What is a Generative AI Developer?

A **Generative AI Developer** builds AI systems that can create text, images, music, code, etc., using models like ChatGPT, DALL·E, Stable Diffusion, etc. This involves understanding machine learning, large language models (LLMs), prompt engineering, fine-tuning, and building apps with AI features.

□ Step-by-Step Guide (Beginner to Builder)

□ Step 1: Learn Python Programming □

The universal language of AI and machine learning

- Learn: Variables, functions, loops, classes, file I/O, libraries
- □ Tools: `Python`, `Jupyter`, `VS Code`
- □ Resources:
  - W3Schools Python
  - freeCodeCamp Python Course

□ Step 2: Learn Math for AI □

Focus only on what's essential for generative models

- Topics:
  - Linear Algebra (vectors, matrices)
  - Probability & Statistics (distributions, sampling)
  - Calculus (basic derivatives)
- □ Resource: Khan Academy Math for AI

□ Step 3: Understand Machine Learning □

Learn how machines "learn" patterns from data

- Topics:
  - Supervised/unsupervised learning
  - Regression, classification
  - Overfitting, training, evaluation

- Tools: `Scikit-learn`, `Pandas`, `NumPy`
- □ Resources:
  - Google ML Crash Course
  - Coursera ML by Andrew Ng

□ Step 4: Learn Deep Learning □

This powers neural networks and large generative models

- Topics:
  - Neural networks
  - Activation functions
  - Backpropagation
  - CNNs, RNNs, Transformers
- Tools: `PyTorch` or `TensorFlow` (start with one)
- □ Resources:
  - DeepLearning.ai Intro to DL (Coursera)
  - fast.ai Practical DL

□ Step 5: Explore Generative AI Models □

Learn what powers GPT, DALL·E, Stable Diffusion, etc.

- Learn:
  - What are LLMs (like GPT-4, LLaMA)?
  - How do text-to-image models work?
  - What's a diffusion model?
  - Prompt engineering basics
- Tools:
  - `transformers` by Hugging Face
  - `diffusers` for image generation
  - OpenAI API or open-source models
- □ Try:
  - Hugging Face Transformers Course
  - OpenAI Cookbook

□ Step 6: Build Generative AI Apps □

Learn how to wrap models into usable applications

□ Projects to build:

- A Chatbot using OpenAI API
- An AI image generator (using Stable Diffusion)
- A PDF Q&A bot (RAG + LangChain)
- AI code assistant (like Codex clone)

□ Tools to use:

- `LangChain` – for LLM agents and tools
- `Streamlit` or `Gradio` – for UI
- `FAISS` – for vector search in RAG
- `Flask` or `FastAPI` – for web apps

□ Step 7: Learn about RAG & Fine-tuning □

Retrieval-Augmented Generation (RAG) gives your model context.

- Learn:
  - How to use vector databases (FAISS, Chroma)
  - Embed documents and search them
  - Pass results to an LLM
- Try:
  - LangChain RAG guide
  - LlamaIndex guide

□ Step 8: Host & Share Your AI Projects □

Deploy your apps to the web!

- Tools:
  - `Streamlit Cloud`, `Render`, `Hugging Face Spaces`, `Vercel`
  - `GitHub` for code sharing
- Try:
  - Build a chatbot → Deploy with Streamlit → Share the URL

□ Step 9: Stay Updated & Contribute □□

- Follow:
  - Hugging Face
  - OpenAI
  - Papers with Code
- Contribute to:
  - Open-source LLM agents like OpenDevin, CrewAI, etc.

□ Beginner Tool Stack Summary

Category	Tool / Framework
Programming	Python
Data + ML	Pandas, Scikit-learn
Deep Learning	PyTorch or TensorFlow
LLMs & Generative AI	Hugging Face, OpenAI
UI	Streamlit, Gradio
Agents & RAG	LangChain, FAISS
Hosting	GitHub, Vercel, HF Spaces

□ Learning Plan (Optional)

Week	Focus
1-2	Python + Math Basics
3-4	ML + Data Projects
5-6	Deep Learning Fundamentals
7-8	Transformers + Generative AI
9-10	Build AI Apps + RAG + Deploy

□ OPEN-SOURCE TOOLS FOR GENERATIVE AI (Beginner Edition)

□ 1. Python

□ The foundation language for AI & GenAI

- Why: Every major AI library is based in Python.
- Tools to learn:
  - Jupyter Notebooks
  - VS Code

□ Learn via: W3Schools Python

□ 2. Hugging Face Transformers

□ For text generation, LLMs, chatbots

- Why: Run and fine-tune pre-trained models like GPT-2, LLaMA, BERT.
- Learn:
  - Text classification
  - Text generation
  - Prompting and pipelines

□ Install:

```
pip install transformers
```

□ https://huggingface.co/transformers

□ 3. Diffusers (by Hugging Face)

□ For image generation (e.g. Stable Diffusion)

- Why: Used to build tools like AI art generators.
- Learn:
  - Text-to-image generation
  - Custom prompts and models

 Install:

```
pip install diffusers
```

 https://huggingface.co/docs/diffusers

---

4. LangChain

 For building intelligent LLM-powered agents

- Why: Lets you build chatbots, RAG systems, and agents with tools and memory.
- Learn:
  - Prompt templates
  - Chains and agents
  - Tool use (web search, calculator, etc.)

 Install:

```
pip install langchain
```

 https://docs.langchain.com

---

5. LlamaIndex (aka GPT Index)

 Connect LLMs to your own data (PDFs, docs, etc.)

- Why: Used in Retrieval-Augmented Generation (RAG) systems.
- Learn:
  - Indexing documents
  - Querying private data
  - Chat with PDFs, Notion, websites

 Install:

```
pip install llama-index
```

 https://docs.llamaindex.ai

---

6. FAISS

 Fast vector search for RAG (by Meta AI)

- Why: Lets your AI search large text/vector databases quickly.
- Learn:
  - Create vector embeddings
  - Similarity search
  - Use with LangChain/LlamaIndex

 Install:

```
pip install faiss-cpu
```

 https://github.com/facebookresearch/faiss

---

7. Gradio or Streamlit

 Build simple UIs for your AI apps

- Why: Share your GenAI apps with a web interface.
- Learn:
  - Interactive chatbot apps
  - File upload interfaces
  - Deploy models easily

 Install:

```
pip install gradio
# or
pip install streamlit
```

 https://gradio.app | https://streamlit.io

---

8. Open Source LLMs (Optional but powerful)

 Use models locally, without APIs

- Tools:
  - **LLaMA 2 / 3**: Meta's LLMs
  - **Mistral**: Lightweight open models
  - **Ollama**: Tool to run LLMs locally easily

 Try with:

```
curl -fsSL https://ollama.com/install.sh | sh
ollama run mistral
```

 https://ollama.com

Summary: What to Learn First?

Goal	Tool
Run text models	`transformers`
Generate images	`diffusers`
Build LLM agents	`langchain`
Use your own data (RAG)	`llama-index`, `FAISS`
Build UIs for your models	`streamlit` / `gradio`
Run models locally	`Ollama`, `LLaMA`

Next Steps

1. Start with **Python + Transformers**
2. Build a chatbot or image generator
3. Learn **LangChain** to build agents
4. Try **RAG with LlamaIndex**
5. Build & share your app using **Gradio/Streamlit**