DateTime

In [5]:

```
import datetime as dt
```

In [7]:

```
print(dt.datetime(22,3,19))
```

```
0022-03-19 00:00:00
```

In [8]:

```
print(dt.timedelta(3-3-2022,3-3-2021))
```

```
-2023 days, 23:26:19
```

In [10]:

```
a = dt.datetime.now()
print(a)
print(a.year)
print(a.month)
print(a.date)
print(a.time)
print(a.day)
```

```
2022-04-28 20:15:57.982372
2022
4
<built-in method date of datetime.datetime object at 0x000001CDB61EA280>
<built-in method time of datetime.datetime object at 0x000001CDB61EA280>
28
```

In [11]:

```
b = dt.datetime(2022,4,25)
c = dt.datetime.now()
```

In [12]:

```
c
```

Out[12]:

```
datetime.datetime(2022, 4, 28, 20, 16, 0, 807024)
```

In [13]:

```
u = c-b
```

In [14]:

```
d = dt.date.today()
```

In [17]:

```
e = d.isoformat()
```

In [18]:

```
e
```

Out[18]:

```
'2022-04-28'
```

In [92]:

```
import timedelta

aa = dt.datetime(2022,3,23)
bb = dt.datetime(2022,1,12)
cc = dt.datetime(2021,2,16)
dd = dt.datetime(2021,4,25)
ee = dt.datetime(2021,7,14)
ff = dt.datetime(2021,8,12)
gg = dt.datetime(2022,2,14)
hh = dt.datetime(2022,2,25)
ii = dt.datetime(2021,11,16)

a = timedelta.Timedelta(dt.datetime.today()- aa)
b = timedelta.Timedelta(dt.datetime.today()- bb)
c = timedelta.Timedelta(dt.datetime.today()- cc)
d = timedelta.Timedelta(dt.datetime.today()- dd)
e = timedelta.Timedelta(dt.datetime.today()- ee)
f = timedelta.Timedelta(dt.datetime.today()- ff)
g = timedelta.Timedelta(dt.datetime.today()- gg)
h = timedelta.Timedelta(dt.datetime.today()- hh)
i = timedelta.Timedelta(dt.datetime.today()- ii)
l = []
products = {'a':a,'b':b,
            'c':c,'d':d,
            'e':e,'f':f,
            'g':g,'h':h,
            'i':i}
for i,j in products.items():
    l.append(j.total.days)
for i in l:
    if i < 100:
        print("Product is Near to the expiry date",i)
```

```
Product is Near to the expiry date 36
Product is Near to the expiry date 73
Product is Near to the expiry date 62
```

In [22]:

```
a = '22-01-23'
z = len(a)
b = dt.datetime.strptime(a,'%y-%m-%d')
c = dt.datetime.today()
e = c-b
print(e)
f = str(e)
l = []
print(f)
for i in f.split(" "):
    l.append(i)
s = int(l[0])
print(type(s))
print(s)
```

```
95 days, 20:17:26.470609
95 days, 20:17:26.470609
<class 'int'>
95
```

In [23]:

```
import pandas as pd
```

Reading Data From CSV

In [1]:

```
import pandas as pd
```

In [56]:

```
d = pd.DataFrame({'Name':['Harinath','Rohit','Rahul','Yashwanth','Snehith','Suresh'],"NickN
```

In [59]:

```
print(d['Name']=='Harinath')
```

```
0     True
1    False
2    False
3    False
4    False
5    False
Name: Name, dtype: bool
```

In [24]:

```
df = pd.read_csv(r"C:\Users\Administrator\OneDrive\Desktop\annual-enterprise-survey-2020-fi
```

In [25]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 37080 entries, 0 to 37079
Data columns (total 10 columns):
 #   Column                       Non-Null Count  Dtype 
---  ------                       --------------  ----- 
 0   Year                         37080 non-null  int64 
 1   Industry_aggregation_NZSIOC  37080 non-null  object
 2   Industry_code_NZSIOC         37080 non-null  object
 3   Industry_name_NZSIOC         37080 non-null  object
 4   Units                        37080 non-null  object
 5   Variable_code                37080 non-null  object
 6   Variable_name                37080 non-null  object
 7   Variable_category            37080 non-null  object
 8   Value                        37080 non-null  object
 9   Industry_code_ANZSIC06       37080 non-null  object
dtypes: int64(1), object(9)
memory usage: 2.8+ MB
```

In [26]:

