

Pandas Series

A Pandas Series is like a column in a table. It is a one-dimensional array holding data of any type.

```
In [1]: ▶ import pandas as pd

a = ["CS", "IT", "CSCE"]
branch = pd.Series(a)
print(branch)
```

```
0      CS
1      IT
2     CSCE
dtype: object
```

Lables

If noting else is specified, the values are labled with their index number. First value has index 0, second value has index 1, etc.

Create Labels

```
In [2]: ▶ a = ["CS", "IT", "CSCE"]
branch = pd. Series(a, index=['a', 'b', 'c'])
print(branch)
print(branch['b'])
```

```
a      CS
b      IT
c     CSCE
dtype: object
IT
```

Data Frames

A pandas dataframe is a 2 dimensional data structure, like a 2 dimensional array, or a table with rows and columns

```
In [3]: ▶ import pandas as pd
data = {
    'empid': [1,2,3],
    'salary': [50000, 40000, 45000]
}

df = pd.DataFrame(data)
print(df)
```

	empid	salary
0	1	50000
1	2	40000
2	3	45000

Local Row

Pandas use the loc attribute to return one or more specified rows

```
In [4]: ▶ print(df.loc[0])
```

	empid	salary
0	1	50000

Name: 0, dtype: int64

```
In [6]: ▶ print(df.loc[[0, 1]])
```

	empid	salary
0	1	50000
1	2	40000

```
In [7]: ▶ import pandas as pd #Just like the nickname given to people
```

```
In [8]: ▶ pd.__version__
```

Out[8]: '2.0.3'

```
dir(pd)
```

```
Out[9]: ['ArrowDtype',  
         'BooleanDtype',  
         'Categorical',  
         'CategoricalDtype',  
         'CategoricalIndex',  
         'DataFrame',  
         'DateOffset',  
         'DatetimeIndex',  
         'DatetimeTZDtype',  
         'ExcelFile',  
         'ExcelWriter',  
         'Flags',  
         'Float32Dtype',  
         'Float64Dtype',  
         'Grouper',  
         'HDFStore',  
         'Index',  
         'IndexSlice',  
         'Int16Dtype',  
         'Int32Dtype',
```

```
df = pd.read_csv('Salaries.csv')
```

```
In [11]: print(df.to_string()) #Print entire Data Frame
```

	rank	discipline	phd	service	sex	salary
0	Prof	B	56.0	49	Male	186960.0
1	Prof	A	12.0	6	Male	93000.0
2	Prof	A	23.0	20	Male	110515.0
3	Prof	A	40.0	31	Male	131205.0
4	Prof	B	20.0	18	Male	104800.0
5	Prof	A	20.0	20	Male	122400.0
6	AssocProf	A	20.0	17	Male	81285.0
7	Prof	A	18.0	18	Male	NaN
8	Prof	A	29.0	19	Male	94350.0
9	Prof	A	51.0	51	Male	57800.0
10	Prof	B	39.0	33	Male	128250.0
11	Prof	B	23.0	23	Male	134778.0
12	AsstProf	B	1.0	0	Male	88000.0
13	Prof	B	NaN	33	Male	162200.0
14	Prof	B	25.0	19	Male	153750.0
15	Prof	B	17.0	3	Male	150480.0
16	AsstProf	B	8.0	3	Male	75044.0
17	AsstProf	B	4.0	0	Male	92000.0
18	Prof	A	19.0	7	Male	107300.0
19	Prof	A	29.0	27	Male	150500.0
20	AsstProf	B	4.0	4	Male	92000.0
21	Prof	A	33.0	30	Male	103106.0
22	AsstProf	A	4.0	2	Male	73000.0
23	AsstProf	A	2.0	0	Male	85000.0
24	Prof	A	30.0	23	Male	91100.0
25	Prof	B	35.0	31	Male	99418.0
26	Prof	A	38.0	19	Male	148750.0
27	Prof	A	45.0	43	Male	155865.0
28	AsstProf	B	7.0	2	Male	NaN
29	Prof	B	21.0	20	Male	123683.0
30	AssocProf	B	9.0	7	Male	107008.0
31	Prof	B	22.0	21	Male	155750.0
32	Prof	A	27.0	19	Male	103275.0
33	Prof	B	18.0	18	Male	120000.0
34	AssocProf	B	NaN	8	Male	119800.0
35	Prof	B	28.0	23	Male	126933.0
36	Prof	B	45.0	45	Male	146856.0
37	Prof	A	20.0	8	Male	102000.0
38	AsstProf	B	4.0	3	Male	91000.0
39	Prof	B	18.0	18	Female	129000.0
40	Prof	A	39.0	36	Female	137000.0
41	AssocProf	A	13.0	8	Female	74830.0
42	AsstProf	B	4.0	2	Female	80225.0
43	AsstProf	B	5.0	0	Female	77000.0
44	Prof	B	23.0	19	Female	151768.0
45	Prof	B	25.0	25	Female	140096.0
46	AsstProf	B	11.0	3	Female	74692.0
47	AssocProf	B	11.0	11	Female	103613.0
48	Prof	B	17.0	17	Female	111512.0
49	Prof	B	17.0	18	Female	122960.0
50	AsstProf	B	10.0	5	Female	97032.0
51	Prof	B	20.0	14	Female	127512.0

