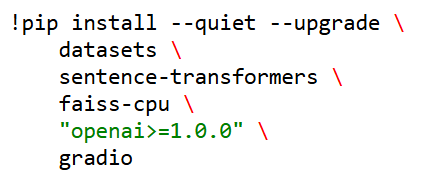
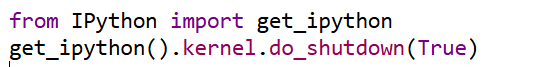
**1. RAG\_OpenAI.ipynb**

1. **Cell 1: Install Dependencies**



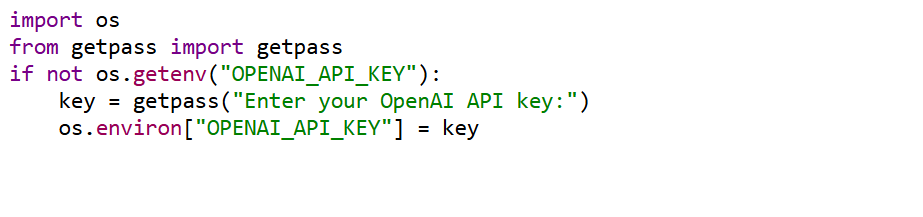
* **Purpose:** Installs all Python libraries needed (HuggingFace datasets, embedding models, FAISS, OpenAI SDK v1.x, Gradio).  
  **Notes:** Forces openai>=1.0.0 so we can use the new client API.

1. **Cell 2: Restart Kernel**



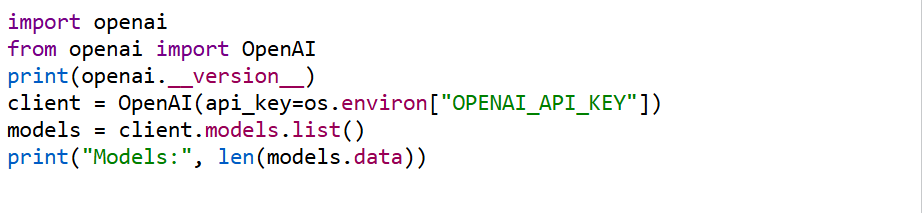
* **Purpose:** Clears the runtime so that the upgraded packages take effect.  
  **Notes:** This is a common Colab trick after pip upgrades. We can even do it manually as well.

1. **Cell 3: Prompt for OpenAI Key**



* **Purpose:** Securely prompts the user to input their OpenAI API key at runtime (no echo).  
  **Input:** None  
  **Output:** Sets OPENAI\_API\_KEY in os.environ.

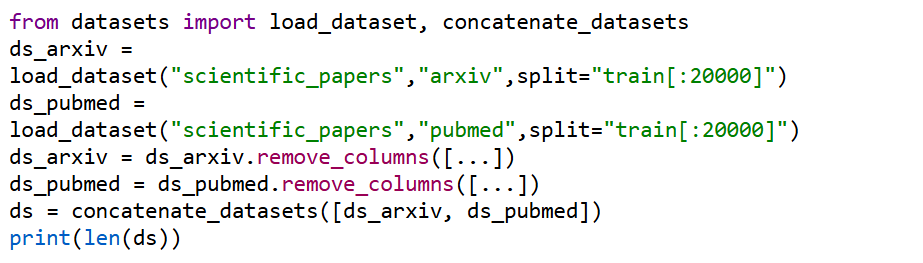
1. **Cell 4: Verify & Instantiate OpenAI Client**



**Purpose:**

* Checks that the OpenAI SDK is v1.x
* Creates a reusable client for chat and embeddings
* Lists available models to confirm authentication.  
  **Output:** Number of models your key can access.

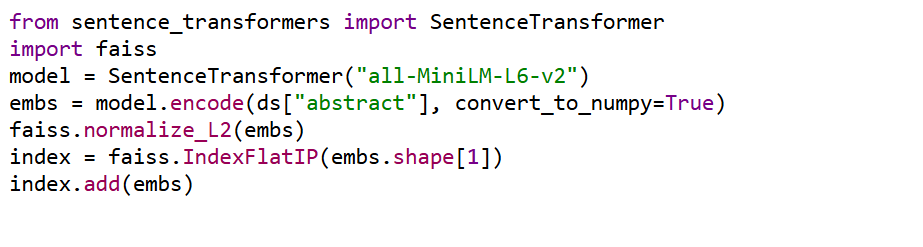
1. **Cell 5: Load & Combine Abstracts**



**Purpose:**

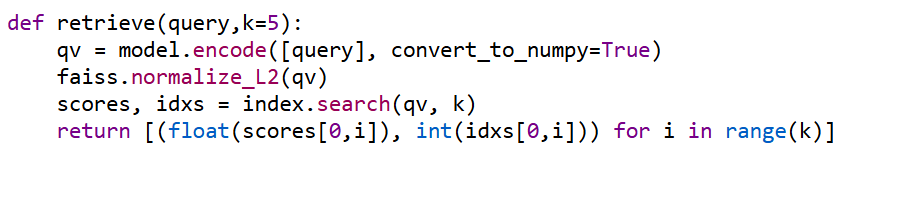
* Loads 20 000 abstracts from arXiv and PubMed
* Keeps only the abstract field
* Merges into a single ds for embedding.  
  **Output:** Combined dataset with ~40k abstracts.

