Importing Necessary Libraries

This section imports all the necessary libraries required for the rest of the notebook.

```
import IPython
import sys

def clean_notebook():
    IPython.display.clear_output(wait=True)
    print("Notebook cleaned.")

# Run the installation commands

!pip install openai
!pip install gradio
!pip install tiktoken
!pip install faiss-gpu
!pip install  datasets
# Clean up the notebook
clean_notebook()
```

```
Notebook cleaned.
```

```

import openai
openai.api_key = "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"
```

Downloading Dataset

This section downloads the dataset using the wget command.

```
 !wget -nc https://raw.githubusercontent.com/Tuchsanai/AIMaster-seagate-training-2024/6a91fb201929394a4c640ed32a29b69136c1a6f7/Dataset/bbc_text_cls.csv

```

```
--2024-06-04 11:42:14--  https://raw.githubusercontent.com/Tuchsanai/AIMaster-seagate-training-2024/6a91fb201929394a4c640ed32a29b69136c1a6f7/Dataset/bbc_text_cls.csv
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.111.133, 185.199.109.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.111.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5085081 (4.8M) [text/plain]
Saving to: ‘bbc_text_cls.csv’

bbc_text_cls.csv    100%[===================>]   4.85M  --.-KB/s    in 0.04s

2024-06-04 11:42:15 (127 MB/s) - ‘bbc_text_cls.csv’ saved [5085081/5085081]
```

Importing Necessary Libraries

```
# text-embedding-3-small    $0.02 / 1M tokens
# text-embedding-3-large    $0.13 / 1M tokens

import tiktoken
import pandas as pd
import numpy as np

encoding = tiktoken.encoding_for_model("gpt-3.5-turbo")

```

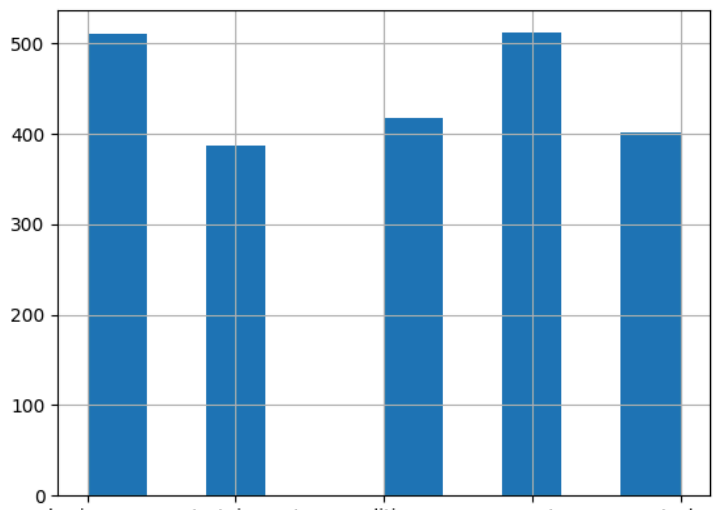
Read Data and Calcuate Tokens

```
def update_total(text):
    ints = encoding.encode(text)
    num_token = len(ints)
    return num_token


df = pd.read_csv('./bbc_text_cls.csv')
df['num_token']= df['text'].apply(update_total)

print("there are token = ", df['num_token'].sum())

df['labels'].hist()

df
```

```
there are token =  1068453
```

	text	labels	num_token
0	Ad sales boost Time Warner profit\n\nQuarterly...	business	564
1	Dollar gains on Greenspan speech\n\nThe dollar...	business	461
2	Yukos unit buyer faces loan claim\n\nThe owner...	business	344
3	High fuel prices hit BA's profits\n\nBritish A...	business	557
4	Pernod takeover talk lifts Domecq\n\nShares in...	business	391
...	...	...	...
2220	BT program to beat dialler scams\n\nBT is intr...	tech	503
2221	Spam e-mails tempt net shoppers\n\nComputer us...	tech	435
2222	Be careful how you code\n\nA new European dire...	tech	1205
2223	US cyber security chief resigns\n\nThe man mak...	tech	439
2224	Losing yourself in online gaming\n\nOnline rol...	tech	3540

2225 rows × 3 columns



Next steps: View recommended plots

```
from IPython.display import Markdown, display

idx = 0

display(Markdown(df.loc[idx, "labels"]))
display(Markdown("==================================="))
display(Markdown(df.loc[idx, "text"]))
```

business

===================================

Ad sales boost Time Warner profit

Quarterly profits at US media giant TimeWarner jumped 76% to $1.13bn (£600m) for the three months to December, from 639m year-earlier.

