In [1]:

```
# Pandas
import pandas as pd
```

In [2]:

```
# creating 1st series
data1 = pd.Series([1,2,3,4,5])
data1
```

Out[2]:

```
0    1
1    2
2    3
3    4
4    5
dtype: int64
```

In [3]:

```
# creating 2nd series
data2 = pd.Series([1,2,3,4,5])
data2
```

Out[3]:

```
0    1
1    2
2    3
3    4
4    5
dtype: int64
```

In [4]:

```
# adding two series
data1+data2
```

Out[4]:

```
0     2
1     4
2     6
3     8
4    10
dtype: int64
```

In [5]:

```
type(data1)
```

Out[5]:

```
pandas.core.series.Series
```

In [6]:

```
type(data2)
```

Out[6]:

```
pandas.core.series.Series
```

In [7]:

```
res = data1+data2
res
```

Out[7]:

```
0     2
1     4
2     6
3     8
4    10
dtype: int64
```

In [8]:

```
type(res)
```

Out[8]:

```
pandas.core.series.Series
```

In [9]:

```
data3 = pd.Series([1,2,3])
data3
```

Out[9]:

```
0    1
1    2
2    3
dtype: int64
```

In [10]:

```
data2
```

Out[10]:

```
0    1
1    2
2    3
3    4
4    5
dtype: int64
```

In [11]:

```
# add two diff values set of series
data3+data2
```

Out[11]:

```
0    2.0
1    4.0
2    6.0
3    NaN
4    NaN
dtype: float64
```

In [ ]:

```
# NaN represnts Missing Value ....
```

In [12]:

```
data4 = pd.Series([1,2.2,30,40])
data5 = pd.Series([1,2.2,'30','40'])
```

In [13]:

```
#adding diff datatype series
data4+data5
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\Anaconda3\lib\site-packages\pandas\core\ops.py in na_op(x, y)
   1504         try:
-> 1505             result = expressions.evaluate(op, str_rep, x, y, **eval_
kwargs)
   1506         except TypeError:

~\Anaconda3\lib\site-packages\pandas\core\computation\expressions.py in eval
uate(op, op_str, a, b, use_numexpr, **eval_kwargs)
    207     if use_numexpr:
--> 208         return _evaluate(op, op_str, a, b, **eval_kwargs)
    209     return _evaluate_standard(op, op_str, a, b)

~\Anaconda3\lib\site-packages\pandas\core\computation\expressions.py in _eva
luate_numexpr(op, op_str, a, b, truediv, reversed, **eval_kwargs)
    122     if result is None:
--> 123         result = _evaluate_standard(op, op_str, a, b)
    124

~\Anaconda3\lib\site-packages\pandas\core\computation\expressions.py in _eva
luate_standard(op, op_str, a, b, **eval_kwargs)
     67     with np.errstate(all='ignore'):
---> 68         return op(a, b)
     69

TypeError: unsupported operand type(s) for +: 'float' and 'str'

During handling of the above exception, another exception occurred:

TypeError                                 Traceback (most recent call last)
<ipython-input-13-b89d752e2700> in <module>
      1 #adding diff datatype series
----> 2 data4+data5

~\Anaconda3\lib\site-packages\pandas\core\ops.py in wrapper(left, right)
   1581             rvalues = rvalues.values
   1582
-> 1583         result = safe_na_op(lvalues, rvalues)
   1584         return construct_result(left, result,
   1585                                 index=left.index, name=res_name, dty
pe=None)

~\Anaconda3\lib\site-packages\pandas\core\ops.py in safe_na_op(lvalues, rval
ues)
   1527         try:
   1528             with np.errstate(all='ignore'):
-> 1529                 return na_op(lvalues, rvalues)
   1530         except Exception:
   1531             if is_object_dtype(lvalues):

~\Anaconda3\lib\site-packages\pandas\core\ops.py in na_op(x, y)
   1505             result = expressions.evaluate(op, str_rep, x, y, **eval_
kwargs)
   1506         except TypeError:
-> 1507             result = masked_arith_op(x, y, op)
```

```
   1508
   1509         result = missing.fill_zeros(result, x, y, op_name, fill_zero
s)

~\Anaconda3\lib\site-packages\pandas\core\ops.py in masked_arith_op(x, y, o
p)
   1007             with np.errstate(all='ignore'):
   1008                 result[mask] = op(xrav[mask],
-> 1009                                   com.values_from_object(yrav[mas
k]))
   1010
   1011     else:

TypeError: unsupported operand type(s) for +: 'float' and 'str'
```

In [ ]: