structure
- **Examples:** TensorFlow, PyTorch, Django





Code Repositories & Collaboration Platforms



GitHub - CODE COLLABORATION PLATFORM

⌚ WHAT IS GITHUB?

- Online platform for storing and sharing code
- Uses Git for version control (tracks changes over time) Social network for developers

★ WHY USE GITHUB FOR AI?

- **Portfolio** - Show your projects to employers
- **Collaboration** - Work with others on projects
- **Learning** - Explore open-source AI projects
- **Backup** - Never lose your code

Integration: Works seamlessly with VS Code, Jupyter, and AI assistants



Hugging Face – A Model Hub

WHAT IS HUGGING FACE?

- Platform with thousands of pre-trained AI models
- Focus on Natural Language Processing (NLP) and computer vision
- Easy-to-use interface for testing models

⭐ WHY IT'S VALUABLE

- No training required - use models immediately
- State-of-the-art models - latest research available
- Easy integration - works with your Jupyter notebooks

💡 EXAMPLES

Text classification

Image recognition

Chatbots

Translation

Access: huggingface.co



KAGGLE –LEARNING & PRACTICE PLATFORM

❓ WHAT IS KAGGLE?

- Platform for data science competitions
- Free datasets for practice
- Community of data scientists sharing knowledge

🎓 GREAT FOR LEARNING

- **Real datasets** - practice with actual data
- **Public notebooks** - learn from others' approaches
- **Competitions** - test your skills
- **Free compute** - GPU access for training models



Perfect for building your portfolio and learning from the community

KEY TAKEAWAYS

CORE CONCEPTS

Jupyter notebooks
are the foundation -
same everywhere

Python + libraries
are your building
blocks

AI assistants accelerate
learning and
development

ENVIRONMENT STRATEGY

- Start with one environment, expand gradually
- Cloud for experimentation, local for serious development
- AI assistants are productivity multipliers, not crutches

LEARNING APPROACH

- Focus on understanding, not just using tools
- Build projects to solidify knowledge
- Engage with the community for continuous learning