In [1]:
```
import pandas as pd
```

In [2]:
```
d = {'a': 1, 'b': 2, 'c': 3}
ser = pd.Series(data=d, index=['a', 'b', 'c'])
ser
```

Out[2]:
```
a    1
b    2
c    3
dtype: int64
```

In [3]:
```
d={'a':1,'b':2,'c':3}
ser=pd.Series(data=d,index=['x','y','z'])
ser
```

Out[3]:
```
x   NaN
y   NaN
z   NaN
dtype: float64
```

In [4]:
```
r=[1,2]
ser=pd.Series(r,copy=False)
ser.iloc[0]=999
r
```

Out[4]:
```
[1, 2]
```

In [5]:
```
ser
```

Out[5]:
```
0    999
1      2
dtype: int64
```

In [15]:
```
 pd.Series([1, 2, 3]).array
```

Out[15]:
```
<PandasArray>
[1, 2, 3]
Length: 3, dtype: int64
```

In [19]:
```
ser = pd.Series(pd.Categorical(['a', 'b', 'a']))
ser.array
```

Out[19]:
```
['a', 'b', 'a']
Categories (2, object): ['a', 'b']
```

In [18]:
```
df = pd.DataFrame([[0, 2, 3], [0, 4, 1], [10, 20, 30]],
index=[4, 5, 6], columns=['A', 'B', 'C'])
df
```

Out[18]:

	A	B	C
4	0	2	3
5	0	4	1
6	10	20	30

In [22]:
```
d = {"a": 0.0, "b": 1.0, "c": 2.0}
```

In [23]:
```
pd.Series(d)
```

Out[23]:
```
a    0.0
b    1.0
c    2.0
dtype: float64
```

In [26]:
```
pd.Series(d,index=['b','b','c','d','e'])
```

In [27]:
```
s=pd.Series(np.random.randn(5),name='something')
s
```

Out[27]:
```
0   -1.921197
1   -1.527592
2    0.158199
3    0.678766
4    2.672852
Name: something, dtype: float64
```

In [28]:
```
s.name
```

Out[28]:
```
'something'
```

In [30]:
```
s2=s.rename('different')
```

In [31]:
```
s2
```

Out[31]:
```
0   -1.921197
1   -1.527592
2    0.158199
3    0.678766
4    2.672852
Name: different, dtype: float64
```

In [32]:
```
s2.name
```

Out[32]:
```
'different'
```

In [33]:
```
d = {
"one": pd.Series([1.0, 2.0, 3.0], index=["a", "b", "c"]),
"two": pd.Series([1.0, 2.0, 3.0, 4.0], index=["a", "b", "c", "d"]),
}
```

In [34]:
```
d
```

Out[34]:
```
{'one': a    1.0
 b    2.0
 c    3.0
 dtype: float64,
 'two': a    1.0
 b    2.0
 c    3.0
 d    4.0
 dtype: float64}
```

In [35]:
```
df=pd.DataFrame(d)
```

In [36]:
```
df
```

Out[36]:

	one	two
a	1.0	1.0
b	2.0	2.0
c	3.0	3.0
d	NaN	4.0

In [37]:
```
pd.DataFrame(d,index=['1','2','3'])
```

Out[37]:

	one	two
1	NaN	NaN
2	NaN	NaN
3	NaN	NaN

In [38]:
```
pd.DataFrame(d,index=['1','2','3'],columns=['two','three'])
```

Out[38]:

	two	three
1	NaN	NaN

	two	three
2	NaN	NaN
3	NaN	NaN

In [41]:
```
data2=[{'a':1,'b':2},{'a':5,'b':10,'c':20}]
pd.DataFrame(data2)
```

Out[41]:

	a	b	c
0	1	2	NaN
1	5	10	20.0

In [42]:
```
pd.DataFrame(data2,index=['first','second'])
```

Out[42]:

	a	b	c
first	1	2	NaN
second	5	10	20.0

In [43]:
```
pd.DataFrame(data2,columns=['a','b'])
```

Out[43]:

	a	b
0	1	2
1	5	10

In [45]:
```
from collections import namedtuple
point=namedtuple('point','x y')
pd.DataFrame([point(0,0),(0,3),(2,3)])
```

