```
import numpy as np
import pandas as pd
cols=['a','b','c','d','e']
rows=['x','y','z']
data=np.round(np.random.randn(3,5),2)
df=pd.DataFrame(data,rows,cols)
print(df)
```

```
      a     b     c     d     e
x  1.92 -1.44  0.34 -0.85  0.07
y  1.43  0.29  0.77 -1.81  0.29
z -0.68 -0.28 -0.12 -0.82  0.29
```

ACCESSING COLOUMNS

```
df['a']
```

```
x    1.92
y    1.43
z   -0.68
Name: a, dtype: float64
```

```
df['b']
```

```
x   -1.44
y    0.29
z   -0.28
Name: b, dtype: float64
```

```
df['c']
```

```
x    0.34
y    0.77
```

```
z   -0.12
Name: c, dtype: float64
```

```
df[[a,b]]
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-8-5991351a8bf3> in <cell line: 1>()
----> 1 df[[a,b]]

NameError: name 'a' is not defined
```

SEARCH STACK OVERFLOW

```
df['a','b']
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
/usr/local/lib/python3.10/dist-packages/pandas/core/indexes/base.py in get_loc(self, key, method, tolerance)
   3801             try:
-> 3802                 return self._engine.get_loc(casted_key)
```

```
df[['a','b']]
```

	a	b
x	1.92	-1.44
y	1.43	0.29
z	-0.68	-0.28

```
df[['a','c']]
```

	a	c
x	1.92	0.34
y	1.43	0.77
z	-0.68	-0.12

SEARCH STACK OVERFLOW

ACCESSING ROWS

```
#df.loc['coloumn name']
df.loc['x']
```

```
a    1.92
b   -1.44
c    0.34
```

```
d   -0.85
e    0.07
Name: x, dtype: float64
```

```
df.loc['y']
```

```
a    1.43
b    0.29
c    0.77
d   -1.81
e    0.29
Name: y, dtype: float64
```

```
df.loc[['x','y']]
```

	a	b	c	d	e
x	1.92	-1.44	0.34	-0.85	0.07
y	1.43	0.29	0.77	-1.81	0.29

ACCESSING THE PARTICULAR VALUES

```
df['a']['y']
```

```
1.43
```

```
df['b']['z']
```

```
-0.28
```

```
df
```

	a	b	c	d	e
x	1.92	-1.44	0.34	-0.85	0.07
y	1.43	0.29	0.77	-1.81	0.29

```
df['a+b']=df['a']+df['b']
```

```
df['c']
```

```
x    0.34
y    0.77
z   -0.12
Name: c, dtype: float64
```

```
df['c']>0 # It prints the boolean values of a coloumn based on the condition if that value is greater than 0 then it returns 1else it returns
```

```
x     True
y     True
z    False
Name: c, dtype: bool
```

```
df[df['c']>0] # it displays all the rows of a table where c value >0
```

	a	b	c	d	e
x	1.92	-1.44	0.34	-0.85	0.07
y	1.43	0.29	0.77	-1.81	0.29

```
df.drop('a+b',axis=1)# this wont eliminate the 'a+b' coloumn perminantly it doesnt make the change in the actual dataframe
df.drop('a+b',axis=1,inplace=True)
df
```

	a	b	c	d	e
x	1.92	-1.44	0.34	-0.85	0.07
y	1.43	0.29	0.77	-1.81	0.29
z	-0.68	-0.28	-0.12	-0.82	0.29

```
df['c']>0   # it displays the
```

```
x     True
y     True
z    False
Name: c, dtype: bool
```

```
df[df['c']>0]
```

	a	b	c	d	e
x	1.92	-1.44	0.34	-0.85	0.07
y	1.43	0.29	0.77	-1.81	0.29

```
df
```

	a	b	c	d	e
x	1.92	-1.44	0.34	-0.85	0.07
y	1.43	0.29	0.77	-1.81	0.29
z	-0.68	-0.28	-0.12	-0.82	0.29

```
df[df['c']>0 & df['b']>0]
df
#it raises error because we are not placing brackets correct syntax is
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
/usr/local/lib/python3.10/dist-packages/pandas/core/ops/array_ops.py in na_logical_op(x, y, op)
    304         #  (xint or xbool) and (yint or bool)
--> 305         result = op(x, y)
    306     except TypeError:
```

8 frames

```
TypeError: ufunc 'bitwise_and' not supported for the input types, and the inputs could not be safely coerced to any supported
types according to the casting rule ''safe''

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
ValueError: Buffer dtype mismatch, expected 'Python object' but got 'double'

The above exception was the direct cause of the following exception:

TypeError                                 Traceback (most recent call last)
/usr/local/lib/python3.10/dist-packages/pandas/core/ops/array_ops.py in na_logical_op(x, y, op)
    326             ) as err:
    327                 typ = type(y).__name__
--> 328                 raise TypeError(
    329                     f"Cannot perform '{op.__name__}' with a dtyped [{x.dtype}] array "
    330                     f"and scalar of type [{typ}]"

TypeError: Cannot perform 'rand_' with a dtyped [float64] array and scalar of type [bool]
```

SEARCH STACK OVERFLOW

```
df[(df['c']>0) & (df['b']>0)] # it returns the dataframes where the coloumns c and b contains value greater than 0
```

	a	b	c	d	e
y	1.43	0.29	0.77	-1.81	0.29

creating another temporary dataframe

```
data=np.round(np.random.randn(3,5),2)
cf=pd.DataFrame(data,rows,cols)
print(cf)
```

```
      a     b     c     d     e
x -1.46  0.43 -0.75 -0.60  0.10
y -1.08 -0.21 -0.63  0.60 -0.14
z -0.42 -0.85  1.59  1.54 -0.53
```

Reseting indexes to 0 's and 1 ' s

```
cf.reset_index()
```

	index	a	b	c	d	e
0	x	-1.46	0.43	-0.75	-0.60	0.10
1	y	-1.08	-0.21	-0.63	0.60	-0.14
2	z	-0.42	-0.85	1.59	1.54	-0.53

```
cf.set_index('a')
```

	b	c	d	e
a				
-1.46	0.43	-0.75	-0.60	0.10
-1.08	-0.21	-0.63	0.60	-0.14
-0.42	-0.85	1.59	1.54	-0.53

```
cf.columns
```

```
Index(['a', 'b', 'c', 'd', 'e'], dtype='object')
```

```
kf=cf
kf
```

	a	b	c	d	e
x	-1.46	0.43	-0.75	-0.60	0.10
y	-1.08	-0.21	-0.63	0.60	-0.14
z	-0.42	-0.85	1.59	1.54	-0.53

```
kf.columns
```

```
Index(['a', 'b', 'c', 'd', 'e'], dtype='object')
```

```
kf.columns=[1,2,3,4,5]
```

```
kf.columns
```

```
Int64Index([1, 2, 3, 4, 5], dtype='int64')
```

✓ 0s completed at 10:00 PM ● ✕