



# Week 0 - Lesson 0: AI Agents Development Environment Setup

## Welcome

This lesson is designed for industry professionals who want to understand and implement AI agents in enterprise environments. We'll cover the essential setup and configuration needed to start building AI agents using Python and modern AI frameworks.

### What You'll Learn:

- Python installation and environment setup
  - Two ways to run notebooks: Jupyter Online and Google Colab (recommended)
  - How to configure your essential API keys: OpenAI and Google AI Studio
  - OpenAI API and Google AI Studio API setup
- 

## Why Python for AI Agents?



Python is the industry standard for AI and machine learning development due to its:

- **Extensive AI/ML libraries:** LangChain, OpenAI, TensorFlow, PyTorch
- **Simple syntax and readability:** Easy to learn and maintain
- **Strong community support:** Large ecosystem and documentation
- **Enterprise adoption:** Used by major tech companies worldwide

## No Python Installation Required!

Both **Jupyter Online** and **Google Colab** come with Python pre-installed, so you

can start coding immediately without any setup.

---

## Development Environment Options

You have **two main options** to run the course notebooks:

### Option 1: Jupyter Online (Quick Start)

- **Best for:** Getting started immediately
- **Requirements:** Just a web browser
- **Limitations:** Temporary environment, limited resources

### Option 2: Google Colab (Recommended)

- **Best for:** Full course experience
- **Requirements:** Google account
- **Advantages:** Persistent storage, GPU access, better interface

**We recommend Google Colab for this course** as it provides a more complete and professional development environment.

---

### Option 1: Jupyter Online (Quick Start)



## What are Jupyter Notebooks?

Jupyter Notebooks are interactive development environments that allow you to:

- Write and execute code in cells
- Mix code with documentation using Markdown
- Visualize data and results inline
- Share reproducible research and experiments

## Key Concepts:

- **Cells**: Individual code or text blocks
- **Kernel**: The computational engine that executes your code
- **Notebook**: A file containing cells and their outputs

## Try Jupyter Online (No Installation Required):

### **Perfect for getting started immediately!**

- Visit [jupyter.org/try](https://jupyter.org/try) - No installation needed!
- Choose from multiple interfaces:
  - **Classic Notebook**: Traditional Jupyter experience
  - **JupyterLab**: Modern, extensible interface
- **Temporary Environment**: Your work is saved until you close the browser

## How to Use Course Notebooks:

1. Download the `.ipynb` files from the course
  2. Upload them to Jupyter Online
  3. Start coding immediately!
-

## Option 2: Google Colab (Recommended)



### What is Google Colab?

Google Colab is a **cloud-based Jupyter notebook environment** by Google that provides:

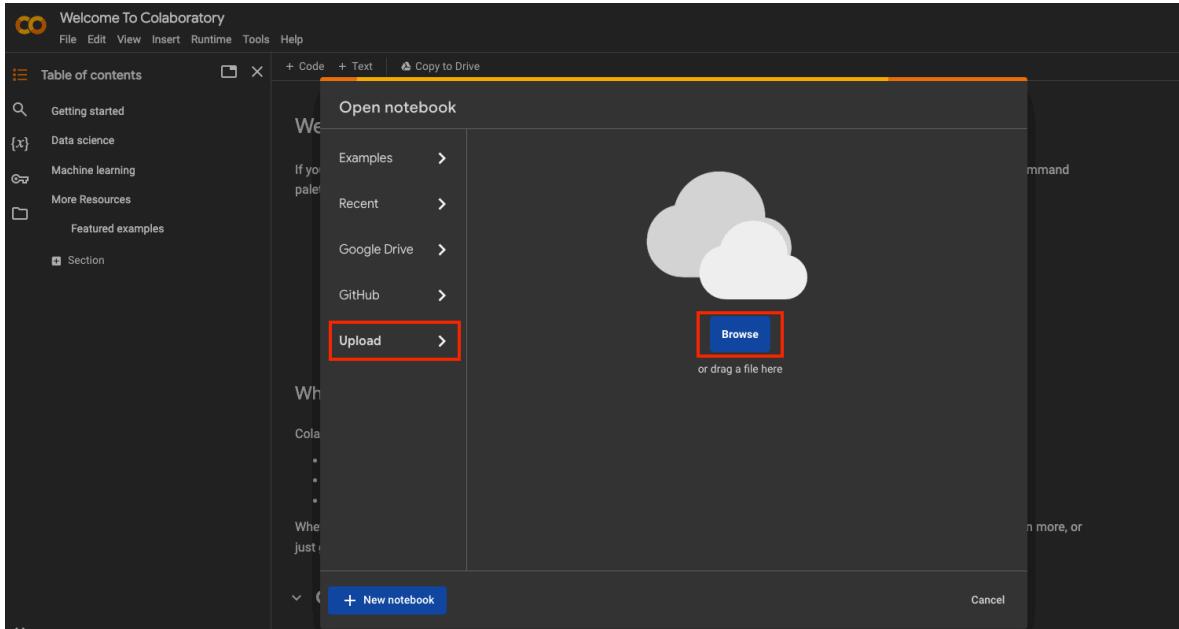
- **No installation required** - runs entirely in your browser
- **Free access to GPUs and TPUs** for faster computation
- **Persistent storage** - your work is saved to Google Drive
- **Easy sharing and collaboration** in real time
- **Professional interface** with enhanced features