The firm, which is now one of the biggest investors in Google, benefited from sales of high-speed internet connections and higher advert sales. TimeWarner said fourth quarter sales rose 2% to $11.1bn from 10.9bn. Its profits were buoyed by one-off gains which offset a profit dip at Warner Bros, and less users for AOL.

Time Warner said on Friday that it now owns 8% of search-engine Google. But its own internet business, AOL, had has mixed fortunes. It lost 464,000 subscribers in the fourth quarter profits were lower than in the preceding three quarters. However, the company said AOL's underlying profit before exceptional items rose 8% on the back of stronger internet advertising revenues. It hopes to increase subscribers by offering the online service free to TimeWarner internet customers and will try to sign up AOL's existing customers for high-speed broadband. TimeWarner also has to restate 2000 and 2003 results following a probe by the US Securities Exchange Commission (SEC), which is close to concluding.

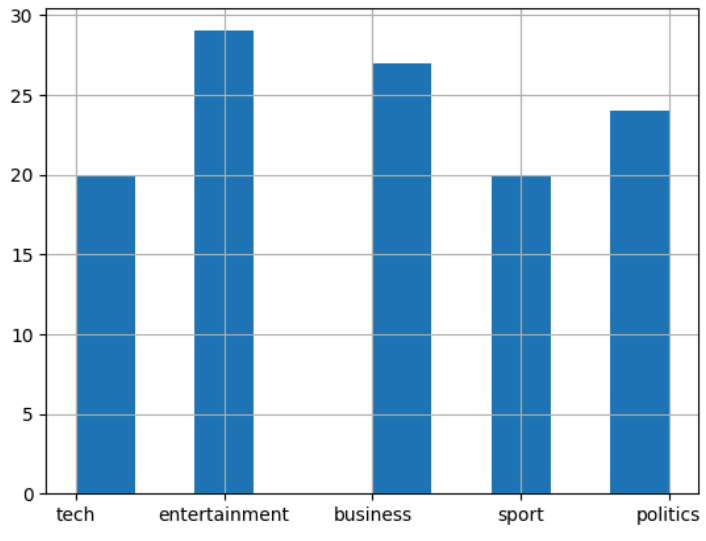
Time Warner's fourth quarter profits were slightly better than analysts' expectations. But its film division saw profits slump 27% to $284m, helped by box-office flops Alexander and Catwoman, a sharp contrast to year-earlier, when the third and final film in the Lord of the Rings trilogy boosted results. For the full-year, Time Warner posted a profit of 3.36bn, up 27% from its 2003 performance, while revenues grew 6.4% to $42.09bn. "Our financial performance was strong, meeting or exceeding all of our full-year objectives and greatly enhancing our flexibility," chairman and chief executive Richard Parsons said. For 2005, TimeWarner is projecting operating earnings growth of around 5%, and also expects higher revenue and wider profit margins.

TimeWarner is to restate its accounts as part of efforts to resolve an inquiry into AOL by US market regulators. It has already offered to pay $300m to settle charges, in a deal that is under review by the SEC. The company said it was unable to estimate the amount it needed to set aside for legal reserves, which previously stood at 500m. It intends to adjust the way it accounts for a deal with German music publisher Bertelsmann's purchase of a stake in AOL Europe, which it had reported as advertising revenue. It will now book the sale of its stake in AOL Europe as a loss on the value of that stake.

Reduce Size

```
df_small = df.sample(120).reset_index(drop=True)
print("there are token = ", df_small['num_token'].sum())
df_small['labels'].hist()
```

```
there are token =  60612
<Axes: >
```



Generate Embeddings

```
def get_embedding(text):
    text = text.replace("\n", "")
    emb = openai.embeddings.create(
        input=[text],
        model='text-embedding-3-small',
    )
    emb = np.array(emb.data[0].embedding).reshape(1, -1)
    # normalize
    emb = emb / np.linalg.norm(emb)

    return emb
```

```
df_small['embeddings'] = df_small['text'].apply(get_embedding)

df_small
```

	text	labels	num_token	embeddings
0	Souped-up wi-fi is on the horizon\n\nSuper hig...	tech	605	[[-0.0010582132391095988, 0.013093332560933032...
1	French honour for director Parker\n\nBritish f...	entertainment	278	[[0.008362293326998924, -0.011032002156981817,...
2	BMW drives record sales in Asia\n\nBMW has for...	business	357	[[0.014949961644249563, 0.0295736586371495, 0....
3	Williams battles to Aussie title\n\nSerena Wil...	sport	477	[[0.0213576147400522, 0.026941377913078896, 0....
4	Hewitt fights back to reach final\n\nLleyton H...	sport	553	[[0.003041838709496746, -0.003704473778702312,...
...	...	...	...	...
115	Gervais writing Simpsons episode\n\nThe Office...	entertainment	309	[[0.01721736346820815, 0.06436663606593675, -0...
116	De Niro completes box office coup\n\nRobert De...	entertainment	308	[[-0.047659075908246876, 0.033896140155352236,...
117	Venus stunned by Farina Elia\n\nVenus Williams...	sport	351	[[-0.0026242874084650486, 0.007989238660319439...
118	Cannabis hopes for drug firm\n\nA prescription...	business	383	[[0.018043435043778422, 0.012143673911248239, ...
119	Top Tories on Lib Dem 'hit list'\n\nThe Libera...	politics	361	[[-0.014959511111661289, 0.01318830499742569, 0...

120 rows × 4 columns

Next steps: View recommended plots

```
embedding_vector = np.concatenate(df_small['embeddings'].tolist(), dtype="float32")
embedding_vector.shape
```

```
(120, 1536)
```

Create and Populate the FAISS Index

We will use FAISS, a library for efficient similarity search and clustering of dense vectors, to create an index and add our image embeddings to it.

```
import faiss

# Create and populate the Faiss index
d = embedding_vector.shape[1]  # Dimension of the embeddings

index = faiss.IndexFlatL2(d)
index.reset()

print("Number of vectors in the index: ", index.ntotal)
print("number of dimension = ", d)
```

```
Number of vectors in the index:  0
number of dimension =  1536
```

```
# clear the index
index.reset()
# add the embedding vector to the index
index.add(embedding_vector)
print("Number of vectors in the index: ", index.ntotal)

```

```
Number of vectors in the index:  120
```

Perform a Search with a Query

```
# Perform a search
def search_text(query, k=5):
    query_embedding = get_embedding(query).astype("float32")
    # normalize the query embedding
    query_embedding = query_embedding / np.linalg.norm(query_embedding)
    distances, indices = index.search(query_embedding, k)

    return distances[0], indices[0]

```

```
query = """By what percentage did China's economy expand in 2004?"""
distances, top_k_indices = search_text(query, k=5)
```

```
top_k_indices
```

```
array([69, 50, 83, 24, 35])
```

```
distances
```

```
array([0.6846211, 1.2235043, 1.2428131, 1.2964716, 1.3067598],
      dtype=float32)
```

Display the Results

```
print("Query:", query)
print("====================================")

for i, idx in enumerate(top_k_indices):
    if idx < 0:
        continue
    print(f"Rank {i+1}:")
    print("class = ",df_small.loc[idx, "labels"])
    print("distance = ", distances[i])
    print(df_small.loc[idx, "text"])
    print("====================================")
```

```
Online gaming sales are predicted to top a billion US dollars next year, according to the UK-based journal Screen Digest. The video market is also seen as a big opportunity, although piracy levels
====================================
Rank 2:
class =  business
distance =  1.2590168
BMW drives record sales in Asia

BMW has forecast sales growth of at least 10% in Asia this year after registering record sales there in 2004.

The luxury carmaker saw strong sales of its three marques - BMW, Mini and Rolls-Royce - in Asia last year after the launch of three new models. The company, which is vying with Mercedes-Benz for th

BMW sold nearly 95,000 cars in Asia last year, up 2.6% on 2003.

BMW-brand sales rose 2.3% to 80,600 while sales of Mini models rose 3.6% to 14,800. There was also a significant increase in sales of Rolls-Royces on the continent. BMW sold more than 100 of the ic

China remains the main area of concern for BMW after sales there fell 16% last year. However, BMW is hopeful of a much better year in 2005 as its direct investment in China begins to pay dividends.
====================================
Rank 3:
class =  business
distance =  1.2760848
S Korea spending boost to economy

South Korea will boost state spending next year in an effort to create jobs and kick start its sputtering economy.

It has earmarked 100 trillion won ($96bn) for the first six months of 2005, 60% of its total annual budget. The government's main problems are "slumping consumption and a contraction in the constru

The government has set an economic growth rate target of 5% for next year and hinted that would be in danger unless it took action. "Internal and external economic conditions are likely to remain u

It blamed "continuing uncertainties such as fluctuating oil prices and foreign exchange rates and stagnant domestic demand that has shown few signs of a quick rebound". In 2004, growth will be betw

The problem facing South Korea is that many consumers are reeling from the effects of a credit bubble that only recently burst. Millions of South Koreans are defaulting on their credit card bills,

The government voiced concern about the effect of redundancies in the building trade. "Given the economic spill over and employment effect in the construction sector, a sharp downturn in the constr

As a result, South Korea will give private companies also will be given the chance to build schools, hospitals, houses and other public buildings. It also will look at real estate tax system. Other
====================================
Rank 4:
class =  business
distance =  1.2957528
S Korean consumers spending again

South Korea looks set to sustain its revival thanks to renewed private consumption, its central bank says.

The country's economy has suffered from an overhang of personal debt after its consumers' credit card spending spree. Card use fell sharply last year, but is now picking up again with a rise in spe
====================================
Rank 5:
class =  business
distance =  1.2964206
Record year for Chilean copper

Chile's copper industry has registered record earnings of $14.2bn in 2004, the governmental Chilean Copper Commission (Cochilco) has reported.

Strong demand from China's fast-growing economy and high prices have fuelled production, said Cochilco vice president Patricio Cartagena. He added that the boom has allowed the government to collec

"With these investments, clearly we are going to continue being the principle actor in the mining of copper. It's a consolidation of the industry with new projects and expansions that will support
====================================
```

