```
import numpy as np
import pandas as pd
```

```
Mass = pd.Series([0.3,4,5,0.6,1000,560,80,102,0.01],index=['merc','ven','eart','mar','
Mass
```

```
merc       0.30
ven        4.00
eart       5.00
mar        0.60
jup     1000.00
sat      560.00
ura       80.00
nept     102.00
plu        0.01
dtype: float64
```

```
Mass+Mass
```

```
merc       0.60
ven        8.00
eart      10.00
mar        1.20
jup     2000.00
sat     1120.00
ura      160.00
nept     204.00
plu        0.02
dtype: float64
```

```
Mass*Mass
```

```
merc          0.0900
ven          16.0000
eart         25.0000
mar           0.3600
jup     1000000.0000
sat      313600.0000
ura        6400.0000
nept      10404.0000
plu           0.0001
dtype: float64
```

```
BigMass = Mass[Mass>100]
BigMass
```

```
jup     1000.0
sat      560.0
nept     102.0
dtype: float64
```

```
NewMass = Mass+BigMass
NewMass #nan bcz bigmass had nan
```

```
eart       NaN
jup     2000.0
mar        NaN
merc       NaN
nept     204.0
plu        NaN
sat     1120.0
ura        NaN
ven        NaN
dtype: float64
```

```
pd
```

```
<module 'pandas' from '/usr/local/lib/python3.7/dist-packages/pandas/__init__.py'>
```

```
print(pd)
```

```
<module 'pandas' from '/usr/local/lib/python3.7/dist-packages/pandas/__init__.py'>
```

```
pd.isnull(NewMass)
```

```
eart     True
jup     False
mar      True
merc     True
nept    False
plu      True
sat     False
ura      True
ven      True
dtype: bool
```

```
NewMass[pd.isnull(NewMass)]
```

```
eart   NaN
mar    NaN
merc   NaN
plu    NaN
ura    NaN
ven    NaN
dtype: float64
```

```
NewMass[~pd.isnull(NewMass)]
```

```
jup     2000.0
nept     204.0
sat     1120.0
dtype: float64
```

```
Mass['eris']=0.00029
Mass
```

```
merc       0.30000
```

```
ven        4.00000
eart       5.00000
mar        0.60000
jup     1000.00000
sat      560.00000
ura       80.00000
nept     102.00000
plu        0.01000
eris       0.00029
dtype: float64
```

```
Mass['moon']= 0.7
Mass
```

```
merc       0.30000
ven        4.00000
eart       5.00000
mar        0.60000
jup     1000.00000
sat      560.00000
ura       80.00000
nept     102.00000
plu        0.01000
eris       0.00029
moon       0.70000
dtype: float64
```

```
Mass.drop['moon'] #braces error
Mass
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-93-483e0ea56531> in <module>()
----> 1 Mass.drop['moon'] #braces error
      2 Mass

TypeError: 'method' object is not subscriptable
```

SEARCH STACK OVERFLOW

```
Mass=Mass.drop(['moon']) #just Mass.drop wont drop , i have to save
```

```
Mass
```

```
merc       0.30000
ven        4.00000
eart       5.00000
mar        0.60000
jup     1000.00000
sat      560.00000
ura       80.00000
nept     102.00000
plu        0.01000
eris       0.00029
dtype: float64
```

```
diameter=pd.Series((4000,12000,12000,3000,6000,142000,120000,51000,49000,2000),index=[
diameter
```

```
merc      4000
ven      12000
eart     12000
mar       3000
jup       6000
sat     142000
ura     120000
nept     51000
plu      49000
eris      2000
dtype: int64
```

```
  #density = mass / volume
```

```
  density=Mass/((np.pi*(diameter**3))/6)
  density
```

```
merc    8.952466e-12
ven     4.420971e-12
eart    5.526213e-12
mar     4.244132e-11
jup     8.841941e-09
sat     3.735290e-13
ura     8.841941e-14
nept    1.468558e-12
plu     1.623354e-16
eris    6.923240e-14
dtype: float64
```