pandas DataFrame

Setup

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

Creation

Create Simple DataFrame

```
mda = np.array([
        [1, 2, 3],
        [4, 5, 6],
        [7, 8, 9]
    ])
df1 = pd.DataFrame(mda)
df1
```

	0	**1**	**2**
0	1	2	3
1	4	5	6
2	7	8	9

```
df1.columns=['A','B','C']
df1
```

	A	**B**	**C**
0	1	2	3
1	4	5	6
2	7	8	9

```
df1.index=np.arange(1,len(df1)+1)
df1
```

	A	B	C
1	1	2	3
2	4	5	6
3	7	8	9

```
df2 = pd.DataFrame(mda, columns=['A','B','C'], index=np.arange(1,len(mda)+1) )
df2
```

	A	B	C
1	1	2	3
2	4	5	6
3	7	8	9

Create DataFrame using Series as Rows

```
people = pd.Series(['Aaron','Brian','Christine','Di'], index=['A','B','C','D'])
places = pd.Series(['Alington','Boston','Cleveland'], index=['A','B','C'])
things = pd.Series(['Apple','Banana','Car'], index=['A','B','C'])

df3 = pd.DataFrame([people, places, things])
df3
```

	A	B	C	D
0	Aaron	Brian	Christine	Di
1	Alington	Boston	Cleveland	NaN
2	Apple	Banana	Car	NaN

```
df4 = pd.DataFrame([people, places, things],
                   index = ['People','Place','Thing'],
                   columns = ['A','B','D'])
df4
```

	A	B	D
People	Aaron	Brian	Di
Place	Alington	Boston	NaN
Thing	Apple	Banana	NaN

Create DataFrame using concat()

```
np.random.seed(1)
ar1 = np.random.choice(['A','B','C','D','F'], 100, p=[.2,.4,.3,.08,.02])
ar2 = np.random.choice(['A','B','C','D','F'], 50, p=[.3,.4,.2,.1,0])
ar3 = np.random.choice(['a','b','c','d','f'], 200, p=[.15,.45,.25,.13,.02])
s1 = pd.Series(ar1)
s2 = pd.Series(ar2)
s3 = pd.Series(ar3)
df5 = pd.concat([s1,s2,s3], axis=1)
df5.columns=['grades1','grades2','grades3']
df5
```

	grades1	grades2	grades3
0	B	B	a
1	C	B	d
2	A	C	b
3	B	B	b
4	A	D	b
...	...	...	...
195	NaN	NaN	c
196	NaN	NaN	c
197	NaN	NaN	c
198	NaN	NaN	c
199	NaN	NaN	a

200 rows × 3 columns

Create DataFrame from CSV

```
df = pd.read_csv('/content/ds_salaries.csv')
df
```

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary
0	0	2020	MI	FT	Data Scientist	70000
1	1	2020	SE	FT	Machine Learning Scientist	260000
2	2	2020	SE	FT	Big Data Engineer	85000
3	3	2020	MI	FT	Product Data Analyst	20000
4	4	2020	SE	FT	Machine Learning Engineer	150000
...	...	...	...	...	...	...
602	602	2022	SE	FT	Data Engineer	154000

Properties

604	604	2022	SE	FT	Data	129000

Column Names and Row index

Analyst

```
df.columns
```

```
Index(['Unnamed: 0', 'work_year', 'experience_level', 'employment_type',
       'job_title', 'salary', 'salary_currency', 'salary_in_usd',
       'employee_residence', 'remote_ratio', 'company_location',
       'company_size'],
      dtype='object')
```

```
df.index
```

```
RangeIndex(start=0, stop=607, step=1)
```

Shape

```
df.shape
```

```
(607, 12)
```

Number of Columns and Rows

```
num_rows = len(df)
num_cols = len(df.columns)
num_rows, num_cols
```

```
(607, 12)
```

Access

head() and tail()

```
df.head()
```

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary	sa
0	0	2020	MI	FT	Data Scientist	70000	
1	1	2020	SE	FT	Machine Learning Scientist	260000	
2	2	2020	SE	FT	Big Data Engineer	85000	
3	3	2020	MI	FT	Product Data Analyst	20000	
4	4	2020	SE	FT	Machine Learning Engineer	150000	

```
df.tail()
```