Out[45]:

	x	y
0	0	0
1	0	3
2	2	3

In [46]:
```
from dataclasses import make_dataclass
Point = make_dataclass("Point", [("x", int), ("y", int)])
pd.DataFrame([Point(0, 0), Point(0, 3), Point(2, 3)])
```

Out[46]:

	x	y
0	0	0
1	0	3
2	2	3

In [47]:
```
pd.DataFrame.from_dict(dict([("A", [1, 2, 3]), ("B", [4, 5, 6])]))
```

Out[47]:

	A	B
0	1	4
1	2	5
2	3	6

In [51]:
```
pd.DataFrame.from_dict(
dict([("A1", [1, 2, 3]), ("A2", [4, 5, 6])]),
orient="index",
columns=["affu1", "affu2", "affu3"],
)
```

Out[51]:

	affu1	affu2	affu3

	affu1	affu2	affu3
A1	1	2	3
A2	4	5	6

In [57]:
```
dfa = pd.DataFrame({"A": [1, 2, 3], "B": [4, 5, 6]})
```

In [58]:
```
dfa.assign(C=lambda x: x["A"] + x["B"], D=lambda x: x["A"] + x["C"])
```

Out[58]:

	A	B	C	D
0	1	4	5	6
1	2	5	7	9
2	3	6	9	12

In [59]:
```
df["one"]
```

Out[59]:
```
a    1.0
b    2.0
c    3.0
d    NaN
Name: one, dtype: float64
```

In [60]:
```
df["three"] = df["one"] * df["two"]
```

In [61]:
```
df["flag"] = df["one"] > 2
df
```

Out[61]:

	one	two	three	flag
a	1.0	1.0	1.0	False
b	2.0	2.0	4.0	False
c	3.0	3.0	9.0	True
d	NaN	4.0	NaN	False

In [62]:
```
del df["two"]
```

In [63]:
```
df
```

Out[63]:

	one	three	flag
a	1.0	1.0	False
b	2.0	4.0	False
c	3.0	9.0	True
d	NaN	NaN	False

In [64]:
```
df["foo"] = "bar"
```

In [65]:
```
df
```

Out[65]:

	one	three	flag	foo
a	1.0	1.0	False	bar
b	2.0	4.0	False	bar
c	3.0	9.0	True	bar
d	NaN	NaN	False	bar

In [69]:
```
df["one_trunc"] = df["one"][2:]
df
```

Out[69]:

	one	three	flag	foo	one_trunc
a	1.0	1.0	False	bar	NaN
b	2.0	4.0	False	bar	NaN
c	3.0	9.0	True	bar	3.0
d	NaN	NaN	False	bar	NaN

In [70]:
```
df.insert(1, "bar", df["one"])
```

In [71]:
```
df
```

Out[71]:

	one	bar	three	flag	foo	one_trunc
a	1.0	1.0	1.0	False	bar	NaN
b	2.0	2.0	4.0	False	bar	NaN
c	3.0	3.0	9.0	True	bar	3.0
d	NaN	NaN	NaN	False	bar	NaN

In [73]:
```
df.loc["b"]
```

Out[73]:
```
one             2.0
bar             2.0
three           4.0
flag          False
foo             bar
one_trunc       NaN
Name: b, dtype: object
```

In [76]:
```
 df.iloc[3]
```

Out[76]:
```
one             NaN
bar             NaN
three           NaN
flag          False
foo             bar
one_trunc       NaN
Name: d, dtype: object
```

In [77]:
```
df1 = pd.DataFrame({"a": [1, 0, 1], "b": [0, 1, 1]}, dtype=bool)
```

In [78]:
```
df2 = pd.DataFrame({"a": [0, 1, 1], "b": [1, 1, 0]}, dtype=bool)
```

