```python
"""


3 - Write a Python script using the Azure SDK that uploads a file
to an Azure Blob Storage container. Check if the container exists,
Create it if it does not exists.
"""

import os
from typing import Tuple
from azure.storage.blob import BlobServiceClient

def get_account_credentials()-> Tuple[str, str]:
    """
    Return the storage account name
    and the storage account url.
    Both are used by the BlobServiceClient class interact with azure datalake and
    among other thing authenticate, ...

    Documentation:
    --------------
        https://learn.microsoft.com/en-us/python/api/azure-storage-blob/azure.storage.blob.blobserviceclient?view=azure-python

    Return:
    ------
        STORAGE_ACCOUNT_KEY: The storage account key
        ACCOUNT_URL: the storage account url
    """
    try:
        STORAGE_ACCOUNT_NAME = os.environ["STORAGE_ACCOUNT_NAME"]
        STORAGE_ACCOUNT_KEY = os.environ["STORAGE_ACCOUNT_KEY"]
        ACCOUNT_URL = f"https://{STORAGE_ACCOUNT_NAME}.blob.core.windows.net"
        return STORAGE_ACCOUNT_KEY, ACCOUNT_URL
    except KeyError:
        print("Missing account key or/and account name")
        return None, None


def upload_file_to_data_lake_container(file_path: str, container_name: str)->None:
    """
    Uploads a file to an azure datalake container.

    Arguments:
    ----------
        file_path: the path of the file to upload
        container_name: the container to upload the file to

    Returns:
    --------
        None
    """
    account_key, account_url = get_account_credentials()
    if account_key is None or account_url is None:
        raise Exception("Account Key or account name is invalide")
    if not os.path.exists(file_path):
        raise Exception("File to upload does not exist")
    try:
        blob_service_client = BlobServiceClient(account_url=account_url, credential=account_key)
        container_client = blob_service_client.get_container_client(container_name)
        if not container_client.exists():
            container_client = blob_service_client.create_container(container_name)
            print(f"Container does not exists. Created container {container_name}")
            with open(file_path, "rb") as file_object:
                container_client.upload_blob(name=file_path, data=file_object)
            print(f"Uploading file  {file_path} to  container {container_name}")
        else:
            print(f"Container exists. Uploading file {file_path} to container {container_name}")
            with open(file_path, "rb") as file_object:
                container_client.upload_blob(name=file_path, data=file_object)
    except Exception as e:
        print(e)



if __name__ =="__main__":
    account_name, account_key = get_account_credentials()
    container_name = "yapi-donatien-achou"
    files = [
        "yapi-achou-category-aggregate-tourism-dataset.csv",
        "yapi-achou-country-aggregate-tourism_dataset.csv"
    ]

    for file in files:
```

```
        upload_file_to_data_lake_container(file, container_name)
```