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary
602	602	2022	SE	FT	Data Engineer	154000
603	603	2022	SE	FT	Data Engineer	126000
604	604	2022	SE	FT	Data Analyst	129000

```
df.tail(10)
```

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary
597	597	2022	SE	FT	Data Analyst	170000
598	598	2022	MI	FT	Data Scientist	160000
599	599	2022	MI	FT	Data Scientist	130000
600	600	2022	EN	FT	Data Analyst	67000
601	601	2022	EN	FT	Data Analyst	52000
602	602	2022	SE	FT	Data Engineer	154000
603	603	2022	SE	FT	Data Engineer	126000
604	604	2022	SE	FT	Data Analyst	129000
605	605	2022	SE	FT	Data Analyst	150000
606	606	2022	MI	FT	AI Scientist	200000

```
df.describe()
```

	Unnamed: 0	work_year	salary	salary_in_usd	remote_ratio
count	607.000000	607.000000	6.070000e+02	607.000000	607.00000
mean	303.000000	2021.405272	3.240001e+05	112297.869852	70.92257
std	175.370085	0.692133	1.544357e+06	70957.259411	40.70913
min	0.000000	2020.000000	4.000000e+03	2859.000000	0.00000
25%	151.500000	2021.000000	7.000000e+04	62726.000000	50.00000
50%	303.000000	2022.000000	1.150000e+05	101570.000000	100.00000
75%	454.500000	2022.000000	1.650000e+05	150000.000000	100.00000
max	606.000000	2022.000000	3.040000e+07	600000.000000	100.00000

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 607 entries, 0 to 606
Data columns (total 12 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   Unnamed: 0          607 non-null    int64 
 1   work_year           607 non-null    int64 
 2   experience_level    607 non-null    object
 3   employment_type     607 non-null    object
 4   job_title           607 non-null    object
 5   salary              607 non-null    int64 
 6   salary_currency     607 non-null    object
 7   salary_in_usd       607 non-null    int64 
 8   employee_residence  607 non-null    object
 9   remote_ratio        607 non-null    int64 
 10  company_location    607 non-null    object
 11  company_size        607 non-null    object
dtypes: int64(5), object(7)
memory usage: 57.0+ KB
```

Accessing Columns

```
df['salary']
```

```
    0       70000
    1      260000
    2       85000
    3       20000
    4      150000
            ...
    602    154000
    603    126000
    604    129000
    605    150000
    606    200000
    Name: salary, Length: 607, dtype: int64
```

```
type(df['salary'])
```

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

```
df.salary
```

```
    0       70000
    1      260000
    2       85000
    3       20000
    4      150000
            ...
    602    154000
    603    126000
    604    129000
    605    150000
    606    200000
    Name: salary, Length: 607, dtype: int64
```

```
type(df.salary)
```

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

```
df[['salary','remote_ratio','job_title']]
```

	salary	remote_ratio	job_title
0	70000	0	Data Scientist
1	260000	0	Machine Learning Scientist
2	85000	50	Big Data Engineer
3	20000	0	Product Data Analyst
4	150000	50	Machine Learning Engineer
...	...	...	...
602	154000	100	Data Engineer
603	126000	100	Data Engineer
604	129000	0	Data Analyst
605	150000	100	Data Analyst
606	200000	100	AI Scientist

607 rows × 3 columns

```
type(df[['salary','remote_ratio','job_title']])
```

```
pandas.core.frame.DataFrame
```

Accessing Rows

```
df.loc[0]
```

```
Unnamed: 0                          0
work_year                        2020
experience_level                   MI
employment_type                    FT
job_title              Data Scientist
salary                          70000
salary_currency                   EUR
salary_in_usd                   79833
employee_residence                 DE
remote_ratio                        0
company_location                   DE
company_size                        L
Name: 0, dtype: object
```

```
type(df.loc[0])
```

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

```
df.loc[[0,10,20]]
```

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary
0	0	2020	MI	FT	Data Scientist	70000
10	10	2020	EN	FT	Data Scientist	45000