52	Prof	A	12.0	0	Female	105000.0
53	AsstProf	A	5.0	3	Female	73500.0
54	AssocProf	A	25.0	22	Female	62884.0
55	AsstProf	A	2.0	0	Female	72500.0
56	AssocProf	A	10.0	8	Female	77500.0
57	AsstProf	A	3.0	1	Female	72500.0
58	Prof	B	36.0	26	Female	144651.0
59	AssocProf	B	12.0	10	Female	103994.0
60	AsstProf	B	3.0	3	Female	92000.0
61	AssocProf	B	13.0	10	Female	103750.0
62	AssocProf	B	14.0	7	Female	109650.0
63	Prof	A	29.0	27	Female	91000.0
64	AssocProf	A	26.0	24	Female	73300.0
65	Prof	A	36.0	19	Female	117555.0
66	AsstProf	A	7.0	6	Female	63100.0
67	Prof	A	17.0	11	Female	90450.0
68	AsstProf	A	4.0	2	Female	77500.0
69	Prof	A	28.0	7	Female	116450.0
70	AsstProf	A	8.0	3	Female	78500.0
71	AssocProf	B	12.0	9	Female	71065.0
72	Prof	B	24.0	15	Female	161101.0
73	Prof	B	18.0	10	Female	105450.0
74	AssocProf	B	19.0	6	Female	104542.0
75	Prof	B	17.0	17	Female	124312.0
76	Prof	A	28.0	14	Female	109954.0
77	Prof	A	23.0	15	Female	109646.0

In [12]: `print(df)`

	rank	discipline	phd	service	sex	salary
0	Prof	B	56.0	49	Male	186960.0
1	Prof	A	12.0	6	Male	93000.0
2	Prof	A	23.0	20	Male	110515.0
3	Prof	A	40.0	31	Male	131205.0
4	Prof	B	20.0	18	Male	104800.0
..
73	Prof	B	18.0	10	Female	105450.0
74	AssocProf	B	19.0	6	Female	104542.0
75	Prof	B	17.0	17	Female	124312.0
76	Prof	A	28.0	14	Female	109954.0
77	Prof	A	23.0	15	Female	109646.0

[78 rows x 6 columns]

In [13]: `type(df)`

Out[13]: `pandas.core.frame.DataFrame`

In [14]: `df.shape`

Out[14]: `(78, 6)`

In [15]: `df.ndim`

Out[15]: 2

In [16]: `df.size`

Out[16]: 468

In [17]: `df.columns`

Out[17]: Index(['rank', 'discipline', 'phd', 'service', 'sex', 'salary'], dtype='object')

In [18]: `df.columns.tolist()`

Out[18]: ['rank', 'discipline', 'phd', 'service', 'sex', 'salary']