```
df2 = df.groupby(['Variable_code','Industry_aggregation_NZSIOC'])
x = df2.first()
```

Saving the data into CSV Format

In [27]:

```
x.to_csv("Group.csv")
```

In [28]:

```
pwd()
```

Out[28]:

```
'C:\\Users\\Administrator'
```

In [29]:

```
ls
```

```
 Volume in drive C is Windows
 Volume Serial Number is 9C88-68EB

 Directory of C:\Users\Administrator

28-04-2022  20:15    <DIR>          .
28-11-2021  10:40    <DIR>          ..
27-12-2021  11:06    <DIR>          .android
09-04-2022  17:27    <DIR>          .aws
25-02-2022  12:35    <DIR>          .azcopy
27-03-2022  14:35             8,371 .bash_history
13-03-2022  19:49    <DIR>          .dotnet
30-01-2022  08:54               137 .gitconfig
07-01-2022  15:24    <DIR>          .idlerc
28-04-2022  09:06    <DIR>          .ipynb_checkpoints
30-11-2021  10:16    <DIR>          .ipython
27-04-2022  18:52    <DIR>          .jupyter
13-04-2022  16:35    <DIR>          .keras
27-03-2022  09:16                20 .lesshst
30-11-2021  10:16    <DIR>          .matplotlib
03-02-2022  10:33    <DIR>          .ssh
13-03-2022  19:08    <DIR>          .templateengine
07-02-2022  09:32             1,124 .viminfo
05-01-2022  09:22    <DIR>          .vscode
28-11-2021  10:40    <DIR>          ansel
30-01-2022  13:17    <DIR>          BasicGit
03-02-2022  10:33    <DIR>          chia-blockchain
28-11-2021  10:40    <DIR>          Contacts
27-03-2022  09:09    <DIR>          Desktop
13-03-2022  18:59    <DIR>          Documents
28-04-2022  19:39    <DIR>          Downloads
14-02-2019  14:12    <DIR>          FaceAssets
28-11-2021  10:40    <DIR>          Favorites
03-02-2022  09:35    <DIR>          Githud
03-02-2022  10:27    <DIR>          gitprojects
28-04-2022  20:17            20,750 Group.csv
04-01-2022  12:05                 0 hari.log
04-01-2022  20:16               287 hari1.log
13-04-2022  11:49                22 INFO_CONN.sql
30-11-2021  10:17    <DIR>          Jedi
28-11-2021  10:40    <DIR>          Links
12-04-2022  18:57    <DIR>          logs
26-03-2022  10:05    <DIR>          MATLAB
28-11-2021  10:40    <DIR>          Music
01-02-2022  22:28    <DIR>          newrepo
28-04-2022  13:20    <DIR>          OneDrive
12-04-2022  22:18    <DIR>          Oracle
31-01-2022  21:18    <DIR>          PlayListTake2
27-03-2022  10:36    <DIR>          Postman
05-01-2022  13:49           918,165 Practice.ipynb
07-02-2022  09:35    <DIR>          project_dir
26-04-2022  14:45    <DIR>          PycharmProjects
08-02-2022  21:34    <DIR>          Python
19-12-2021  22:17            10,933 Python by Hari.ipynb
28-11-2021  10:40    <DIR>          Saved Games
28-11-2021  11:31    <DIR>          Searches
```

```
30-01-2022  17:36    <DIR>          shopping
03-02-2022  10:20    <DIR>          Songs
13-03-2022  18:51    <DIR>          source
28-12-2021  12:01            11,417 StringFunctionsByHari.ipynb
05-01-2022  14:38    <DIR>          test
01-01-2022  14:39    <DIR>          Untitled Folder 3
30-12-2021  20:04               588 Untitled.ipynb
26-03-2022  10:53             2,876 Untitled1.ipynb
13-04-2022  14:26                72 Untitled2.ipynb
25-04-2022  19:54               765 Untitled3.ipynb
28-04-2022  09:03             6,825 Untitled4.ipynb
28-04-2022  20:15            72,832 Untitled5.ipynb
29-01-2022  13:47    <DIR>          Videos
              17 File(s)      1,055,184 bytes
              47 Dir(s)  54,111,469,568 bytes free
```

In [32]:

```
import requests
import io
url="https://raw.githubusercontent.com/cs109/2014_data/master/countries.csv"
s=requests.get(url).content
c=pd.read_csv(io.StringIO(s.decode('utf-8')))
```

In [33]:

```
c
```

Out[33]:

	Country	Region
0	Algeria	AFRICA
1	Angola	AFRICA
2	Benin	AFRICA
3	Botswana	AFRICA
4	Burkina	AFRICA
...	...	...
189	Paraguay	SOUTH AMERICA
190	Peru	SOUTH AMERICA
191	Suriname	SOUTH AMERICA
192	Uruguay	SOUTH AMERICA
193	Venezuela	SOUTH AMERICA

194 rows × 2 columns

In [44]:

```
e = c.groupby("Region")
```

In [45]:

```
e.first()
```

Out[45]:

Region	Country
AFRICA	Algeria
ASIA	Afghanistan
EUROPE	Albania
NORTH AMERICA	Antigua and Barbuda
OCEANIA	Australia
SOUTH AMERICA	Argentina

In [34]:

```
dfexcel = pd.read_excel(r"C:\Users\Administrator\OneDrive\Desktop\kidney_disease.xlsx")
```

In [186]:

```
dfexcel
```

Out[186]:

	id	age	bp	sg	al	su	rbc	pc	pcc	ba	...	pcv	wc	
0	0	48.0	80.0	1.020	1.0	0.0	NaN	normal	notpresent	notpresent	...	44	7800	
1	1	7.0	50.0	1.020	4.0	0.0	NaN	normal	notpresent	notpresent	...	38	6000	N
2	2	62.0	80.0	1.010	2.0	3.0	normal	normal	notpresent	notpresent	...	31	7500	N
3	3	48.0	70.0	1.005	4.0	0.0	normal	abnormal	present	notpresent	...	32	6700	
4	4	51.0	80.0	1.010	2.0	0.0	normal	normal	notpresent	notpresent	...	35	7300	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
395	395	55.0	80.0	1.020	0.0	0.0	normal	normal	notpresent	notpresent	...	47	6700	
396	396	42.0	70.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	...	54	7800	
397	397	12.0	80.0	1.020	0.0	0.0	normal	normal	notpresent	notpresent	...	49	6600	
398	398	17.0	60.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	...	51	7200	
399	399	58.0	80.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	...	53	6800	

400 rows × 26 columns

In [35]:

```
dfexcel
```

Out[35]:

	id	age	bp	sg	al	su	rbc	pc	pcc	ba	...	pcv	wc	
0	0	48.0	80.0	1.020	1.0	0.0	NaN	normal	notpresent	notpresent	...	44	7800	
1	1	7.0	50.0	1.020	4.0	0.0	NaN	normal	notpresent	notpresent	...	38	6000	N
2	2	62.0	80.0	1.010	2.0	3.0	normal	normal	notpresent	notpresent	...	31	7500	N
3	3	48.0	70.0	1.005	4.0	0.0	normal	abnormal	present	notpresent	...	32	6700	
4	4	51.0	80.0	1.010	2.0	0.0	normal	normal	notpresent	notpresent	...	35	7300	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
395	395	55.0	80.0	1.020	0.0	0.0	normal	normal	notpresent	notpresent	...	47	6700	
396	396	42.0	70.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	...	54	7800	
397	397	12.0	80.0	1.020	0.0	0.0	normal	normal	notpresent	notpresent	...	49	6600	
398	398	17.0	60.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	...	51	7200	
399	399	58.0	80.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	...	53	6800	

400 rows × 26 columns

In [38]:

```
df.info
```

Out[38]:

```
<bound method DataFrame.info of        Year Industry_aggregation_NZSIOC Indu
stry_code_NZSIOC  \
0      2020                     Level 1                99999
1      2020                     Level 1                99999
2      2020                     Level 1                99999
3      2020                     Level 1                99999
4      2020                     Level 1                99999
...     ...                         ...                  ...
37075  2013                     Level 3                 ZZ11
37076  2013                     Level 3                 ZZ11
37077  2013                     Level 3                 ZZ11
37078  2013                     Level 3                 ZZ11
37079  2013                     Level 3                 ZZ11

             Industry_name_NZSIOC               Units Variable_code  \
0                All industries  Dollars (millions)           H01
1                All industries  Dollars (millions)           H04
2                All industries  Dollars (millions)           H05
3                All industries  Dollars (millions)           H07
4                All industries  Dollars (millions)           H08
...                         ...                 ...           ...
37075  Food product manufacturing          Percentage           H37
37076  Food product manufacturing          Percentage           H38
37077  Food product manufacturing          Percentage           H39
37078  Food product manufacturing          Percentage           H40
37079  Food product manufacturing          Percentage           H41