### Why We Recommend Colab for This Course:

- **Better Performance:** Access to free GPU resources
- **Persistent Environment:** Your work is automatically saved
- **Professional Features:** Enhanced debugging and development tools
- **Easy Integration:** Seamless connection with Google AI services

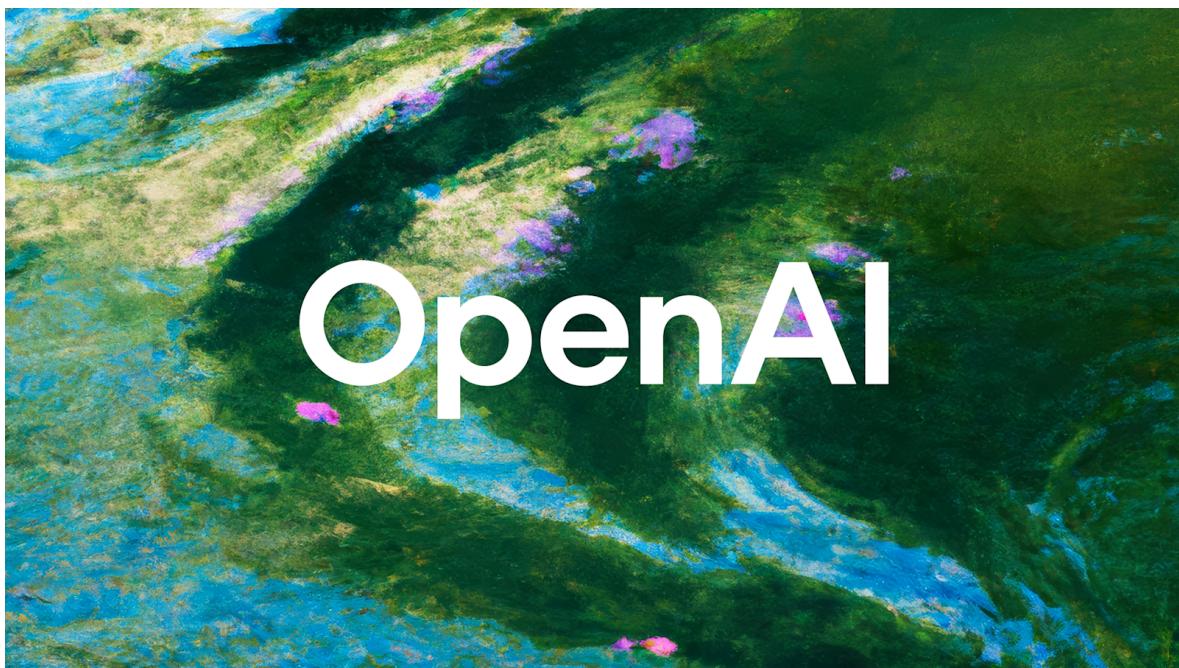
### Getting Started with Colab:

1. Visit [Google Colab](#)
2. Sign in with your Google account
3. Click "New Notebook" to start coding right away
4. **Upload course notebooks:** You can upload `.ipynb` files from your computer or Google Drive



---

## Getting Your OpenAI API Key (Recommended)



## Why OpenAI API?

OpenAI provides the most advanced AI models for agent development:

- **GPT-4**: Most capable model for complex reasoning
- **GPT-3.5-turbo**: Fast and cost-effective for most tasks
- **GPT-4o-mini**: Optimized for speed and efficiency

## Course Usage:

- **Used in:** All lessons
- **Purpose:** Primary AI model provider for agent development
- **Required:** Essential for completing the course

## How to Get Your OpenAI API Key:

1. Visit [OpenAI Platform](#)
2. Sign up or log in to your account
3. Go to **API Keys** section in your dashboard
4. Click "**Create new secret key**"
5. **Important:** Copy and save the key

**Note:** You must add credits to your OpenAI account to use their API services.

For your security, we recommend removing your credit card from the platform after adding credits to avoid unexpected charges.