In [19]: `df.head()` *#Default 5 rows*

Out[19]:

	rank	discipline	phd	service	sex	salary
0	Prof	B	56.0	49	Male	186960.0
1	Prof	A	12.0	6	Male	93000.0
2	Prof	A	23.0	20	Male	110515.0
3	Prof	A	40.0	31	Male	131205.0
4	Prof	B	20.0	18	Male	104800.0

In [20]: `df.head(20)`

Out[20]:

	rank	discipline	phd	service	sex	salary
0	Prof	B	56.0	49	Male	186960.0
1	Prof	A	12.0	6	Male	93000.0
2	Prof	A	23.0	20	Male	110515.0
3	Prof	A	40.0	31	Male	131205.0
4	Prof	B	20.0	18	Male	104800.0
5	Prof	A	20.0	20	Male	122400.0
6	AssocProf	A	20.0	17	Male	81285.0
7	Prof	A	18.0	18	Male	NaN
8	Prof	A	29.0	19	Male	94350.0
9	Prof	A	51.0	51	Male	57800.0
10	Prof	B	39.0	33	Male	128250.0
11	Prof	B	23.0	23	Male	134778.0
12	AsstProf	B	1.0	0	Male	88000.0
13	Prof	B	NaN	33	Male	162200.0
14	Prof	B	25.0	19	Male	153750.0
15	Prof	B	17.0	3	Male	150480.0
16	AsstProf	B	8.0	3	Male	75044.0
17	AsstProf	B	4.0	0	Male	92000.0
18	Prof	A	19.0	7	Male	107300.0
19	Prof	A	29.0	27	Male	150500.0

In [21]: `df.tail()`

Out[21]:

	rank	discipline	phd	service	sex	salary
73	Prof	B	18.0	10	Female	105450.0
74	AssocProf	B	19.0	6	Female	104542.0
75	Prof	B	17.0	17	Female	124312.0
76	Prof	A	28.0	14	Female	109954.0
77	Prof	A	23.0	15	Female	109646.0

```
In [22]: df.tail(10)
```

```
Out[22]:
```

	rank	discipline	phd	service	sex	salary
68	AsstProf	A	4.0	2	Female	77500.0
69	Prof	A	28.0	7	Female	116450.0
70	AsstProf	A	8.0	3	Female	78500.0
71	AssocProf	B	12.0	9	Female	71065.0
72	Prof	B	24.0	15	Female	161101.0
73	Prof	B	18.0	10	Female	105450.0
74	AssocProf	B	19.0	6	Female	104542.0
75	Prof	B	17.0	17	Female	124312.0
76	Prof	A	28.0	14	Female	109954.0
77	Prof	A	23.0	15	Female	109646.0

```
In [23]: df.sample() #takes 1 row randomly
```

```
Out[23]:
```

	rank	discipline	phd	service	sex	salary
51	Prof	B	20.0	14	Female	127512.0

```
In [24]: df.sample(5)
```

```
Out[24]:
```

	rank	discipline	phd	service	sex	salary
32	Prof	A	27.0	19	Male	103275.0
7	Prof	A	18.0	18	Male	NaN
36	Prof	B	45.0	45	Male	146856.0
11	Prof	B	23.0	23	Male	134778.0
0	Prof	B	56.0	49	Male	186960.0

```
In [25]: df["salary"]
```

```
Out[25]:
```

0	186960.0
1	93000.0
2	110515.0
3	131205.0
4	104800.0
...	
73	105450.0
74	104542.0
75	124312.0
76	109954.0
77	109646.0

Name: salary, Length: 78, dtype: float64

In [26]: `df.rank`

```
Out[26]: <bound method NDFrame.rank of
sex      salary
0      Prof      B  56.0      49   Male  186960.0
1      Prof      A  12.0       6   Male   93000.0
2      Prof      A  23.0      20   Male  110515.0
3      Prof      A  40.0      31   Male  131205.0
4      Prof      B  20.0      18   Male  104800.0
..      ...      ...      ...      ...      ...
73     Prof      B  18.0      10  Female  105450.0
74  AssocProf      B  19.0       6  Female  104542.0
75     Prof      B  17.0      17  Female  124312.0
76     Prof      A  28.0      14  Female  109954.0
77     Prof      A  23.0      15  Female  109646.0

[78 rows x 6 columns]>
```