Apply to LLM with LLAMA3

```
!pip install transformers accelerate datasets bitsandbytes

# Clean up the notebook
clean_notebook()
```

```
Requirement already satisfied: torch>=1.10.0 in /usr/local/lib/python3.10/dist-packages (from accelerate) (2.3.0+cu121)
Requirement already satisfied: pyarrow>=12.0.0 in /usr/local/lib/python3.10/dist-packages (from datasets) (14.0.2)
Requirement already satisfied: pyarrow-hotfix in /usr/local/lib/python3.10/dist-packages (from datasets) (0.6)
Requirement already satisfied: dill<0.3.9,>=0.3.0 in /usr/local/lib/python3.10/dist-packages (from datasets) (0.3.8)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (from datasets) (2.0.3)
Requirement already satisfied: xxhash in /usr/local/lib/python3.10/dist-packages (from datasets) (3.4.1)
Requirement already satisfied: multiprocess in /usr/local/lib/python3.10/dist-packages (from datasets) (0.70.16)
Requirement already satisfied: fsspec[http]<=2024.3.1,>=2023.1.0 in /usr/local/lib/python3.10/dist-packages (from datasets) (2023.6.0)
Requirement already satisfied: aiohttp in /usr/local/lib/python3.10/dist-packages (from datasets) (3.9.5)
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.3.1)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (23.2.0)
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.4.1)
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (6.0.5)
Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.9.4)
Requirement already satisfied: async-timeout<5.0,>=4.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (4.0.3)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1.0,>=0.23.0->transformers) (4.12.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (2024.2.2)
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (1.12.1)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (3.3)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (3.1.4)
Collecting nvidia-cuda-nvrtc-cu12==12.1.105 (from torch>=1.10.0->accelerate)
  Using cached nvidia_cuda_nvrtc_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (23.7 MB)
Collecting nvidia-cuda-runtime-cu12==12.1.105 (from torch>=1.10.0->accelerate)
  Using cached nvidia_cuda_runtime_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (823 kB)
Collecting nvidia-cuda-cupti-cu12==12.1.105 (from torch>=1.10.0->accelerate)
  Using cached nvidia_cuda_cupti_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (14.1 MB)
Collecting nvidia-cudnn-cu12==8.9.2.26 (from torch>=1.10.0->accelerate)
  Using cached nvidia_cudnn_cu12-8.9.2.26-py3-none-manylinux1_x86_64.whl (731.7 MB)
```

```
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets) (2023.4)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets) (2024.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas->datasets) (1.16.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->torch>=1.10.0->accelerate) (2.1.5)
Requirement already satisfied: mpmath<1.4.0,>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from sympy->torch>=1.10.0->accelerate) (1.3.0)
Installing collected packages: nvidia-nvtx-cu12, nvidia-nvjitlink-cu12, nvidia-nccl-cu12, nvidia-curand-cu12, nvidia-cufft-cu12, nvidia-cuda-runtime-cu12, nvidia-cuda-nvrtc-cu12, nvidia-cuda-cupti-
Successfully installed accelerate-0.30.1 bitsandbytes-0.43.1 nvidia-cublas-cu12-12.1.3.1 nvidia-cuda-cupti-cu12-12.1.105 nvidia-cuda-nvrtc-cu12-12.1.105 nvidia-cuda-runtime-cu12-12.1.105 nvidia-cud
```

```
import os
os.environ['HF_TOKEN'] ="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"
hf_token               = os.environ['HF_TOKEN']

################################################################################
# bitsandbytes parameters
################################################################################

# Activate 4-bit precision base model loading
use_4bit = True

# Compute dtype for 4-bit base models
bnb_4bit_compute_dtype = "float16"

# Quantization type (fp4 or nf4)
bnb_4bit_quant_type = "nf4"

# Activate nested quantization for 4-bit base models (double quantization)
use_nested_quant = False



# Load the entire model on the GPU 0
device_map = {"": 0}

