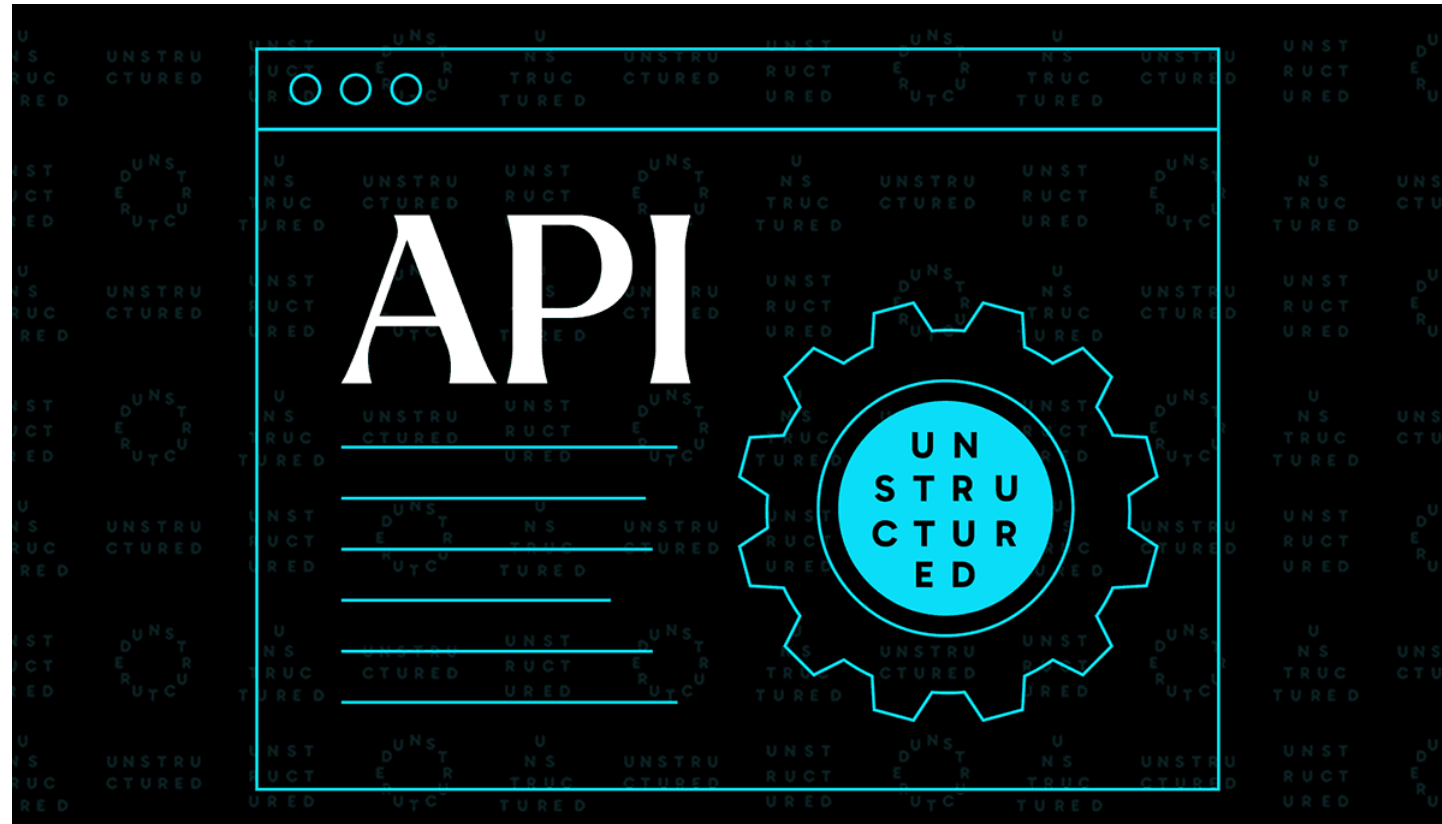


LLM API



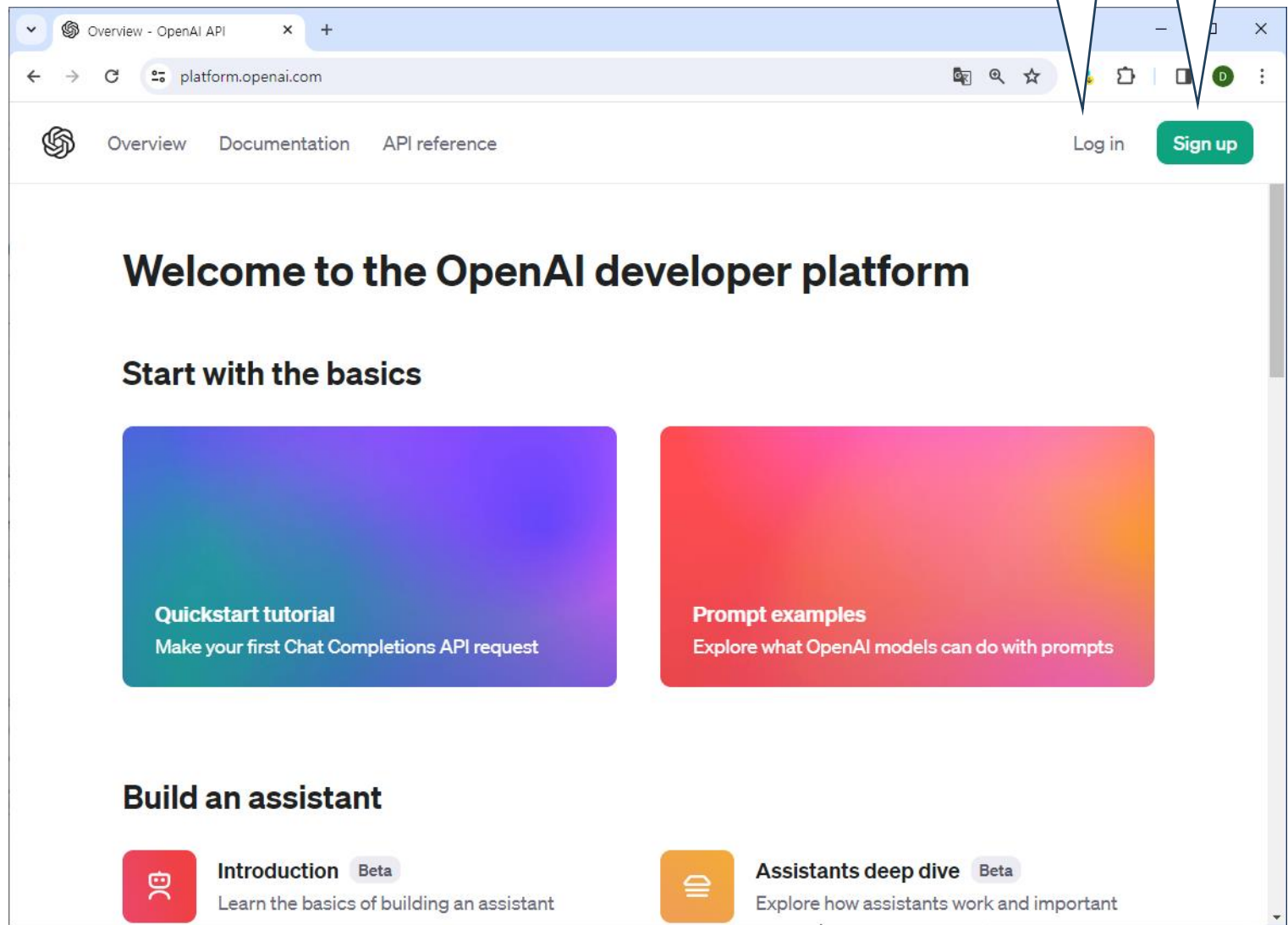
OpenAI API



OpenAI

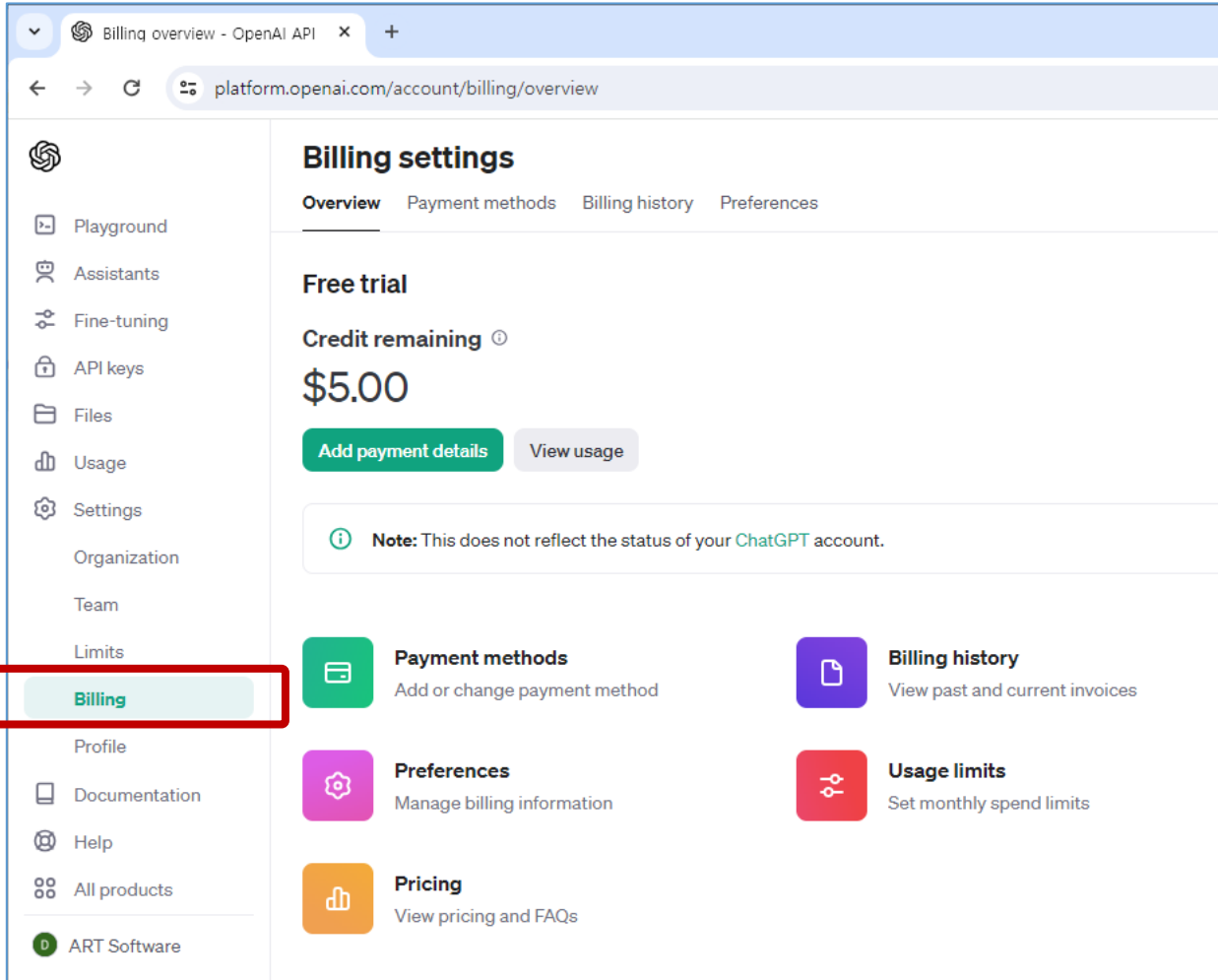
<https://platform.openai.com/>

사이트 접속 및 회원가입



OpenAI API 무료사용

<https://platform.openai.com/account/billing/overview>



Rate limits

MODEL	TOKEN LIMITS	REQUEST AND OTHER LIMITS
gpt-3.5-turbo : LLM	40,000 TPM	3 RPM 200 RPD
text-embedding-3-small	150,000 TPM	3 RPM 200 RPD
dall-e-3 : Text to Image		3 RPM 200 RPD
tts-1 : Text to Speech		3 RPM 200 RPD
whisper-1 : Automatic Speech Recognition		3 RPM 200 RPD

- TPM (tokens per minute)
- TPD (tokens per day)
- RPM (requests per minute)
- RPD (requests per day)
- IPM (images per minute)

- 1 token \approx 4 chars in English
- 1 token \approx $\frac{3}{4}$ words
- 100 tokens \approx 75 words

참고 :

<https://help.openai.com/en/articles/4936856-what-are-tokens-and-how-to-count-them>

OpenAI API 유료사용

<https://platform.openai.com/account/billing/overview>

The image shows the OpenAI API Billing overview page. The left sidebar contains navigation links: Playground, Assistants, Fine-tuning, API keys, Storage, Usage, Settings, Organization, Team, Limits, Billing (highlighted with a red box), Profile, Documentation, and Help. The main content area is titled "Billing settings" and has tabs for Overview, Payment methods, Billing history, and Preferences. Under the Overview tab, it shows "Free trial" and "Credit remaining \$3.77". A red box highlights the "Add payment details" button, with a blue arrow pointing to a modal window. The modal is titled "Add payment details" and contains a message: "Add your credit card details below. This card will be saved to your account and can be removed at any time." Below this, there are sections for "Card information" (with a field for "카드 번호" and "MM / YY CVC"), "Name on card", and "Billing address" (with fields for "Country", "Address line 1", "Address line 2", "City", "Postal code", and "State, county, province, or region"). At the bottom of the modal are "Cancel" and "Continue" buttons. In the background, there is a "What best describes you?" dialog with options for "Individual" (I'm an individual) and "Company" (I'm working on behalf of a company), with a blue arrow pointing from the "Add payment details" button to the "Individual" option.

Billing settings

Overview Payment methods Billing history Preferences

Free trial

Credit remaining ⓘ

\$3.77

Add payment details

ⓘ Note: This does not reflect the status of your account.

Payment methods
Add or change payment method

Billing history
View past and current invoices

Preferences
Manage billing information

Usage limits
Set monthly spend limits

Add payment details

Add your credit card details below. This card will be saved to your account and can be removed at any time.