## Testing your OpenAI key with LangChain

LangChain is a powerful framework that:

- **Provides a unified interface** for different AI providers
- **Simplifies AI agent development** with pre-built components
- **Enables easy switching** between providers (OpenAI, Google, Anthropic, etc.)
- **Offers enterprise-grade features** for production deployments

```
In [ ]: %pip install --quiet langchain langchain-openai python-dotenv
```

```
In [ ]: import os
os.environ["OPENAI_API_KEY"] = "sk-your-openai-key-here"
```

```
In [ ]: # Test OpenAI API with LangChain
from langchain_openai import ChatOpenAI
from langchain.schema import HumanMessage

# Initialize the ChatOpenAI model
try:
    llm = ChatOpenAI(
        model="gpt-3.5-turbo",
        temperature=0.7,
    )
```

```
# Test the connection
response = llm.invoke([HumanMessage(content="Hello! Are you working? Respo
print("OpenAI API connection successful!")
print(f"Response: {response.content}")

except Exception as e:
    print(f"Error connecting to OpenAI: {e}")
    print("Please check your API key")
```

---

## Getting your Google AI Studio API Key (Quick Testing Alternative)



### What is Google AI Studio?

Google AI Studio provides **free access** to Google's Gemini models, offering:

- **High-quality text generation** with Gemini Pro models
- **Integration with Google Cloud services**
- **Easy setup** with Google account

### Why Use Google AI Studio as an Alternative?

**Perfect for quick testing without paid credits!** Google AI Studio is an excellent alternative for:

- **Immediate testing:** No credit card required, just login with your Google account
- **Learning and experimentation:** Ideal for understanding AI providers without financial commitment

- **Rapid prototyping:** Quick setup for testing basic AI agent concepts

## Important Limitations (Why Not Primary Choice):

While the material supports Google AI Studio throughout all notebooks, **we do not recommend it as the primary provider** because:

- **Very low rate/quota limits:** Extremely restrictive for free tier
- **Multi-agent challenges:** Insufficient for advanced multi-agent systems from Week 2 onwards
- **Performance bottlenecks:** Limited requests per minute hinder complex workflows

## Getting Google AI Studio API Key:

1. Visit [Google AI Studio](#)
2. Sign in with your Google account
3. Click "Create API Key"
4. Copy the generated key
5. Add to your `.env` file as `GOOGLE_API_KEY`

## Testing Google AI Studio API

You can use Google AI Studio API as a quick testing alternative to OpenAI. Let's test it:

```
In [ ]: # Install Google AI Studio dependencies
%pip install --quiet langchain-google-genai
```

```
In [1]: # Set up Google AI Studio API Key
import os
os.environ["GOOGLE_API_KEY"] = "your-api-key-here"
```

```
In [ ]: # Test Google AI Studio API with LangChain
from langchain_google_genai import ChatGoogleGenerativeAI
from langchain.schema import HumanMessage

# Initialize the ChatGoogleGenerativeAI model
try:
    llm = ChatGoogleGenerativeAI(
        model="gemini-2.5-flash",
        api_key=os.environ["GOOGLE_API_KEY"],
        temperature=0,
        max_tokens=None,
    )
```

```
# Test the connection
response = llm.invoke([HumanMessage(content="Hello! Are you working? Respo
    print("Google AI Studio API connection successful!")
    print(f"Response: {response.content}")

except Exception as e:
    print(f"Error connecting to Google AI Studio: {e}")
    print("Please check your API key")
```

---

## Important: API Provider Recommendation

### Why We Recommend OpenAI for This Course

While both OpenAI and Google AI Studio are supported throughout all notebooks, we **strongly recommend using OpenAI** for the following reasons:

#### **Google AI Studio Limitations:**

- **Extremely limited rate limits:** Even though it's free, the rate limits are very restrictive
- **Multi-agent difficulties:** Starting from Week 2, you'll face significant challenges running multi-agent systems
- **Performance bottlenecks:** Limited requests per minute make complex agent interactions problematic

### Course Design Philosophy

All notebooks in this course are designed to work with **both providers**, but we prioritize OpenAI:

- **Flexibility:** You can always switch between providers
- **OpenAI-first:** Examples and optimizations favor OpenAI
- **Fallback option:** Google AI Studio remains available for simple tasks

**Recommendation:** Start with OpenAI for the best learning experience, especially from Week 2 onwards.

---

# LangChain Provider Flexibility

One of LangChain's key advantages is easy provider switching. All notebooks support both providers:

## OpenAI (Recommended)

```
from langchain_openai import ChatOpenAI
llm = ChatOpenAI(model='gpt-3.5-turbo')
```

## Google AI Studio (Alternative)

```
from langchain_google_genai import ChatGoogleGenerativeAI
llm = ChatGoogleGenerativeAI(model='gemini-2.5-flash')
```

## Other Providers Available

```
from langchain_anthropic import ChatAnthropic
llm = ChatAnthropic(model='claude-3-sonnet-20240229')
```

**Same interface, different providers!** This flexibility allows you to:

- Switch between providers easily
- Compare performance across different models
- Use the best provider for each specific task

For more information: <https://python.langchain.com/docs/integrations/providers/>

---

## Quick Reference:

- **OpenAI API:** [platform.openai.com](https://platform.openai.com)
- **Google AI Studio:** [aistudio.google.com](https://aistudio.google.com)
- **Google Colab:** [colab.research.google.com](https://colab.research.google.com)
- **Jupyter Online:** [jupyter.org/try](https://jupyter.org/try)

**Happy coding!** ☺