In [1]:

```
#Number of approaches to Create Series object
```

In [2]:

```
import pandas as pd
```

In [3]:

```
#Syntax for creating series object
# varname=pandas.Series(object,dtype)
```

In [4]:

```
a=100
print(a,type(a))
s=pd.Series(a)
print(s,type(s))
```

```
100 <class 'int'>
0    100
dtype: int64 <class 'pandas.core.series.Series'>
```

In [6]:

```
a=100
print(a,type(a))
s=pd.Series(a,dtype="float")
print(s,type(s))
```

```
100 <class 'int'>
0    100.0
dtype: float64 <class 'pandas.core.series.Series'>
```

In [7]:

```
a=100.45
print(a,type(a))
s=pd.Series(a)
print(s,type(s))
```

```
100.45 <class 'float'>
0    100.45
dtype: float64 <class 'pandas.core.series.Series'>
```

In [8]:

```
r=range(10,16)
print(r,type(r))
s=pd.Series(r)
print(s,type(s))
```

```
range(10, 16) <class 'range'>
0    10
1    11
2    12
3    13
4    14
5    15
dtype: int64 <class 'pandas.core.series.Series'>
```

In [9]:

```
lst=[100,"Rossum",34.56,"Python",1]
print(lst,type(lst))
s=pd.Series(lst)
print(s,type(s))
```

```
[100, 'Rossum', 34.56, 'Python', 1] <class 'list'>
0       100
1    Rossum
2     34.56
3    Python
4         1
dtype: object <class 'pandas.core.series.Series'>
```

In [10]:

```
#Creating Series object with programmer defined Indices
#Syntax: varname=pandas.Series(object,index,dtype)
```

In [12]:

```
lst=[100,"Rossum",34.56,"Python",1]
print(lst,type(lst))
print("-"*20)
s=pd.Series(lst,["eno","ename","sal","sub","rank"])
print(s,type(s))
```

```
[100, 'Rossum', 34.56, 'Python', 1] <class 'list'>
--------------------
eno         100
ename    Rossum
sal       34.56
sub      Python
rank          1
dtype: object <class 'pandas.core.series.Series'>
```

In [13]:
```python
set={100,"Rossum",34.56,"Python",1}
print(set,type(set))
print("-"*20)
s=pd.Series(set,["eno","ename","sal","sub","rank"])  # TypeError: 'set' type i
print(s,type(s))
```

```
{1, 34.56, 100, 'Python', 'Rossum'} <class 'set'>
--------------------

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[13], line 4
      2 print(set,type(set))
      3 print("-"*20)
----> 4 s=pd.Series(set,["eno","ename","sal","sub","rank"])
      5 print(s,type(s))

File ~\anaconda3\Lib\site-packages\pandas\core\series.py:509, in Series.__ini
t__(self, data, index, dtype, name, copy, fastpath)
    507         data = data.copy()
    508 else:
--> 509     data = sanitize_array(data, index, dtype, copy)
    511     manager = get_option("mode.data_manager")
    512     if manager == "block":

File ~\anaconda3\Lib\site-packages\pandas\core\construction.py:590, in saniti
ze_array(data, index, dtype, copy, allow_2d)
    581     return sanitize_array(
    582         data,
    583         index=index,
   (...)
    586         allow_2d=allow_2d,
    587     )
    589 else:
--> 590     _sanitize_non_ordered(data)
    591     # materialize e.g. generators, convert e.g. tuples, abc.ValueView
    592     data = list(data)

File ~\anaconda3\Lib\site-packages\pandas\core\construction.py:641, in _sanit
ize_non_ordered(data)
    637 """
    638 Raise only for unordered sets, e.g., not for dict_keys
    639 """
    640 if isinstance(data, (set, frozenset)):
--> 641     raise TypeError(f"'{type(data).__name__}' type is unordered")

TypeError: 'set' type is unordered
```

In [14]:

```
d1={"sno":100,"sname":"Rossum","sal":34.56,"sub":"Python","rank":1}
print(d1,type(d1))
print("-"*20)
s=pd.Series(d1)
print(s,type(s))
```

```
{'sno': 100, 'sname': 'Rossum', 'sal': 34.56, 'sub': 'Python', 'rank': 1} <cl
ass 'dict'>
--------------------
sno         100
sname    Rossum
sal       34.56
sub      Python
rank          1
dtype: object <class 'pandas.core.series.Series'>
```

In [15]:

```
import numpy as np
lst=[100,200,300,400,500]
a=np.array(lst)
print(a,type(a))
```

```
[100 200 300 400 500] <class 'numpy.ndarray'>
```

In [16]:

```
s=pd.Series(a)
print(s,type(s))
```

```
0    100
1    200
2    300
3    400
4    500
dtype: int32 <class 'pandas.core.series.Series'>
```

In [17]:

```
s=pd.Series(a,index=["ID1","ID2","ID3","ID4","ID5"])
print(s,type(s))
```

```
ID1    100
ID2    200
ID3    300
ID4    400
ID5    500
dtype: int32 <class 'pandas.core.series.Series'>
```

In [18]:

```
s["ID1"]=155
print(s,type(s))
```

```
ID1    155
ID2    200
ID3    300
ID4    400
ID5    500
dtype: int32 <class 'pandas.core.series.Series'>
```

In [20]:

```
s["ID1"]
```

Out[20]:

```
155
```

In [21]:

```
s[0]
```

Out[21]:

```
155
```

In [22]:

```
d1={"sno":100,"sname":"Rossum","sal":34.56,"sub":"Python","rank":1}
print(d1,type(d1))
print("-"*20)
s=pd.Series(d1)
print(s,type(s))
```

```
{'sno': 100, 'sname': 'Rossum', 'sal': 34.56, 'sub': 'Python', 'rank': 1} <cl
ass 'dict'>
--------------------
sno         100
sname    Rossum
sal       34.56
sub      Python
rank          1
dtype: object <class 'pandas.core.series.Series'>
```

In [24]:

```
s["sno"]
```

Out[24]:

```
100
```

In [27]:

```
s["sno"]=1000
```

In [28]:

```
print(s)
```

```
sno        1000
sname    Rossum
sal       34.56
sub      Python
rank          1
dtype: object
```

In [29]:

```
s[0]
```

Out[29]:

```
1000
```

```
In [30]: d1={"sno":100,"sname":"Rossum","sal":34.56,"sub":"Python","rank":1}
         print(d1,type(d1))
         print("-"*20)
         s=pd.Series(d1)
         print(s,type(s))

         {'sno': 100, 'sname': 'Rossum', 'sal': 34.56, 'sub': 'Python', 'rank': 1} <cl
         ass 'dict'>
         --------------------
         sno         100
         sname    Rossum
         sal       34.56
         sub      Python
         rank          1
         dtype: object <class 'pandas.core.series.Series'>
```

```
In [31]: s[0]

Out[31]: 100
```

```
In [32]: s[3]

Out[32]: 'Python'
```

```
In [33]: s[4]

Out[33]: 1
```

```
In [ ]:
```