1. **Cell 6: Embed & Build FAISS Index**

**Purpose:**

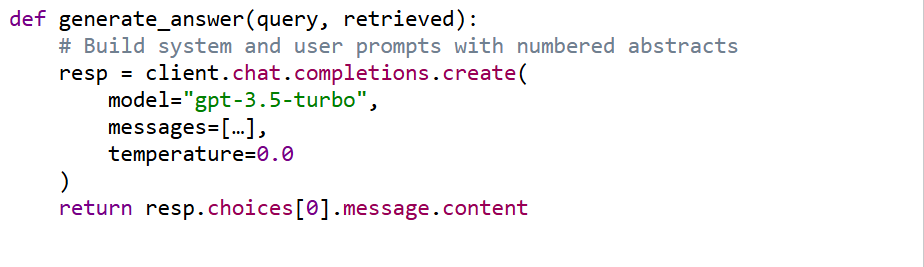
* Loads a lightweight embedding model
* Generates embeddings for every abstract
* Normalizes and indexes them for cosine similarity search.  
  **Output:** A FAISS index ready for retrieval.

1. **Cell 7: Define Retrieval Function**



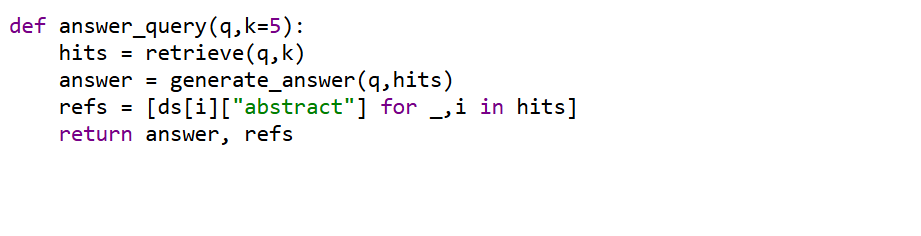
* **Purpose:** Encapsulates semantic search: embed the query, normalize, run FAISS .search().  
  **Output:** List of (score, dataset\_index) for the top‑k abstracts.

1. **Cell 8: Define Answer Generator**

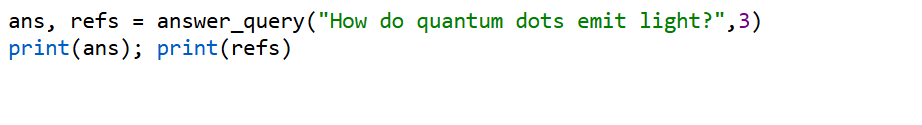
**Purpose:**

* Formats the prompt so the LLM cites each abstract by index
* Calls the ChatCompletion API
* Returns the raw answer text.  
  **Notes:** Uses a system prompt to enforce “cite with [n]”.

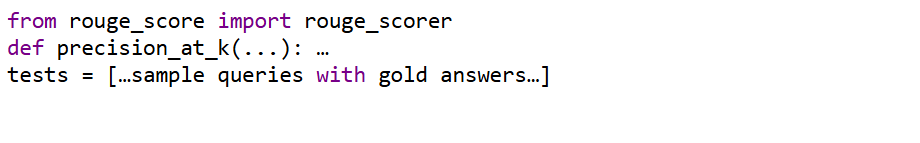
1. **Cell 9: Wrapper answer\_query()**

**Purpose:** Combines retrieval + generation into one function.  
**Output:** Tuple (answer\_text, [abstracts]).

1. **Cell 10: Sample Test**

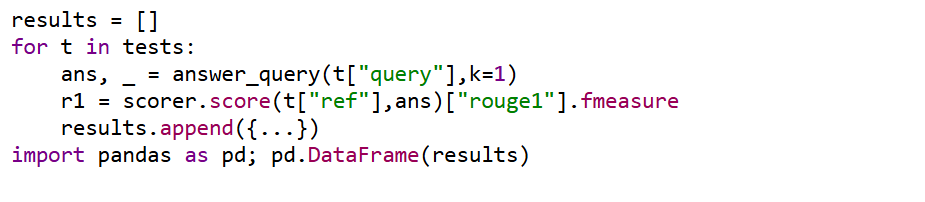
**Purpose:** Quick sanity check to verify end‑to‑end.

1. **Cell 11: Evaluation Setup**

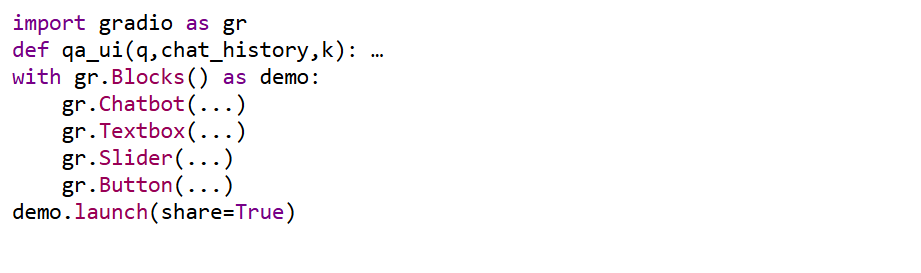
**Purpose:**

* Imports ROUGE for answer evaluation
* Defines IR metrics (Precision@K)
* Prepares a small test set.

1. **Cell 12: Run Evaluation**

**Purpose:** Computes retrieval & generation metrics, displays a DataFrame.

1. **Cell 13: Gradio UI Setup**

**Purpose:**

* Builds an interactive two‑column chat UI
* Passes user input → answer\_query() → displays answer + clickable abstracts.
* Adds a day/night theme toggle via custom CSS/HTML.

1. **Cell 14–17: CSS & Toggle**

* Defines CSS variables for light/dark themes
* Inserts a <button> + <script> to toggle the dark class
* Ensures the UI is accessible with aria-label & tooltips.