In [27]: `df[["rank", "service"]]`

```
Out[27]:
```

	rank	service
0	Prof	49
1	Prof	6
2	Prof	20
3	Prof	31
4	Prof	18
...
73	Prof	10
74	AssocProf	6
75	Prof	17
76	Prof	14
77	Prof	15

78 rows × 2 columns

In [28]: `df["rank"].unique()`

```
Out[28]: array(['Prof', 'AssocProf', 'AsstProf'], dtype=object)
```

In [29]: `df["rank"].value_counts()`

```
Out[29]: rank
Prof      46
AsstProf   19
AssocProf  13
Name: count, dtype: int64
```

```
In [30]: df["service"].value_counts()
```

```
Out[30]: service
3      7
19     6
0      6
18     5
2      4
7      4
8      4
6      3
20     3
10     3
17     3
23     3
14     2
11     2
15     2
27     2
31     2
33     2
30     1
51     1
9      1
24     1
26     1
1      1
22     1
5      1
4      1
25     1
36     1
45     1
21     1
43     1
49     1
Name: count, dtype: int64
```

```
In [31]: df["salary"].max()
```

```
Out[31]: 186960.0
```

```
In [32]: df["salary"].min()
```

```
Out[32]: 57800.0
```

```
In [33]: df["rank"].value_counts()
```

```
Out[33]: rank
Prof      46
AsstProf  19
AssocProf 13
Name: count, dtype: int64
```

```
In [34]: df["service"].value_counts()
```

```
Out[34]: service
3      7
19     6
0      6
18     5
2      4
7      4
8      4
6      3
20     3
10     3
17     3
23     3
14     2
11     2
15     2
27     2
31     2
33     2
30     1
51     1
9      1
24     1
26     1
1      1
22     1
5      1
4      1
25     1
36     1
45     1
21     1
43     1
49     1
Name: count, dtype: int64
```

```
In [35]: df["salary"].max()
```

```
Out[35]: 186960.0
```

```
In [36]: df["salary"].min()
```

```
Out[36]: 57800.0
```

```
In [37]: df["salary"].mean()
```

```
Out[37]: 108003.3552631579
```

```
In [38]: df["rank"].value_counts(normalize=True)
```

```
Out[38]: rank
Prof      0.589744
AsstProf  0.243590
AssocProf 0.166667
Name: proportion, dtype: float64
```

```
In [39]: df["salary"] > 100000
```

```
Out[39]: 0      True
1      False
2      True
3      True
4      True
...
73     True
74     True
75     True
76     True
77     True
Name: salary, Length: 78, dtype: bool
```

```
In [41]: df[df["salary"] > 100000]
```

```
Out[41]:
```

	rank	discipline	phd	service	sex	salary
0	Prof	B	56.0	49	Male	186960.0
2	Prof	A	23.0	20	Male	110515.0
3	Prof	A	40.0	31	Male	131205.0
4	Prof	B	20.0	18	Male	104800.0
5	Prof	A	20.0	20	Male	122400.0
10	Prof	B	39.0	33	Male	128250.0
11	Prof	B	23.0	23	Male	134778.0
13	Prof	B	NaN	33	Male	162200.0
14	Prof	B	25.0	19	Male	153750.0
15	Prof	B	17.0	3	Male	150480.0
18	Prof	A	19.0	7	Male	107300.0
19	Prof	A	29.0	27	Male	150500.0
21	Prof	A	33.0	30	Male	103106.0
26	Prof	A	38.0	19	Male	148750.0
27	Prof	A	45.0	43	Male	155865.0
29	Prof	B	21.0	20	Male	123683.0
30	AssocProf	B	9.0	7	Male	107008.0
31	Prof	B	22.0	21	Male	155750.0
32	Prof	A	27.0	19	Male	103275.0
33	Prof	B	18.0	18	Male	120000.0
34	AssocProf	B	NaN	8	Male	119800.0
35	Prof	B	28.0	23	Male	126933.0
36	Prof	B	45.0	45	Male	146856.0
37	Prof	A	20.0	8	Male	102000.0
39	Prof	B	18.0	18	Female	129000.0
40	Prof	A	39.0	36	Female	137000.0
44	Prof	B	23.0	19	Female	151768.0
45	Prof	B	25.0	25	Female	140096.0
47	AssocProf	B	11.0	11	Female	103613.0
48	Prof	B	17.0	17	Female	111512.0
49	Prof	B	17.0	18	Female	122960.0
51	Prof	B	20.0	14	Female	127512.0
52	Prof	A	12.0	0	Female	105000.0
58	Prof	B	36.0	26	Female	144651.0

