PANDAS DATAFRAME EXERCISES[1]

1).- Viết chương trình tạo dataframe có dạng như hình sau:

	X	Y	Z
0	78	84	86
1	85	94	97
2	96	89	96
3	80	83	72
4	86	86	83

Minh họa kết quả câu a

a/- ***df1*** Dataframe: Với giá trị các phần từ được phát sinh ngẫu nhiên trong khoảng từ 60-100. Thực hiện bằng 3 cách:

(i). Khai báo 1 dict gồm 3 thành phần, dưa mydict làm đối số cho hàm DataFrame.

(ii). Khai báo dict ngay trong hàm tạo DataFrame.

(iii). Khai báo riêng 3 dict, sau đó kết hợp 3 dict vào 1 dict.

b/-`print(df1)` ***df2*** Dataframe:

(i) Ngoại trừ cột *name* là giá trị tự nhập, giá trị của các cột còn lại sẽ phát sinh ngẫu nhiên như sau:

- *score*: Giá trị phát sinh ngẫu nhiên và nằm trong khoảng từ *0* đến *20*. Trong đó, mỗi giá trị là 1 số lẻ và số lẻ này chỉ thuộc 1 trong 2 giá trị: *0* hoặc *5*.
- *attempts*: Giá trị phát sinh ngẫu nhiên và nằm trong khoảng từ *1* đến *3*.
- *qualify*: Giá trị phát sinh ngẫu nhiên và chỉ nhận 1 trong 2 giá *yes* hoặc *no*.

Sau khi tạo df2 hoàn tất, gán giá trị NaN cho 2 hàng 4 và 8 của cột score.

(ii) Sử dụng hàm *info* để hiển thị thông tin tóm tắt về *df2* DataFrame (tên cột, số lượng giá trị null, kiểu dữ liệu, …) như hình minh họa.

(iii) Sử dụng hàm describe để hiển thị các thống kê cơ bản (mean, min, max, …) của các thuộc tính kiểu số trong *df2* DataFrame.

(iv) Đếm số lượng giá trị *NaN* có trong *df2*.

(v) Đếm số lượng giá trị *NaN* có trong từng cột của *df2*.

(vi) Điền giá trị zero cho những giá trị NaN trong *df2*.

(vii) Viết lệnh để thêm mới 1 cột chuyển đổi chỉ mục của *df2*. Xuất kết quả ra màn hình để kiểm tra. Sau khi hoàn tất, sử dụng phương thức *DataFrameName.to_string()* để xuất df2 ra màn hình và không cho hiện cột index vừa tạo.

	name	score	attempts	qualify
a	Tý	12.5	1	yes
b	Sửu	9.0	3	no
c	Dần	16.5	2	yes
d	Mẹo	9.5	3	no
e	Thìn	NaN	2	no
f	Tỵ	20.0	3	yes
g	Ngọ	14.5	1	yes
h	Mùi	17.5	1	no
i	Thân	NaN	2	no
j	Dậu	19.0	1	yes

Minh họa kết quả câu b/-(i)

```
Thông tin tóm tắt về df DataFrame:
<class 'pandas.core.frame.DataFrame'>
Index: 10 entries, a to j
Data columns (total 4 columns):
 # Column Non-Null Count Dtype
--- ------ -------------- -----
 0 name 10 non-null object
 1 score 8 non-null float64
 2 attempts 10 non-null int64
 3 qualify 10 non-null object
dtypes: float64(1), int64(1), object(2)
memory usage: 400.0+ bytes
None
```

Minh họa kết quả câu b/-(ii)

	index	name	score	attempts	qualify
0	a	Tý	12.5	1	yes

index	name	score	attempts	qualify
a	Tý	12.5	1	yes

[1] https://www.w3resource.com/python-exercises/pandas/index-dataframe.php

1	b	Sửu	9.0	3	no
2	c	Dần	16.5	2	yes
3	d	Mẹo	9.5	3	no
4	e	Thìn	NaN	2	no
5	f	Tỵ	20.0	3	yes
6	g	Ngọ	14.5	1	yes
7	h	Mùi	17.5	1	no
8	i	Thân	NaN	2	no
9	j	Dậu	19.0	1	yes

Minh họa kết quả câu b/-(v) với chỉ mục vừa thêm

b	Sửu	9.0	3	no
c	Dần	16.5	2	yes
d	Mẹo	9.5	3	no
e	Thìn	NaN	2	no
f	Tỵ	20.0	3	yes
g	Ngọ	14.5	1	yes
h	Mùi	17.5	1	no
i	Thân	NaN	2	no
j	Dậu	19.0	1	yes

Minh họa kết quả câu b/-(v) với việc cho ẩn chỉ mục vừa thêm

2).- Dựa trên `df2` DataFrame, viết chương trình thực hiện các yêu cầu sau:

a/- Đếm số lượng hàng, số lượng cột.

b/- Lấy danh sách tên các cột của `df2` DataFrame đưa vào 1 `array`. Thực hiện lại tương tự nhưng đưa danh sách vào 1 `list`.