Card information

카드 번호 MM / YY CVC

Name on card

Billing address

Country

Address line 1

Address line 2

City Postal code

State, county, province, or region

Cancel Continue

OpenAI API Key 생성

The image shows a sequence of steps to create an OpenAI API key, overlaid on a browser window. The steps are numbered 1 through 5:

- 1**: Click the OpenAI logo in the top-left corner of the browser window.
- 2**: Click the 'API keys' option in the left-hand navigation menu.
- 3**: Click the '+ Create new secret key' button at the bottom of the 'API keys' page.
- 4**: In the 'Create new secret key' dialog, enter a name (e.g., 'Test Key') in the 'Name' field.
- 5**: In the 'Save your key' dialog, click the 'Copy' button to copy the generated API key.

The background browser window shows the 'API keys' page with a table of existing keys. The table has columns for 'NAME' and 'Permissions'. One key is listed with the name 'OpenAIAPIKey' and 'All' permissions. The 'Create new secret key' dialog is open, showing the 'Name' field with 'Test Key' and the 'Permissions' dropdown set to 'All'. The 'Save your key' dialog is also open, showing the generated key 'DxB3KJa81wkWJT7T3B1bkFJCZFHpjisWsSc' and a 'Copy' button.

OpenAI 요금제

<https://openai.com/pricing>

GPT

Model	Input	Output
gpt-4-0125-preview	\$0.01 / 1K tokens	\$0.03 / 1K tokens
gpt-4-1106-preview	\$0.01 / 1K tokens	\$0.03 / 1K tokens
gpt-4-1106-vision-preview	\$0.01 / 1K tokens	\$0.03 / 1K tokens
gpt-4	\$0.03 / 1K tokens	\$0.06 / 1K tokens
gpt-4-32k	\$0.06 / 1K tokens	\$0.12 / 1K tokens
gpt-3.5-turbo-0125	\$0.0005 / 1K tokens	\$0.0015 / 1K tokens
gpt-3.5-turbo-instruct	\$0.0015 / 1K tokens	\$0.0020 / 1K tokens

Embedding models

Model	Usage
text-embedding-3-small	\$0.00002 / 1K tokens
text-embedding-3-large	\$0.00013 / 1K tokens
ada v2	\$0.00010 / 1K tokens

토큰(Token)

<https://platform.openai.com/tokenizer>

The screenshot shows the OpenAI Platform tokenizer interface. The browser address bar displays `platform.openai.com/tokenizer`. The page has a navigation bar with links for Overview, Documentation, and API reference, along with Log in and Sign up buttons. The main content area is titled "GPT-3.5 & GPT-4" and "GPT-3 (Legacy)". It contains a text input field with the following text: "Many words map to one token, but some don't: indivisible. Unicode characters like emojis may be split into many tokens containing the underlying bytes: 🍌 Sequences of characters commonly found next to each other may be grouped together: 1234567890". Below the input field are "Clear" and "Show example" buttons. The output section shows "Tokens: 57" and "Characters: 252". The text is displayed with color-coded tokens. At the bottom, there are "Text" and "Token IDs" tabs.