	rank	discipline	phd	service	sex	salary
59	AssocProf	B	12.0	10	Female	103994.0
61	AssocProf	B	13.0	10	Female	103750.0
62	AssocProf	B	14.0	7	Female	109650.0
65	Prof	A	36.0	19	Female	117555.0
69	Prof	A	28.0	7	Female	116450.0
72	Prof	B	24.0	15	Female	161101.0
73	Prof	B	18.0	10	Female	105450.0
74	AssocProf	B	19.0	6	Female	104542.0
75	Prof	B	17.0	17	Female	124312.0
76	Prof	A	28.0	14	Female	109954.0
77	Prof	A	23.0	15	Female	109646.0

```
In [42]: df[(df["salary"] > 100000) & (df["rank"] == 'Prof')]
```

Out[42]:

	rank	discipline	phd	service	sex	salary
0	Prof	B	56.0	49	Male	186960.0
2	Prof	A	23.0	20	Male	110515.0
3	Prof	A	40.0	31	Male	131205.0
4	Prof	B	20.0	18	Male	104800.0
5	Prof	A	20.0	20	Male	122400.0
10	Prof	B	39.0	33	Male	128250.0
11	Prof	B	23.0	23	Male	134778.0
13	Prof	B	NaN	33	Male	162200.0
14	Prof	B	25.0	19	Male	153750.0
15	Prof	B	17.0	3	Male	150480.0
18	Prof	A	19.0	7	Male	107300.0
19	Prof	A	29.0	27	Male	150500.0
21	Prof	A	33.0	30	Male	103106.0
26	Prof	A	38.0	19	Male	148750.0
27	Prof	A	45.0	43	Male	155865.0
29	Prof	B	21.0	20	Male	123683.0
31	Prof	B	22.0	21	Male	155750.0
32	Prof	A	27.0	19	Male	103275.0
33	Prof	B	18.0	18	Male	120000.0
35	Prof	B	28.0	23	Male	126933.0
36	Prof	B	45.0	45	Male	146856.0
37	Prof	A	20.0	8	Male	102000.0
39	Prof	B	18.0	18	Female	129000.0
40	Prof	A	39.0	36	Female	137000.0
44	Prof	B	23.0	19	Female	151768.0
45	Prof	B	25.0	25	Female	140096.0
48	Prof	B	17.0	17	Female	111512.0
49	Prof	B	17.0	18	Female	122960.0
51	Prof	B	20.0	14	Female	127512.0
52	Prof	A	12.0	0	Female	105000.0
58	Prof	B	36.0	26	Female	144651.0
65	Prof	A	36.0	19	Female	117555.0
69	Prof	A	28.0	7	Female	116450.0
72	Prof	B	24.0	15	Female	161101.0

	rank	discipline	phd	service	sex	salary
73	Prof	B	18.0	10	Female	105450.0
75	Prof	B	17.0	17	Female	124312.0
76	Prof	A	28.0	14	Female	109954.0
77	Prof	A	23.0	15	Female	109646.0

```
In [43]: df[(df["salary"] > 100000) & (df["rank"] == 'Prof') & (df["sex"] == 'Female')]
```

```
Out[43]:
```