c/- Dựa trên `iterrows` của đối tượng DataFrame, duyệt qua từng hàng và in dữ liệu ra màn hình giá trị trên 3 cột: `index`, `name`, `score`.

d/- Hiển thị dữ liệu của hàng thứ 5.

e/- Hiển thị `3` hàng đầu tiên của dữ liệu.

f/- Chỉ cho hiển thị cột `name` và `score` .

	name	score	attempts	qualify
a	Anastasia	12.5	1	yes
b	Dima	9.0	3	no
c	Katherine	16.5	2	yes

Yêu cầu 2d

g/- Kết hợp câu d và câu e để có kết quả gồm 3 hàng và 2 cột.

h/- Chọn ra những hàng mà dữ liệu bị thiếu (NaN) trên cột `score`.

i/- Chỉ cho hiển thị các hàng có `attemps>2`.

j/- Chọn ra những hàng mà giá trị trên cột score nằm trong khoảng từ `15` đến `20`.

k/- Chọn ra những hàng mà giá trị trên cột score <12 hoặc >=15.

l/- Chọn các hàng có số `attempts` nhỏ hơn 2 và có `score` lớn hơn `15`.

m/- Chọn ra những hàng có `score>=18` (với đầy đủ các cột). Sau đó chỉ cho hiện nội dung của 2 cột `name` và `score`.

n/- Sử dụng thuộc tính `loc` để sửa giá trị của hàng có chỉ mục 4 là trên cột `score` thành `11.5`.

o/- Thay giá trị trên cột `qualify` từ `yes` thành `True` và từ `no` thành `False`.

p/- Thay tên 'Sửu' trong cột `name` thành tên 'Hợi'

q/- Sắp xếp dữ liệu giảm dần theo `name` và tăng dần theo `score`.

r/- Sử dụng thuộc tính `loc` để thêm 1 dòng mới với `index` là `k`, các dữ liệu khác trên dòng là tùy ý. Kiểm tra kết quả thực hiện.

s/- Xóa dòng vừa thêm ở câu trước. Như vậy dữ liệu lúc này tương tự như dữ liệu ban đầu.

t/- Sử dụng phương thức `concat` của pandas để thêm 1 dòng mới với `name`='Hợi', `score`=16.0, `attempts`=1, `qualify`=yes với index =0 vào `df2`.

u/- Đổi vị trí 2 cột `score` và `attempts` cho nhau.

v/- Đổi tên các cột theo thứ tự trái sang phải là: *Tên*, *Điểm*, *Số lần thi*, *Đủ điều kiện*. Kiểm tra kết quả thực hiện. Nếu đã đúng, thay lại tên cũ cho các cột của *df2*.

w/- Xóa cột *'qualify'* khỏi *df2*

3).- Nội dung *df2* DataFrame vào file *DataFrame2.csv*. Sau đó đọc dữ liệu từ file để đưa vào 1 DataFrame mới (*df2_new*) rồi in DataFrame này ra màn hình.

4).- Nối 2 data series vào 1 DataFrame

RETAIL TRANSACTIONS DATASET

5).- Thống kê số lượng giao dịch tại mỗi thành phố. Minh họa kết quả:

	City	Number of Trans
0	Atlanta	99066
1	Boston	100566
2	Chicago	100059
3	Dallas	100559
4	Houston	100050
5	Los Angeles	99879
6	Miami	99839
7	New York	100007
8	San Francisco	99808
9	Seattle	100167

6).- Chọn ra những giao dịch ở thành phố Houston. Đếm số lượng giao dịch vừa có

7).- Thiết lập giá trị cho các cell theo yêu cầu sau:

- Thay tên Tý cho hàng đầu tiên của cột `'Customer_Name'`
- Thay giá trị 22.22 cho hàng thứ 2 của cột `'Total_Cost'`
- Thay giá trị `'Specialty Store'` cho hàng thứ 6 của cột `'Store_Type'`

8).- Phân chia `df2` thành 2 DataFrame `df999` chiếm tỷ lệ `99.9%` dữ liệu của `df2` và `df001` chiếm tỷ lệ `0.1%` dữ liệu của `df2`. Xuất 2 DataFrame vừa có ra màn hình

9).- Xóa các hàng thứ 3 và hàng thứ 5 (không xóa hàng 4).

10).- Xóa từ hàng thứ 1 đến hàng thứ 3. Sau khi hoàn tất, thực hiện reset lại chỉ mục

11).- Sử dụng phương thức `set_option` để thiết lập trên pandas số dòng hiển thị (`display.max_rows`) tối đa là 5, số cột hiển thị tối đa (`display.max_columns`) là 5 và độ rộng hiển thị tối đa (`display.width`) là 50. Sau đó thực hiện:

- In DataFrame ra màn hình để xem kết quả.
- Có thể điều chỉnh nhiều lần các tham số này để thấy được các thay đổi.
- Cuối cùng, thiết lập lại cho các tham số này như sau:
  - `display.max_rows =500`
  - `display.max_columns =500`
  - `display.width =1000`