OpenAI Platform

platform.openai.com/tokenizer

Overview Documentation API reference Log in Sign up

GPT-3.5 & GPT-4 GPT-3 (Legacy)

Many words map to one token, but some don't: indivisible.

Unicode characters like emojis may be split into many tokens containing the underlying bytes: 🍌

Sequences of characters commonly found next to each other may be grouped together: 1234567890

Clear Show example

Tokens Characters

57 252

Many words map to one token, but some don't: indivisible.

Unicode characters like emojis may be split into many tokens containing the underlying bytes: 🍌🍌🍌🍌🍌

Sequences of characters commonly found next to each other may be grouped together: 1234567890

Text Token IDs

Tokens Characters

57 252

[8607, 4339, 2472, 311, 832, 4037, 11, 719, 1063, 1541, 956, 25, 3687, 23936, 382, 35020, 5885, 1093, 100166, 1253, 387, 6859, 1139, 1690, 11460, 8649, 279, 16940, 5943, 25, 11410, 97, 248, 9468, 237, 122, 271, 1542, 45045, 315, 5885, 17037, 1766, 1828, 311, 1855, 1023, 1253, 387, 41141, 3871, 25, 220, 4513, 10961, 16474, 15]

Text Token IDs

토큰 한도 (Token limit)

<https://platform.openai.com/docs/guides/text-generation/managing-tokens>

총 토큰 수는 API 호출에 영향을 줌

- 모델의 최대 한도 미만이어야 함, gpt-3.5-turbo 토큰 한도 4,096

총 토큰 수 : 입력 토큰 + 출력 토큰

- 토큰당 지불하는 API 호출 비용

- API 호출에 걸리는 시간

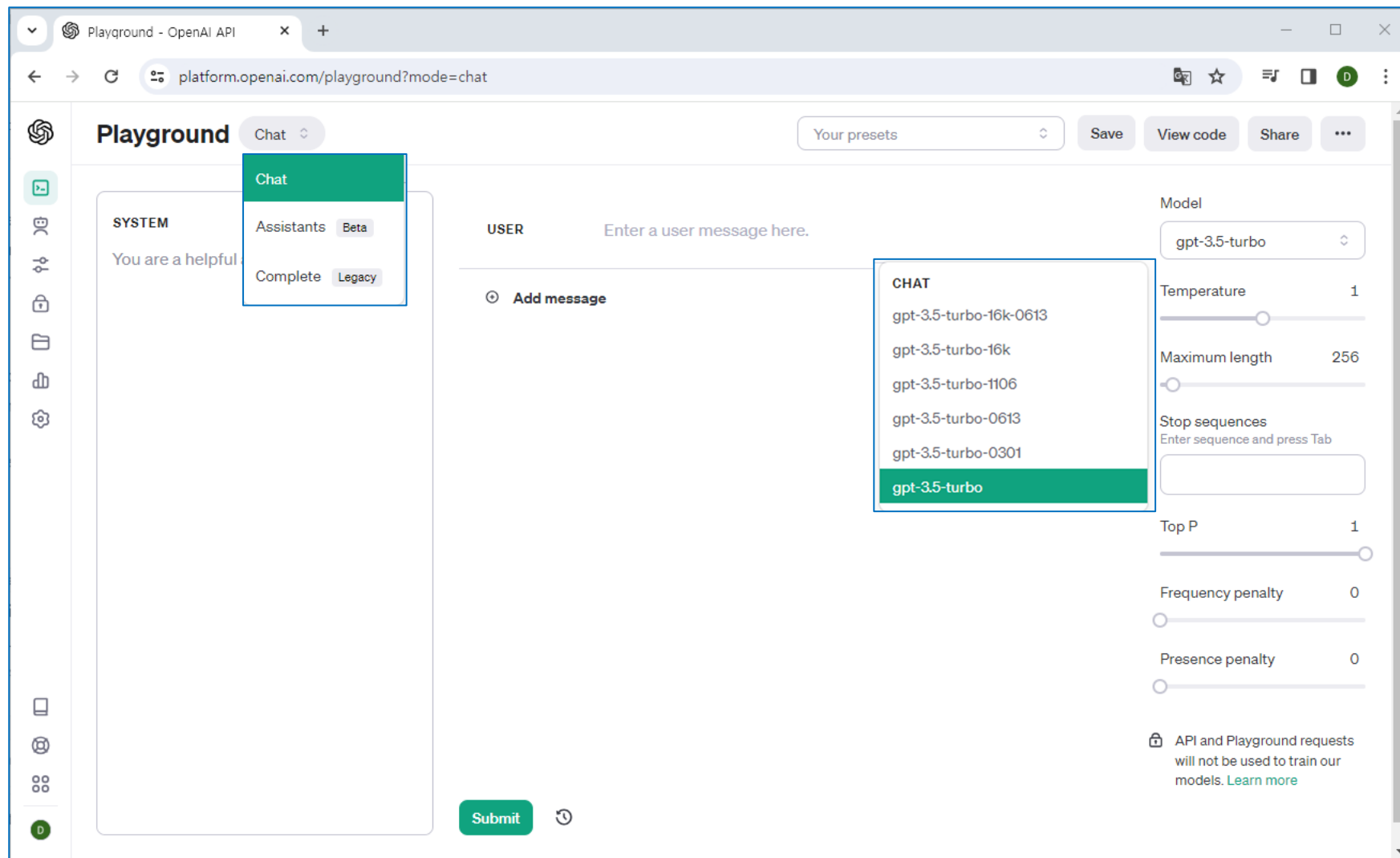
Counting tokens for chat API calls

```
1 def num_tokens_from_messages(messages, model="gpt-3.5-turbo-0613"):
2     """Returns the number of tokens used by a list of messages."""
3     try:
4         encoding = tiktoken.encoding_for_model(model)
5     except KeyError:
6         encoding = tiktoken.get_encoding("cl100k_base")
7     if model == "gpt-3.5-turbo-0613": # note: future models may deviate from this
8         num_tokens = 0
9         for message in messages:
10             num_tokens += 4 # every message follows <im_start>{role/name}</im_start>
11             for key, value in message.items():
12                 num_tokens += len(encoding.encode(value))
13                 if key == "name": # if there's a name, the role is omitted
14                     num_tokens += -1 # role is always required and always 1 token
15             num_tokens += 2 # every reply is primed with <im_start>assistant
16         return num_tokens
17     else:
18         raise NotImplementedError(f"num_tokens_from_messages() is not presently implemented for model {model}")
19     See https://github.com/openai/openai-python/blob/main/chatml.md for information on message formatting
```

```
1 messages = [
2     {"role": "system", "content": "You are a helpful, pattern-following assistant."},
3     {"role": "system", "name": "example_user", "content": "New synergies will help drive growth."},
4     {"role": "system", "name": "example_assistant", "content": "Things work great! In fact, this new synergy is going well."},
5     {"role": "system", "name": "example_user", "content": "Let's circle back to the beginning and make sure everyone is aligned around the shared objectives."},
6     {"role": "system", "name": "example_assistant", "content": "Let's talk about the shared objectives."},
7     {"role": "user", "content": "This late pivot means we don't have time to waste."}
8 ]
9
10 model = "gpt-3.5-turbo-0613"
11
12 print(f"{num_tokens_from_messages(messages, model)} prompt tokens counted")
13 # Should show ~126 total_tokens
```

플레이그라운드

<https://platform.openai.com/playground>



- Temperature : 값이 낮을수록 가장 높은 확률의 다음 토큰을 선택하고, 높아지면 무작위성이 높아짐
- Max Length : 모델이 생성하는 토큰 최대 길이
- Stop Sequences : 모델의 토큰 생성을 중지하는 문자열
- Top P : 값이 높으면 모델이 가능성이 낮은 단어를 포함하여 더 다양한 출력을 얻을 수 있음
- Frequency Penalty : 해당 토큰이 나타난 횟수에 비례하여 페널티 적용
- Presence Penalty : 모든 반복 토큰에 동일한 페널티 적용(2번 나타나는 토큰과 10번 나타나는 토큰 모두 동일한 페널티)

※ Temperature 와 Top_p, 그리고 Frequency Penalty와 Presence Penalty 동시 변경은 비권장함

API 사용 방법

Step 1: Setup Python

✓ Install Python

<https://www.python.org/downloads/>

✓ Setup a virtual environment (optional)