```
type(df.loc[[0,10,20]])
```

```
pandas.core.frame.DataFrame
```

```
df.loc[0:10]
```

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary
0	0	2020	MI	FT	Data Scientist	70000
1	1	2020	SE	FT	Machine Learning Scientist	260000
2	2	2020	SE	FT	Big Data Engineer	85000
3	3	2020	MI	FT	Product Data Analyst	20000
4	4	2020	SE	FT	Machine Learning Engineer	150000
5	5	2020	EN	FT	Data Analyst	72000
6	6	2020	SE	FT	Lead Data Scientist	190000
7	7	2020	MI	FT	Data Scientist	11000000
					Business	

```
df.iloc[0]
```

```
Unnamed: 0                         0
work_year                       2020
experience_level                  MI
employment_type                   FT
job_title             Data Scientist
salary                         70000
salary_currency                  EUR
salary_in_usd                  79833
employee_residence                DE
remote_ratio                       0
company_location                  DE
company_size                       L
Name: 0, dtype: object
```

```
df.iloc[[0,10,20]]
```

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary
0	0	2020	MI	FT	Data Scientist	70000
10	10	2020	EN	FT	Data Scientist	45000

```
df.iloc[0:10]
```

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary
0	0	2020	MI	FT	Data Scientist	70000
1	1	2020	SE	FT	Machine Learning Scientist	260000
2	2	2020	SE	FT	Big Data Engineer	85000
3	3	2020	MI	FT	Product Data Analyst	20000
4	4	2020	SE	FT	Machine Learning Engineer	150000
5	5	2020	EN	FT	Data Analyst	72000

```
EID = ['EID' + str(i) for i in range(100, len(df) + 100)]
df.index = EID
df.head()
```

	Unnamed: 0	work_year	experience_level	employment_type	job_title	sala
EID100	0	2020	MI	FT	Data Scientist	700
EID101	1	2020	SE	FT	Machine Learning Scientist	2600
EID102	2	2020	SE	FT	Big Data Engineer	850

```
df.loc['EID555']
```

```
Unnamed: 0                       455
work_year                       2022
experience_level                  MI
employment_type                   FT
job_title               NLP Engineer
salary                        240000
salary_currency                  CNY
salary_in_usd                  37236
employee_residence                US
remote_ratio                      50
company_location                  US
company_size                       L
Name: EID555, dtype: object
```

Combining Row and Column Selection

```
first5rows = df.iloc[:5]
type(first5rows)
```

```
pandas.core.frame.DataFrame
```

Two Steps - Rows First

```
first5rows = df.iloc[:5]
first5rows[['salary']]
```

	salary
EID100	70000
EID101	260000
EID102	85000
EID103	20000
EID104	150000

Two Steps - Columns First

```
salary = df[['salary']]
salary.iloc[:5]
```

	salary
EID100	70000
EID101	260000
EID102	85000
EID103	20000
EID104	150000

```
s1 = df['salary']
s2 = df[['salary']]
type(s1), type(s2)
```

```
(pandas.core.series.Series, pandas.core.frame.DataFrame)
```

One Step - Rows First

```
df.iloc[:5][['salary']]
```

	salary
EID100	70000
EID101	260000
EID102	85000
EID103	20000
EID104	150000

One Step - Columns First

```
df[['salary']].iloc[:5]
```

	salary
EID100	70000
EID101	260000
EID102	85000
EID103	20000
EID104	150000

Getting a Series

```
df.iloc[:5]['salary']
```

```
EID100     70000
EID101    260000
EID102     85000
EID103     20000
EID104    150000
Name: salary, dtype: int64
```

```
df.iloc[:5].salary
```

```
EID100     70000
EID101    260000
EID102     85000
EID103     20000
EID104    150000
Name: salary, dtype: int64
```

```
df['salary'].iloc[:5]
```

```
EID100     70000
EID101    260000
EID102     85000
EID103     20000
EID104    150000
Name: salary, dtype: int64
```

```
df.salary.iloc[:5]
```

```
EID100     70000
EID101    260000
EID102     85000
EID103     20000
EID104    150000
Name: salary, dtype: int64
```