In [79]:
```
df1
```

Out[79]:

	a	b
0	True	False
1	False	True
2	True	True

In [80]:
```
df2
```

Out[80]:

	a	b
0	False	True
1	True	True
2	True	False

In [81]:
```
df1&df2
```

Out[81]:

	a	b

	a	b
0	False	False
1	False	True
2	True	False

In [82]:
```
df1|df2
```

Out[82]:

	a	b
0	True	True
1	True	True
2	True	True

In [83]:
```
df1^df2
```

Out[83]:

	a	b
0	True	True
1	True	False
2	False	True

In [84]:
```
-df1
```

Out[84]:

	a	b
0	False	True
1	True	False
2	False	False

In [86]:
```
df[:4].T
```

Out[86]:

	a	b	c	d
one	1.0	2.0	3.0	NaN
bar	1.0	2.0	3.0	NaN
three	1.0	4.0	9.0	NaN
flag	False	False	True	False
foo	bar	bar	bar	bar
one_trunc	NaN	NaN	3.0	NaN

In [88]:
```
ser = pd.Series([1, 2, 3, 4])
```

In [89]:
```
np.exp(ser)
```

Out[89]:
```
0     2.718282
1     7.389056
2    20.085537
3    54.598150
dtype: float64
```

In [90]:
```
ser1 = pd.Series([1, 2, 3], index=["a", "b", "c"])
ser2 = pd.Series([1, 3, 5], index=["b", "a", "c"])
```

In [91]:
```
ser1
```

Out[91]:
```
a    1
b    2
c    3
dtype: int64
```

In [92]:
```
ser2
```

Out[92]:
```
b    1
a    3
c    5
dtype: int64
```

In [93]:
```
np.remainder(ser1, ser2)
```

Out[93]:
```
a    1
b    0
c    3
dtype: int64
```

In [94]:
```
ser3 = pd.Series([2, 4, 6], index=["b", "c", "d"])
ser3
```

Out[94]:
```
b    2
c    4
d    6
dtype: int64
```

In [95]:
```
np.remainder(ser1, ser3)
```

Out[95]:
```
a    NaN
b    0.0
c    3.0
d    NaN
dtype: float64
```

In [96]:
```
ser = pd.Series([1, 2, 3])
idx = pd.Index([4, 5, 6])
np.maximum(ser, idx)
```

Out[96]:
```
0    4
1    5
2    6
dtype: int64
```

In [97]:
```
pd.DataFrame(np.random.randn(3, 12))
```

Out[97]:

	0	1	2	3	4	5	6	7	8	9	10	11
0	-0.715203	0.291728	-1.298993	-1.776975	0.274046	-1.347017	0.173392	1.146432	-1.234608	0.168341	1.493683	0.463140
1	-0.856243	-0.207431	-0.503905	0.002601	-0.336264	-0.001820	-0.628585	-1.024277	-0.663727	-0.403741	-0.872216	0.570693
2	0.587196	0.311331	-1.115662	0.252656	0.775999	-0.171518	-1.331914	-0.685189	-0.048110	0.346646	-0.359746	1.671713

In [98]:
```
pd.set_option("display.width", 50)
```

In [99]:
```
pd.DataFrame(np.random.randn(3, 10))
```

Out[99]:

	0	1	2	3	4	5	6	7	8	9
0	0.935369	-0.154843	0.243513	0.006868	-0.377626	0.390975	-0.608133	-0.812526	0.265172	0.113255
1	-0.864348	0.975884	-1.086982	0.152945	-0.072444	0.379817	-1.896713	0.094998	-0.644726	0.727420
2	0.418600	-0.011627	-0.975499	0.346288	-1.609320	0.806050	1.357837	1.984942	-0.253515	-0.046951

In [100…
```
df = pd.DataFrame({"foo1": np.random.randn(5), "foo2": np.random.randn(5)})
df
```

Out[100…

	foo1	foo2
0	-1.298507	0.222342
1	0.032522	-0.285510
2	-0.636398	1.117675
3	-0.734688	-0.384261
4	-0.385618	0.301733

In [101…

```
df.foo1
```

Out[101…

```
0   -1.298507
1    0.032522
2   -0.636398
3   -0.734688
4   -0.385618
Name: foo1, dtype: float64
```