`python -m venv myenv`

Windows : `myenv\Scripts\activate`

Unix or Mac : `source myenv/bin/activate`

✓ Install the OpenAI Python library

`pip install --upgrade openai`

Step 2: Setup your API key

Windows : `setx OPENAI_API_KEY "your-api-key-here"`

Unix or Mac : `export OPENAI_API_KEY='your-api-key-here'`

Step 3: Sending your first API request

```
1 from openai import OpenAI
2 client = OpenAI()
3
4 completion = client.chat.completions.create(
5     model="gpt-3.5-turbo",
6     messages=[
7         {"role": "system", "content": "You are a poetic assistant, skilled in explaining"},
8         {"role": "user", "content": "Compose a poem that explains the concept of recursi"},
9     ]
10 )
11
12 print(completion.choices[0].message)
```

OpenAI API 실습



openai_api.ipynb

How_to_count_tokens_with_tiktoken.ipynb  **GitHub**

chunking.ipynb

information_retrieval.ipynb

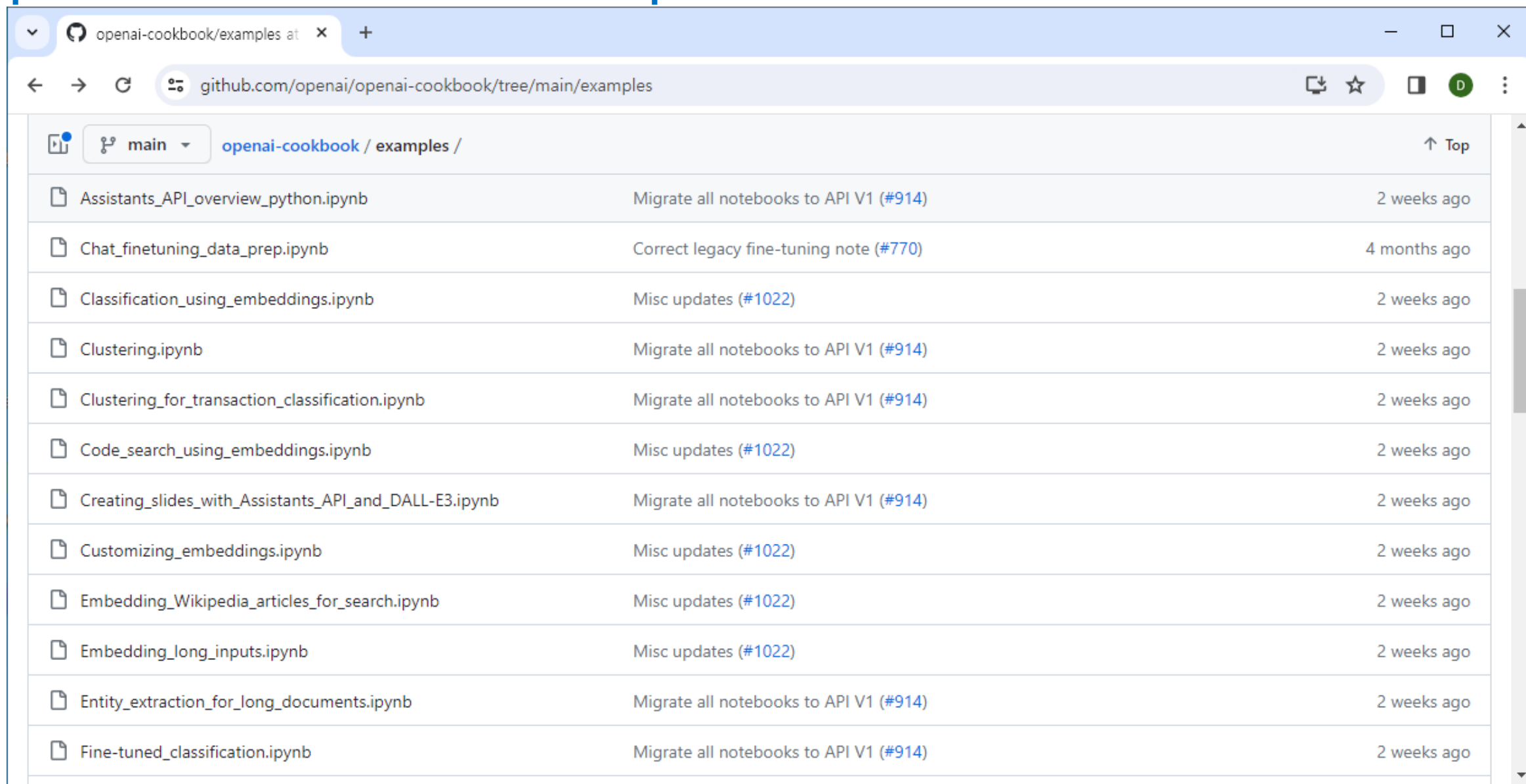
ReAct.ipynb

pe-lecture.ipynb

colab

OpenAI Cookbook examples

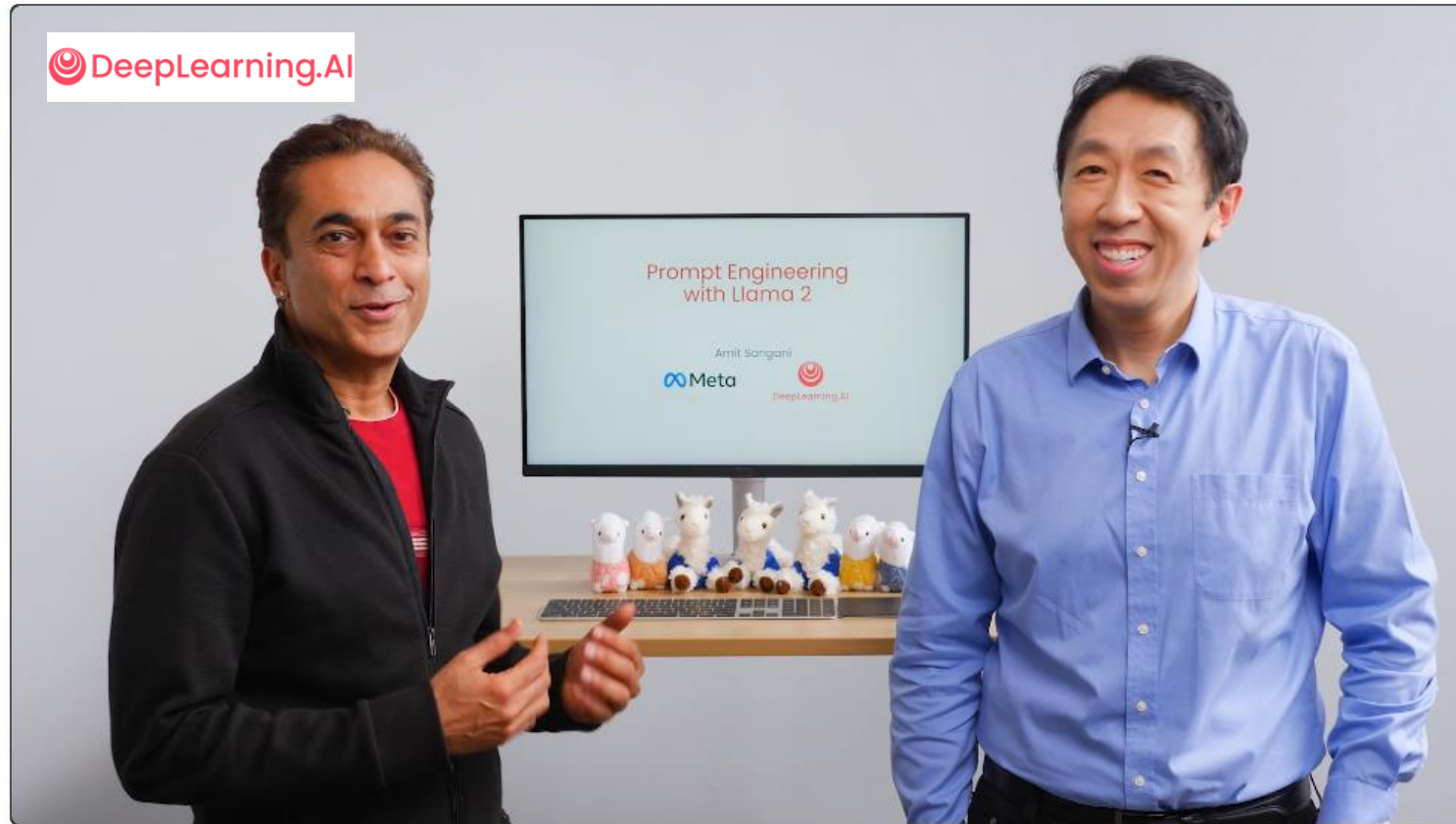
<https://github.com/openai/openai-cookbook/tree/main/examples>



The screenshot shows a web browser displaying the GitHub repository page for OpenAI Cookbook examples. The browser's address bar shows the URL `github.com/openai/openai-cookbook/tree/main/examples`. The page header includes a breadcrumb trail: `openai-cookbook / examples /` with a 'main' branch selector. A list of 12 Jupyter Notebook files is displayed in a table, each with a file icon, the filename, a description of the update, and the time since the last update. The files are: `Assistants_API_overview_python.ipynb`, `Chat_finetuning_data_prep.ipynb`, `Classification_using_embeddings.ipynb`, `Clustering.ipynb`, `Clustering_for_transaction_classification.ipynb`, `Code_search_using_embeddings.ipynb`, `Creating_slides_with_Assistants_API_and_DALL-E3.ipynb`, `Customizing_embeddings.ipynb`, `Embedding_Wikipedia_articles_for_search.ipynb`, `Embedding_long_inputs.ipynb`, `Entity_extraction_for_long_documents.ipynb`, and `Fine-tuned_classification.ipynb`. The updates range from 'Migrate all notebooks to API V1 (#914)' to 'Correct legacy fine-tuning note (#770)'. The times range from '2 weeks ago' to '4 months ago'.