                                      Variable_name      Variable_categor
y  \
0                                      Total income  Financial performanc
e
1      Sales, government funding, grants and subsidies  Financial performanc
e
2                   Interest, dividends and donations  Financial performanc
e
3                                Non-operating income  Financial performanc
e
4                                   Total expenditure  Financial performanc
e
...                                               ...
...
37075                                    Quick ratio       Financial ratio
s
37076             Margin on sales of goods for resale       Financial ratio
s
37077                                  Return on equity       Financial ratio
s
37078                           Return on total assets       Financial ratio
s
37079                            Liabilities structure       Financial ratio
s

         Value                                Industry_code_ANZSIC06
0      733,258  ANZSIC06 divisions A-S (excluding classes K633...
1      660,630  ANZSIC06 divisions A-S (excluding classes K633...
```

```
2       54,342  ANZSIC06 divisions A-S (excluding classes K633...
3       18,285  ANZSIC06 divisions A-S (excluding classes K633...
4      654,872  ANZSIC06 divisions A-S (excluding classes K633...
...        ...                                                ...
37075       52  ANZSIC06 groups C111, C112, C113, C114, C115, ...
37076       40  ANZSIC06 groups C111, C112, C113, C114, C115, ...
37077       12  ANZSIC06 groups C111, C112, C113, C114, C115, ...
37078        5  ANZSIC06 groups C111, C112, C113, C114, C115, ...
37079       46  ANZSIC06 groups C111, C112, C113, C114, C115, ...

[37080 rows x 10 columns]>
```

In [104]:

```
df = pd.DataFrame({'Product' :['a','b','c','d','e','f','g','h','i','j','k','l']
                   ,'Price' : [23,54,745,64,3553,699,599,446,464,35,789,199]})
df
```

Out[104]:

	Product	Price
0	a	23
1	b	54
2	c	745
3	d	64
4	e	3553
5	f	699
6	g	599
7	h	446
8	i	464
9	j	35
10	k	789
11	l	199

In [116]:

```
df.where(df['Price']> 100)
```

Out[116]:

	Product	Price
0	NaN	NaN
1	NaN	NaN
2	c	745.0
3	NaN	NaN
4	e	3553.0
5	f	699.0
6	g	599.0
7	h	446.0
8	i	464.0
9	NaN	NaN
10	k	789.0
11	l	199.0

In [117]:

```
print(df['Product']== 'a')
```

```
0      True
1     False
2     False
3     False
4     False
5     False
6     False
7     False
8     False
9     False
10    False
11    False
Name: Product, dtype: bool
```

In [7]:

```
Salary = [100000,130002,103500,679000,879000,675900]
Tax = [i*0.10 for i in Salary ]
```

In [8]:

```
Tax
```

Out[8]:

```
[10000.0, 13000.2, 10350.0, 67900.0, 87900.0, 67590.0]
```

In [16]:

```
a = [1,2,3,4]
b = [5,6,7,8]
c = [(i,j) for i in a for j in b]
c
```

Out[16]:

```
[(1, 5),
 (1, 6),
 (1, 7),
 (1, 8),
 (2, 5),
 (2, 6),
 (2, 7),
 (2, 8),
 (3, 5),
 (3, 6),
 (3, 7),
 (3, 8),
 (4, 5),
 (4, 6),
 (4, 7),
 (4, 8)]
```

In [49]:

```
class polygon:
    def __init__(self, no_of_sides):
        self.n = no_of_sides
        self.sides = [0 for i in range(no_of_sides)]

    def __c__(self):
        Sum = 1
        for i in args:
            Sum = Sum * i
            print(Sum)


    def inputsides(self):
        self.sides = [float(input("Enter side "+str(i+1)+" : ")) for i in range(self.n)]

class triangle(polygon):
    def __init__(self):
        polygon.__init__(self,3)

    def findArea(self):
        a, b, c = self.sides
        s = (a + b + c) / 2
        area = (s*(s-a)*(s-b)*(s-c)) ** 0.5
        print('The area of the triangle is %0.2f' %area)
```

In [50]:

```
t = triangle()
```

In [52]:

```
t.inputsides()
```

```
Enter side 1 : 15
Enter side 2 : 14
Enter side 3 : 13
```

In [54]:

```
t.findArea()
```

```
The area of the triangle is 84.00
```

In [ ]:

```

```

In [ ]:

```

```