

تمرین 1) از فایل 13، برنامه‌های زیر را پیاده‌سازی کنید:

- برنامه‌های صفحه 11

- برنامه صفحه 14

- برنامه‌های صفحات 16، 17، و 18

تمرین 2)

چهار فولدر در نظر بگیرید: هنری، سیاسی، ورزشی، علمی. در واقع هر کدام از اینها یک دسته‌بندی است.

در هر فولدر 10 فایل متنی وجود داشته باشد. (یعنی هر 10 متن متعلق به یک دسته‌بندی است.)

در هر فولدر، اسم فایل‌ها 1 تا 10 است.

ابتدا یک دیتافریم بسازید که دارای این ستون‌ها باشد: اندیس، متن، نام دسته‌بندی متن، شماره دسته‌بندی متن (معادل هر دسته بندی یک عدد از 1 تا 4 مشخص کنید). سپس برنامه‌ای بنویسید که این 40 فایل از 40 فولدر را بخواند و در این دیتافریم بریزد.

برای حل این مسأله دو رویکرد را می‌توانید در نظر بگیرید:

الف) اینکه از قبل بدانیم تعداد فولدرها 4 و نام فولدرها 1 تا 4 است یا نام آنها را در یک لیست در کد داشته باشیم و همچنین بدانیم نام فایل‌ها در همه فولدرها 1 تا 10 است. سپس در دو حلقه for مسیر و نام هر فایل را به صورت یک رشته متنی ساخته و محتوای آنها را بخوانیم.

ب) استفاده از `os.listdir`:

این متد نام تمام فایل‌ها و دایرکتوری‌های داخل یک دایرکتوری داده شده را در قالب یک لیست برمی‌گرداند.

مثال) در این مثال، نام تمام فایل‌های و دایرکتوری‌های موجود در مسیر داده شده چاپ می‌شود

```
import os, sys

# Open a file
path = "/var/www/html/"
dirs = os.listdir( path )

# This would print all the files and directories
for file in dirs:
    print(file)
```

مثال) در این مثال، لیست شامل نام تمام فایل های و دایرکتوری های موجود در مسیر جاری چاپ می شود

```
# importing os module
import os

# Get the path of current working directory
path = os.getcwd()

# Get the list of all files and directories
dir_list = os.listdir(path)

print("Files and directories in '", path, "' :")
# print the list
print(dir_list)
```

به این ترتیب لازم نیست که نام دایرکتوری ها و فایل های موجود در آنها را از قبل بدانیم و می توانیم با به دست آوردن نام آنها متن شامل نام و مسیر آنها را ساخته و فایل را بخوانیم و پردازش کنیم.

تمرین 3)

پیاده سازی کلاس زیر را کامل کنید. سپس از آن شیء بسازید و متدهای آن را اجرا کنید.

```
import pandas as pd
class MyDataFrame:
    def __init__(self, data, columns):
        self.df = pd.DataFrame(data, columns=columns)

    def add_row(self, row_dic):
        self.df.loc[len(self.df.index)] = row_dic

    def remove_row(self, row_dic):
        ??

    def filter_by_column(self, column_name, value):
        return self.df[self.df[column_name] == value]

    def to_dict(self):
        return self.df.to_dict(orient='records')

    def save_csv(self, file_path):
        self.df.to_csv(file_path, index=False)
```

ورودی متدی که باید پیاده‌سازی کنید، شماره ردیف است.

ایجاد df از این کلاس به شکل زیر خواهد بود:

```
data = [
    {"Name": "John", "Age": 25, "Gender": "Male"},
    {"Name": "Jane", "Age": 30, "Gender": "Female"},
    {"Name": "Bob", "Age": 40, "Gender": "Male"},
    {"Name": "Alice", "Age": 35, "Gender": "Female"}
]
columns = ["Name", "Age", "Gender"]

my_data = MyDataFrame(data, columns)
print(my_data.df)
```

تمرین 4)

کلاس قبل را به این شکل تغییر دهید:

دو متد اضافه کنید، یک از آنها شیء دیتافریم را از دیکشنری ورودی بسازد (اتفاقی که فعلا در تابع init کلاس قبل رخ می‌دهد)، و دیگری شیء را از یک فایل CSV ورودی بسازد. تابع init کلاس قبل را به این شکل تغییر دهید که امکان ساخت دیتافریم را در هر یک از این دو حالت داشته باشد. یعنی اگر ورودی فایل باشد از فایل و اگر دیکشنری باشد، از دیکشنری ساخته شود.

تمرین 5) چندجمله‌ای‌ها را در نظر بگیرید.

می‌خواهیم کلاسی بنویسیم که اطلاعات چندجمله‌ای را نگهداری کند. به این صورت می‌توانیم اشیای چندجمله‌ای ایجاد کنیم که از این کلاس ایجاد شوند.

در تلاش اول: علاوه بر متدهای پایه که به ذهن شما می‌رسد (مانند تابع سازنده، یا نمایش چندجمله‌ای)، سه متد جمع و تفریق و ضرب را برای این کلاس پیاده‌سازی کنید. ورودی این متدها یک چندجمله‌ای دیگر است و با استفاده از `Special methods` overriding انجام شود.