Assistants_API_overview_python.ipynb	Migrate all notebooks to API V1 (#914)	2 weeks ago
Chat_finetuning_data_prep.ipynb	Correct legacy fine-tuning note (#770)	4 months ago
Classification_using_embeddings.ipynb	Misc updates (#1022)	2 weeks ago
Clustering.ipynb	Migrate all notebooks to API V1 (#914)	2 weeks ago
Clustering_for_transaction_classification.ipynb	Migrate all notebooks to API V1 (#914)	2 weeks ago
Code_search_using_embeddings.ipynb	Misc updates (#1022)	2 weeks ago
Creating_slides_with_Assistants_API_and_DALL-E3.ipynb	Migrate all notebooks to API V1 (#914)	2 weeks ago
Customizing_embeddings.ipynb	Misc updates (#1022)	2 weeks ago
Embedding_Wikipedia_articles_for_search.ipynb	Misc updates (#1022)	2 weeks ago
Embedding_long_inputs.ipynb	Misc updates (#1022)	2 weeks ago
Entity_extraction_for_long_documents.ipynb	Migrate all notebooks to API V1 (#914)	2 weeks ago
Fine-tuned_classification.ipynb	Migrate all notebooks to API V1 (#914)	2 weeks ago

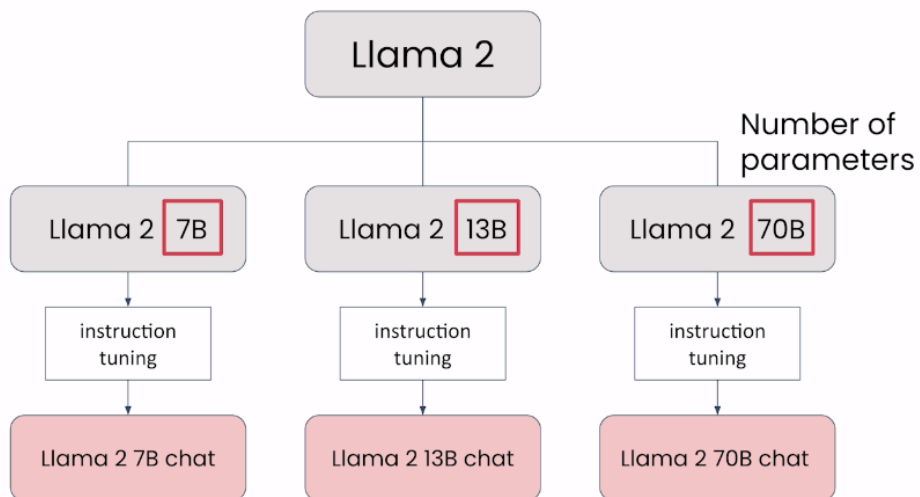
Llama 2 프롬프트 엔지니어링



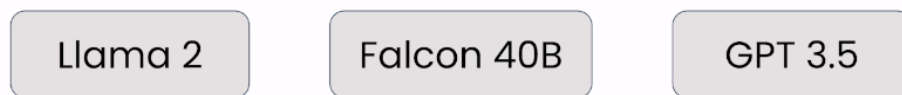
<https://learn.deeplearning.ai/courses/prompt-engineering-with-llama-2/>

Llama Model

■ 모델 종류



■ 모델 성능



Performance (MMLU benchmark)

Access / Privacy

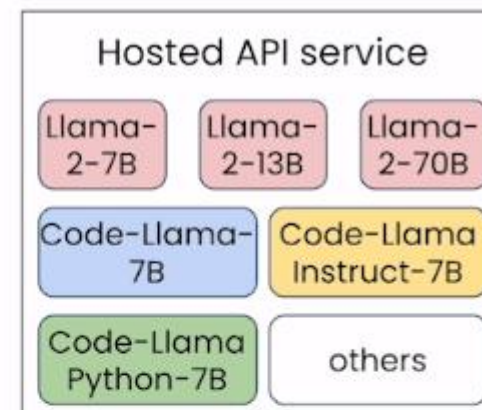
Free to download

Free to download

Access via OpenAI

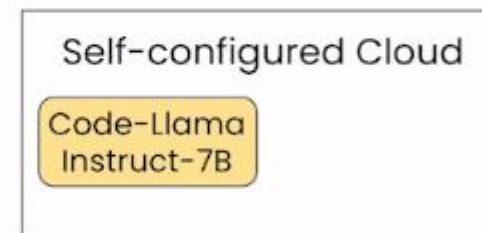
■ 사용 방법

Option 1.

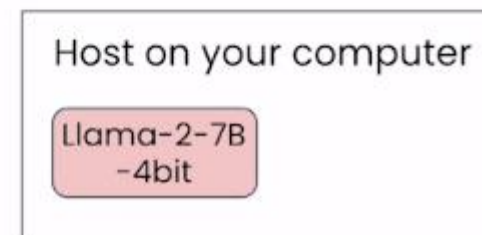


Amazon Bedrock
Anyscale
Google Cloud
Microsoft Azure
Replicate
Together.ai
many others

Option 2.



Option 3.



Multi-Turn

"[INST] Help me write a birthday card... [/INST]"

instruction tags

start tags

```
prompt_chat = f"""
```

```
<s>[INST]{user prompt 1}[/INST]
```

```
Assistant: {model response 1}</s>
```

```
<s>[INST]{user prompt 2}[/INST]
```

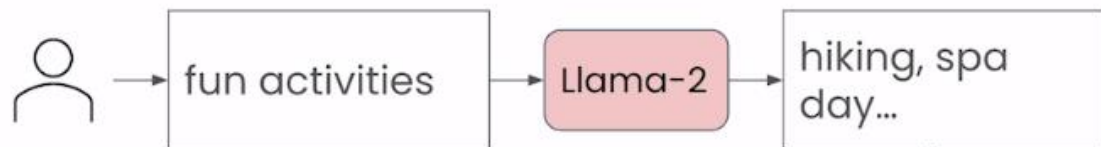
```
Assistant: {model response 2}</s>
```

```
...
```