	rank	discipline	phd	service	sex	salary
39	Prof	B	18.0	18	Female	129000.0
40	Prof	A	39.0	36	Female	137000.0
44	Prof	B	23.0	19	Female	151768.0
45	Prof	B	25.0	25	Female	140096.0
48	Prof	B	17.0	17	Female	111512.0
49	Prof	B	17.0	18	Female	122960.0
51	Prof	B	20.0	14	Female	127512.0
52	Prof	A	12.0	0	Female	105000.0
58	Prof	B	36.0	26	Female	144651.0
65	Prof	A	36.0	19	Female	117555.0
69	Prof	A	28.0	7	Female	116450.0
72	Prof	B	24.0	15	Female	161101.0
73	Prof	B	18.0	10	Female	105450.0
75	Prof	B	17.0	17	Female	124312.0
76	Prof	A	28.0	14	Female	109954.0
77	Prof	A	23.0	15	Female	109646.0

```
In [44]: df.isnull().any(axis=0)
```

```
Out[44]: rank          False
discipline        False
phd                True
service           False
sex               False
salary            True
dtype: bool
```



```
In [45]: df.isnull().any(axis=1)
```

```
Out[45]: 0    False
         1    False
         2    False
         3    False
         4    False
         ...
        73    False
        74    False
        75    False
        76    False
        77    False
        Length: 78, dtype: bool
```

```
In [46]: df[df.isnull().any(axis=1)]
```

```
Out[46]:
```

	rank	discipline	phd	service	sex	salary
7	Prof	A	18.0	18	Male	NaN
13	Prof	B	NaN	33	Male	162200.0
28	AsstProf	B	7.0	2	Male	NaN
34	AssocProf	B	NaN	8	Male	119800.0

Handling Missing Data

```
In [47]: df['phd'].mean()
```

```
Out[47]: 19.605263157894736
```

```
In [48]: df['phd'].fillna(df['phd'].mean())
```

```
Out[48]: 0    56.0
         1    12.0
         2    23.0
         3    40.0
         4    20.0
         ...
        73    18.0
        74    19.0
        75    17.0
        76    28.0
        77    23.0
        Name: phd, Length: 78, dtype: float64
```

```
In [50]: df['phd'] = df['phd'].fillna(df['phd'].mean())
```

In [51]: `df[df.isnull().any(axis=1)]`

Out[51]:

	rank	discipline	phd	service	sex	salary
7	Prof	A	19.605263	18	Male	NaN
28	AsstProf	B	19.605263	2	Male	NaN

In [52]: `#Delete the rows which has missing values
df.dropna(inplace=True)`

In [53]: `df[df.isnull().any(axis=1)]`

Out[53]:

	rank	discipline	phd	service	sex	salary
--	------	------------	-----	---------	-----	--------

In [54]: `df.shape`

Out[54]: (76, 6)

In [55]: `df2 = {'empid':1001, 'rank':'Prof', 'discipline':'B', 'phd':13, 'service':!
df = df.append(df2, ignore_index = True)
display(df)`

```
-----
--
AttributeError                                Traceback (most recent call las
t)
~\AppData\Local\Temp\ipykernel_25564\2108356985.py in ?()
      1 df2 = {'empid':1001, 'rank':'Prof', 'discipline':'B', 'phd':13,
'service':5, 'sex':'Male', 'salary':'NaN'}
----> 2 df = df.append(df2, ignore_index = True)
      3 display(df)

c:\Users\KIIT\anaconda3\Lib\site-packages\pandas\core\generic.py in ?(sel
f, name)
    5985         and name not in self._accessors
    5986         and self._info_axis._can_hold_identifiers_and_holds_n
ame(name)
    5987     ):
    5988         return self[name]
-> 5989     return object.__getattr__(self, name)


AttributeError: 'DataFrame' object has no attribute 'append'
```

In [56]: `del df['phd']`

In [57]:  `df.head()`

Out[57]:

	rank	discipline	service	sex	salary
0	Prof	B	49	Male	186960.0
1	Prof	A	6	Male	93000.0
2	Prof	A	20	Male	110515.0
3	Prof	A	31	Male	131205.0
4	Prof	B	18	Male	104800.0

In [58]:  `#How to access partial data
print(df.iloc[0:10,0:2])`

	rank	discipline
0	Prof	B
1	Prof	A
2	Prof	A
3	Prof	A
4	Prof	B
5	Prof	A
6	AssocProf	A
8	Prof	A
9	Prof	A
10	Prof	B

In []: 