```

```
from transformers import AutoModelForCausalLM,AutoTokenizer,BitsAndBytesConfig
import torch

# Load tokenizer and model with QLoRA configuration
compute_dtype = getattr(torch, bnb_4bit_compute_dtype)

bnb_config = BitsAndBytesConfig(
    load_in_4bit=use_4bit,
    bnb_4bit_quant_type=bnb_4bit_quant_type,
    bnb_4bit_compute_dtype=compute_dtype,
    bnb_4bit_use_double_quant=use_nested_quant,
)

model_id = "meta-llama/Meta-Llama-3-8B-Instruct"

model     = AutoModelForCausalLM.from_pretrained(model_id, quantization_config=bnb_config,device_map="auto",token=hf_token)
tokenizer = AutoTokenizer.from_pretrained(model_id)

```

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:89: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secret in yo
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
config.json: 100%                                                654/654 [00:00<00:00, 38.7kB/s]
model.safetensors.index.json: 100%                                  23.9k/23.9k [00:00<00:00, 1.45MB/s]
Downloading shards: 100%                                          4/4 [01:38<00:00, 20.86s/it]
model-00001-of-00004.safetensors: 100%                              4.98G/4.98G [00:29<00:00, 99.8MB/s]
model-00002-of-00004.safetensors: 100%                              5.00G/5.00G [00:34<00:00, 209MB/s]
model-00003-of-00004.safetensors: 100%                              4.92G/4.92G [00:27<00:00, 157MB/s]
model-00004-of-00004.safetensors: 100%                              1.17G/1.17G [00:05<00:00, 228MB/s]
Loading checkpoint shards: 100%                                   4/4 [01:14<00:00, 16.17s/it]
generation_config.json: 100%                                      187/187 [00:00<00:00, 11.5kB/s]
tokenizer_config.json: 100%                                       51.0k/51.0k [00:00<00:00, 2.96MB/s]
tokenizer.json: 100%                                            9.09M/9.09M [00:00<00:00, 13.9MB/s]
special_tokens_map.json: 100%                                     73.0/73.0 [00:00<00:00, 4.34kB/s]
Special tokens have been added in the vocabulary, make sure the associated word embeddings are fine-tuned or trained.
```

```
def generate_response(prompt, max_tokens=512, temperature=0.6, top_p=0.9):

    messages = [
        {"role": "system", "content": "You are an economist who do analysis and reserch about world economy."},
        {"role": "user", "content": prompt},
    ]

    input_ids = tokenizer.apply_chat_template(messages,add_generation_prompt=True,return_tensors="pt").to(model.device)


    terminators = [
        tokenizer.eos_token_id,
        tokenizer.convert_tokens_to_ids("<|eot_id|>")
    ]

    outputs = model.generate(
        input_ids,
        max_new_tokens=max_tokens,
        pad_token_id=model.config.eos_token_id,
        eos_token_id=terminators,
        do_sample=True,
        temperature=temperature,
        top_p=top_p,
    )

    response = outputs[0][input_ids.shape[-1]:]
    return tokenizer.decode(response, skip_special_tokens=True)

```

```
def qa(question, context):
    prompt = f"""Please answer the question given the provided \
    context.

    Question:

    ```
    {question}
    ```
```

```
    Context:

    ```
    {context}
    ```
    """

    result_txt = generate_response(prompt)
    return result_txt
```