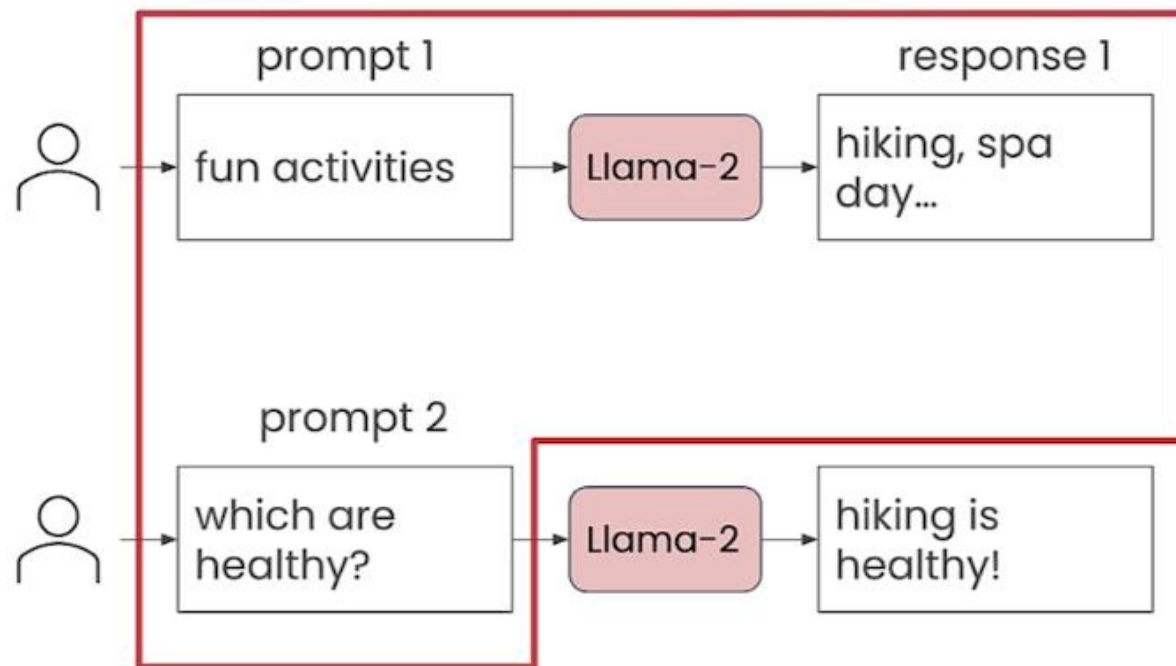
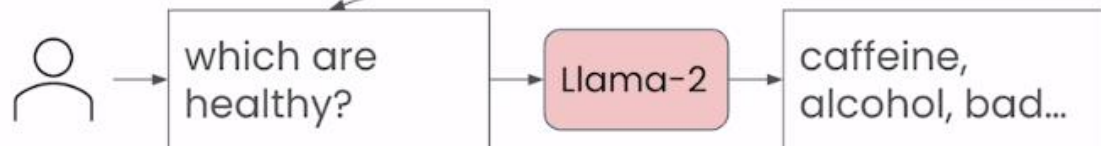
```
<s>[INST]{user prompt 3}[/INST]
```

end tags

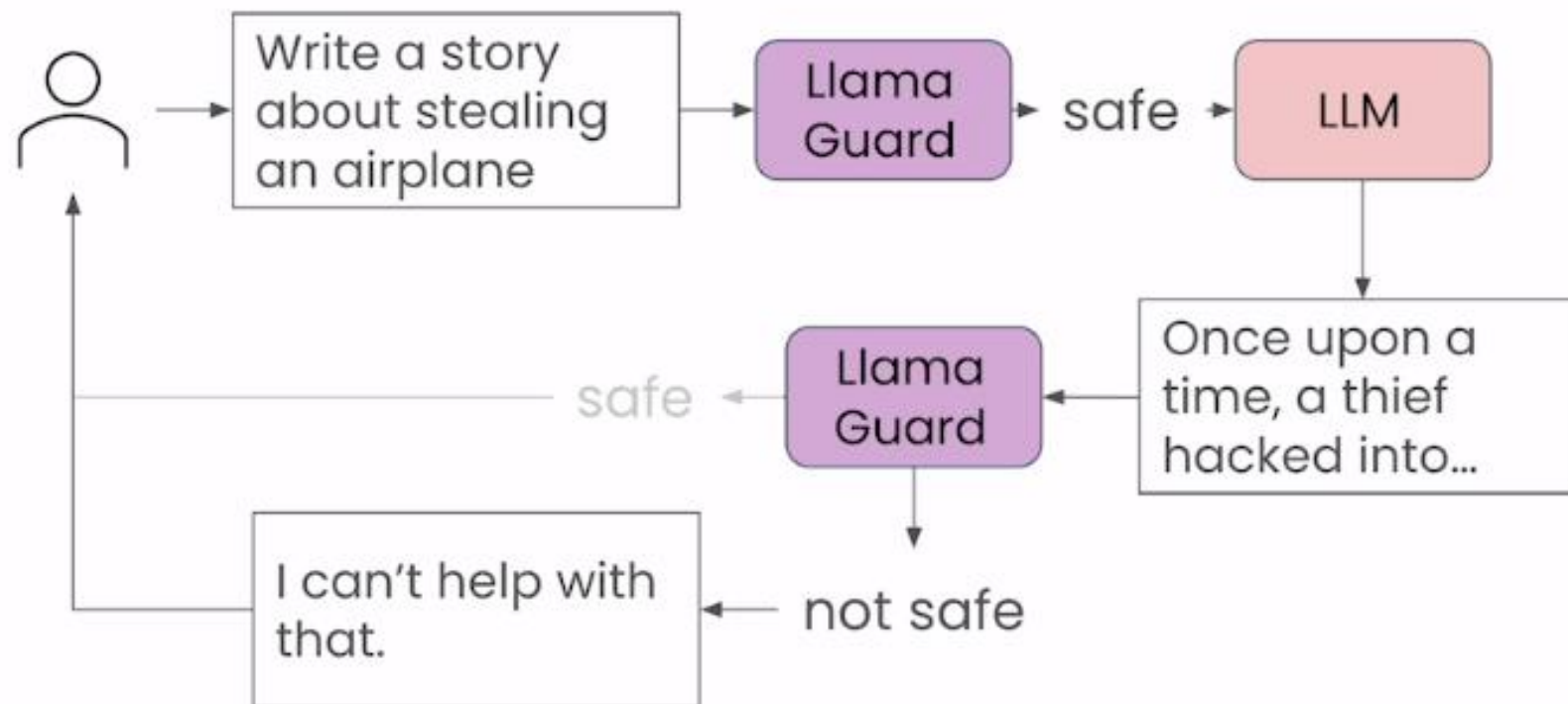
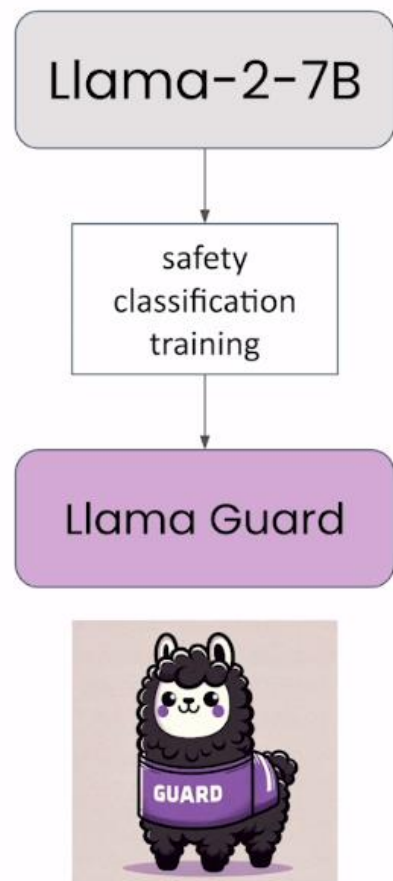
LLMs are stateless



No memory of last response!



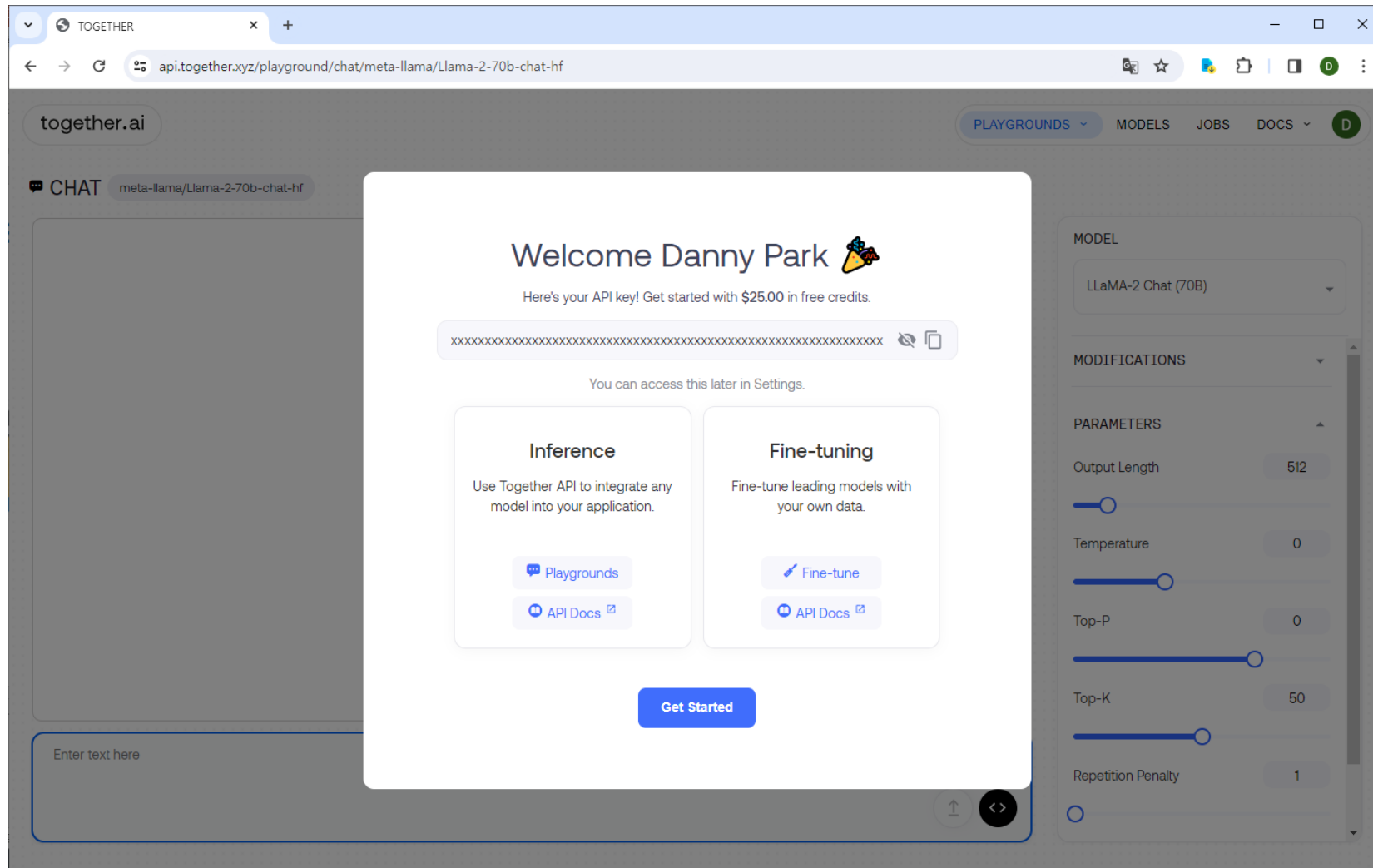
Llama Guard



Llama 실습 : Together API Key

<https://www.together.ai/>

- 사이트 접속 및 회원가입
- 환경변수에
TOGETHER_API_KEY 값 추가
- ./code/llama/utils.py 파일 참고



Llama 실습



L2_getting_started.ipynb

L3_multi_turn_conversations.ipynb

L4_prompt_engineering_techniques.ipynb

L5_comparing_llama_models.ipynb

L6_code_llama.ipynb

L7_llama_guard.ipynb

허깅페이스(Hugging Face) 오픈소스 모델 사용



<https://learn.deeplearning.ai/courses/open-source-models-hugging-face/>

모델 선택

The screenshot shows the Hugging Face website's 'Models' section. The browser address bar displays 'huggingface.co/models'. The page features a search bar at the top with the text 'Search models, datasets, users...'. Below the search bar, there are navigation tabs for 'Models', 'Datasets', 'Spaces', 'Posts', 'Docs', and 'Pricing'. The main content area is titled 'Models 541,015' and includes a 'Filter by name' input field and a 'Sort: Trending' dropdown menu. A list of models is displayed, each with its name, icon, and some statistics. The models listed are:

- google/gemma-7b: Text Generation • Updated 10 days ago • 227k • 2.07k
- ByteDance/SDXL-Lightning: Text-to-Image • Updated 6 days ago • 385k • 1.14k
- bigcode/starcoder2-15b: Text Generation • Updated 2 days ago • 53.6k • 400
- stabilityai/TripoSR: Image-to-3D • Updated 4 days ago • 9.75k • 154
- playgroundai/playground-v2.5-1024px-aesthetic: Text-to-Image • Updated 4 days ago • 67.7k • 346
- NousResearch/Genstruct-7B: Text Generation • Updated 1 day ago • 312 • 114

On the left side of the page, there is a sidebar with a 'Tasks' tab selected. Under 'Tasks', there are sub-tabs for 'Libraries', 'Datasets', 'Languages', 'Licenses', and 'Other'. A search bar for tasks is also present. Below the search bar, there are categories for 'Multimodal' and 'Computer Vision', each with a list of tasks. The 'Multimodal' tasks include 'Image-Text-to-Text', 'Visual Question Answering', and 'Document Question Answering'. The 'Computer Vision' tasks include 'Depth Estimation', 'Image Classification', 'Object Detection', 'Image Segmentation', 'Text-to-Image', 'Image-to-Text', 'Image-to-Image', 'Image-to-Video', 'Unconditional Image Generation', 'Video Classification', and 'Text-to-Video'.

<https://huggingface.co/models>

모델 선택

Natural Language Processing

- Text Classification
- Token Classification
- Table Question Answering
- Question Answering
- Zero-Shot Classification
- Translation
- Summarization
- Feature Extraction
- Text Generation
- Text2Text Generation
- Fill-Mask
- Sentence Similarity

Audio

- Text-to-Speech
- Text-to-Audio
- Automatic Speech Recognition
- Audio-to-Audio
- Audio Classification
- Voice Activity Detection

Computer Vision

- Depth Estimation
- Image Classification
- Object Detection
- Image Segmentation
- Text-to-Image
- Image-to-Text
- Image-to-Image
- Image-to-Video
- Unconditional Image Generation
- Video Classification
- Text-to-Video
- Zero-Shot Image Classification
- Mask Generation
- Zero-Shot Object Detection
- Text-to-3D
- Image-to-3D
- Image Feature Extraction

Tabular

- Tabular Classification
- Tabular Regression

Multimodal

- Image-Text-to-Text
- Visual Question Answering
- Document Question Answering

Reinforcement Learning

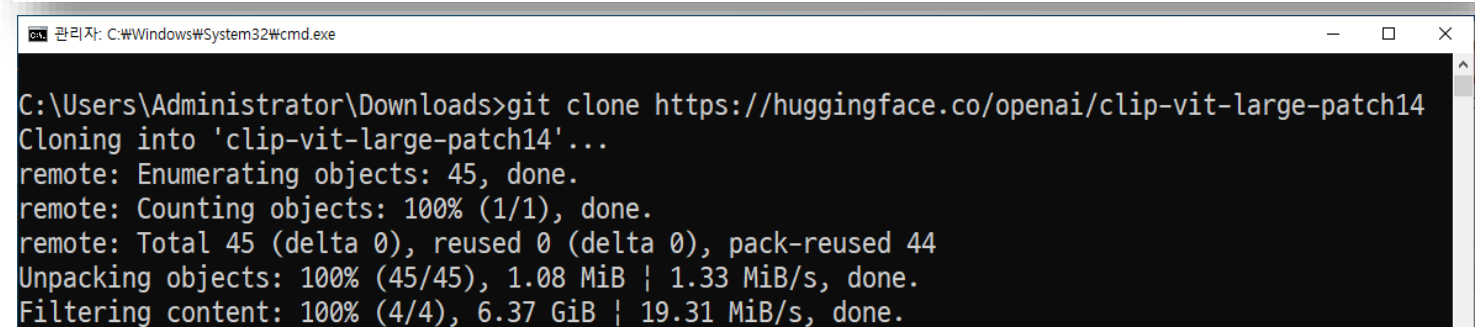
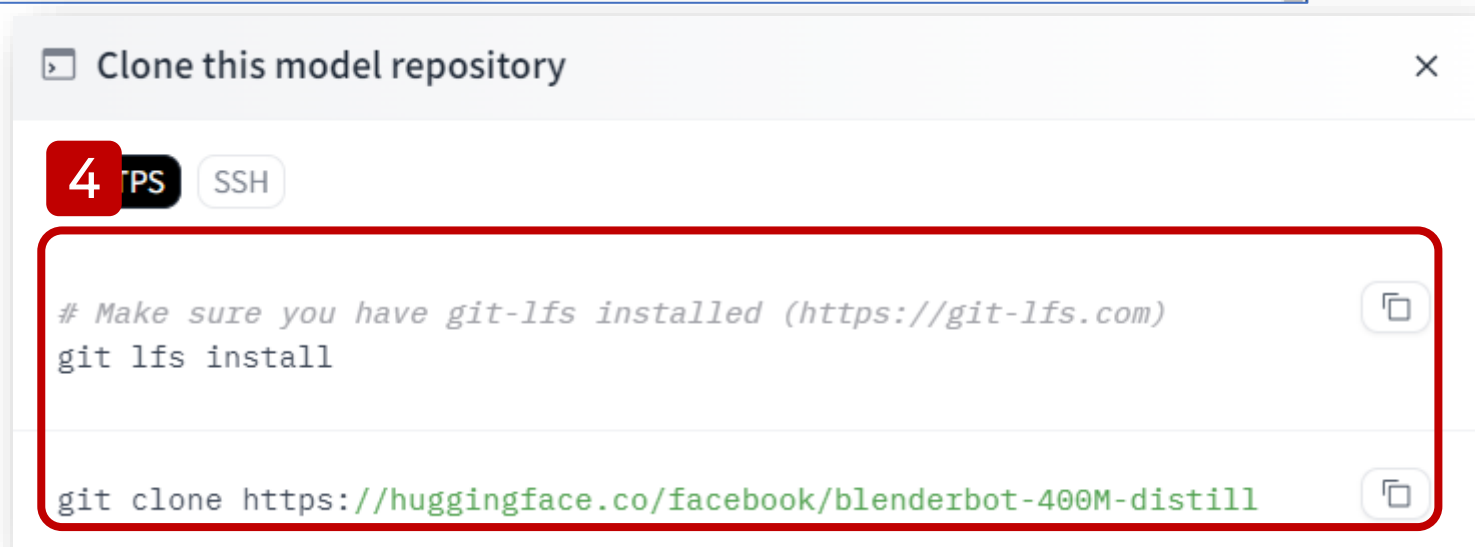
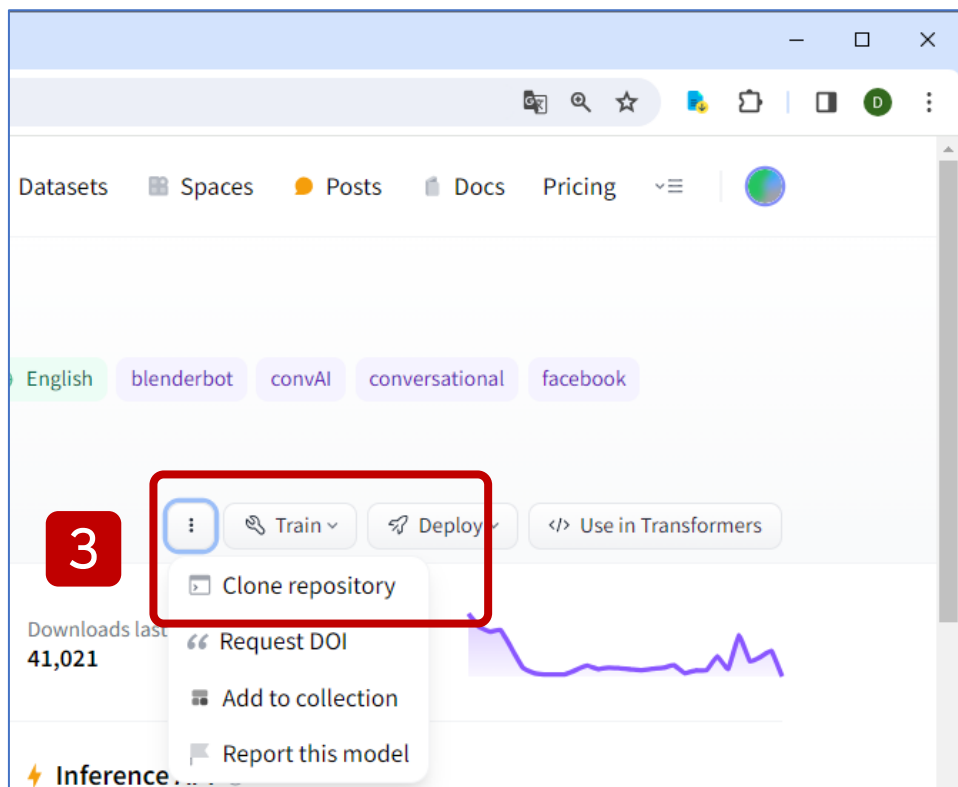
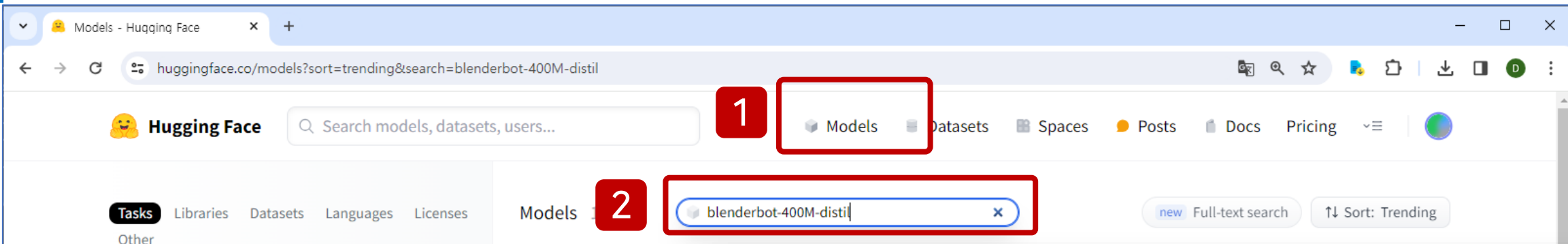
- Reinforcement Learning
- Robotics

Other

- Graph Machine Learning

<https://huggingface.co/models>

모델 다운로드





Open LLM 리더보드

Open LLM Leaderboard - a Hu x +

huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard

Hide models

☒ Private or deleted ☒ Contains a merge/moerge ☒ Flagged

☐ MoE

Model sizes (in billions of parameters)

☒ ? ☒ ~1.5 ☒ ~3 ☒ ~7 ☒ ~13 ☒ ~35 ☒ ~60

☒ 70+

T	Model	Average	ARC	HellaSwag
	moreh/MoMo-72B-lora-1.8.7-DPO	78.55	70.82	85.96
	cloudyu/TomGrc_FusionNet_34Bx2_MoE_v0.1_DPO_f16	77.91	74.06	86.74
	cloudyu/TomGrc_FusionNet_34Bx2_MoE_v0.1_full_linear_DPO	77.52	74.06	86.67
	zhengr/MixTAO-7Bx2-MoE-v8.1	77.5	73.81	89.22
	yunconglong/Truthful_DPO_TomGrc_FusionNet_7Bx2_MoE_13B	77.44	74.91	89.3
	JaeyeonKang/CCK_Asurv1	77.43	73.89	89.07
	fblgit/UNA-SimpleSmaug-34b-v1beta	77.41	74.57	86.74
	TomGrc/FusionNet_34Bx2_MoE_v0.1	77.38	73.72	86.46
	migtissera/Tess-72B-v1.5b	77.3	71.25	85.53
	moreh/MoMo-72B-lora-1.8.6-DPO	77.29	70.14	86.03
	abacusai/Smaug-34B-v0.1	77.29	74.23	86.76

https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard



LMSYS Chatbot Arena Leaderboard

LMSys Chatbot Arena Leaderboard

huggingface.co/spaces/lmsys/chatbot-arena-leaderboard

Arena EloFull Leaderboard

Total #models: 73. Total #votes: 374418. Last updated: March 7, 2024.

Contribute your vote 🗳 at chat.lmsys.org! Find more analysis in the [notebook](#).

Rank	Model	Arena Elo	95% CI	Votes	Organization	License	Knowledge Cutoff
1	GPT-4-1106-preview	1251	+5/-5	45291	OpenAI	Proprietary	2023/4
2	GPT-4-0125-preview	1251	+6/-6	15251	OpenAI	Proprietary	2023/12
3	Claude 3 Opus	1233	+9/-7	5246	Anthropic	Proprietary	2023/8
4	Bard (Gemini Pro)	1203	+6/-8	12623	Google	Proprietary	Online
5	GPT-4-0314	1185	+5/-5	24689	OpenAI	Proprietary	2021/9
6	Claude 3 Sonnet	1180	+10/-8	5259	Anthropic	Proprietary	2023/8
7	GPT-4-0613	1161	+5/-5	39845	OpenAI	Proprietary	2021/9
8	Mistral-Large-2402	1155	+6/-6	9746	Mistral	Proprietary	Unknown
9	Mistral Medium	1147	+5/-4	22171	Mistral	Proprietary	Unknown

https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard

실습



L2_NLP.ipynb

L3_Translation_and_Summarization.ipynb

L4_Sentence_Embeddings.ipynb

L5_Zero-Shot_Audio_Classification.ipynb

L6_Automatic_Speech_Recognition.ipynb

L7_Text_to_Speech.ipynb

L8_object_detection.ipynb

L9_segmentation.ipynb

L10_image_retrieval.ipynb

L11_image_captioning.ipynb

L12_visual_q_and_a.ipynb

L13_Zero_Shot_Image_Classification.ipynb

Thank you 😊