



PYTHON

practice



LISTS



lists

```
In [1]: M = ['sravani','akshesha','anwesha']
        L = [1, 2, 2, 45]
        N = [1.2, 3.4567, 6.7]
        A = [1, 2.3, 'SRAVANI', 5+9j]
        print(M)
        print(L)
        print(N)
        print(A)

['sravani', 'akshesha', 'anwesha']
[1, 2, 2, 45]
[1.2, 3.4567, 6.7]
[1, 2.3, 'SRAVANI', (5+9j)]
```

```
In [2]: List = []
        print("blank list")
        print(List)
        List = [1, 2, 3, 4, 5, 1, 3, 7, 9, 0]
        print('num list')
        print(List)

blank list
[]
num list
[1, 2, 3, 4, 5, 1, 3, 7, 9, 0]
```

```
In [3]: print(len(List))
        print(len(M))
        print(len(N))
        print(len(L))
        print(len(A))
```

```
10  
3  
3  
4  
4
```

```
In [4]: List.append(1)  
List.append(100)  
List.append(200)  
List.append(35)  
print(List)  
  
[1, 2, 3, 4, 5, 1, 3, 7, 9, 0, 1, 100, 200, 35]
```

```
In [5]: for i in range(1,10):  
        List.append(i)  
        print(List)  
  
[1, 2, 3, 4, 5, 1, 3, 7, 9, 0, 1, 100, 200, 35, 1, 2, 3, 4, 5, 6, 7, 8,  
9]
```

```
In [6]: L.insert(3, 'karthik')  
print(L)  
  
[1, 2, 2, 'karthik', 45]
```

```
In [7]: M.extend([3, 4+5j, "skates"])  
print(M)  
  
['sravani', 'akshesha', 'anwesha', 3, (4+5j), 'skates']
```

```
In [8]: ### ACCESSING AN ELEMENT ###  
print(A[1])  
  
2.3
```

```
In [9]: ### NEGATIVE INDEXING ###  
print(N[-1])
```

6.7

```
In [10]: List.remove(100)  
print(List)
```

```
[1, 2, 3, 4, 5, 1, 3, 7, 9, 0, 1, 200, 35, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
In [11]: List.pop()  
print(List)
```

```
[1, 2, 3, 4, 5, 1, 3, 7, 9, 0, 1, 200, 35, 1, 2, 3, 4, 5, 6, 7, 8]
```

```
In [12]: list1 = List[:5]  
print(list1)
```

```
[1, 2, 3, 4, 5]
```

```
In [13]: list2 = List[3:]  
print(list2)
```

```
[4, 5, 1, 3, 7, 9, 0, 1, 200, 35, 1, 2, 3, 4, 5, 6, 7, 8]
```

```
In [15]: list3 = List[:]  
print(list3)
```

```
[1, 2, 3, 4, 5, 1, 3, 7, 9, 0, 1, 200, 35, 1, 2, 3, 4, 5, 6, 7, 8]
```

```
In [16]: list4 = List[:-5]  
print(list4)
```

```
[1, 2, 3, 4, 5, 1, 3, 7, 9, 0, 1, 200, 35, 1, 2, 3]
```

```
In [17]: list5 = List[-10:-5]  
print(list5)
```

```
[200, 35, 1, 2, 3]
```

```
In [18]: M.clear()  
print(M)
```

```
[]
```

```
In [22]: N.index(1.2)
```

```
Out[22]: 0
```

```
In [24]: N.count(6.7)
```

```
Out[24]: 1
```

```
In [26]: N.sort()  
print(N)
```

```
[1.2, 3.4567, 6.7]
```

```
In [27]: N.reverse()  
print(N)
```

```
[6.7, 3.4567, 1.2]
```

```
In [28]: N.copy()
```

```
Out[28]: [6.7, 3.4567, 1.2]
```

```
In [32]: sum(List)
```

```
Out[32]: 307
```

```
In [37]: AN = ('S')  
ord(AN)
```

```
Out[37]: 83
```

```
In [41]: max(List)
```

Out[41]: 200

In [42]: `min(List)`

Out[42]: 0

In [43]: `all(List)`

Out[43]: False

In [44]: `len(List)`

Out[44]: 21

In [45]: `enumerate(List)`

Out[45]: <enumerate at 0x20d0342f3f0>

In [49]: `filter(List, A)`

Out[49]: <filter at 0x20d03430160>

TUPLES



Tuples

```
In [1]: tuple1 = ()  
print(tuple1)  
  
()
```

```
In [2]: tuple1 = ("kendriya", "vidyalaya")  
print(tuple1)  
  
('kendriya', 'vidyalaya')
```

```
In [3]: list1 = [1,2,3,4]  
tuple2 = (tuple1, list1)  
print(tuple2)  
  
(('kendriya', 'vidyalaya'), [1, 2, 3, 4])
```

```
In [4]: tuple2 = tuple1*2  
print(tuple2)  
  
('kendriya', 'vidyalaya', 'kendriya', 'vidyalaya')
```

```
In [5]: ### tuple using loop ###  
tuple3 = (1,2)  
n = 3  
for i in range (int(n)):  
    tuple3 = (tuple3,)  
    print(tuple3)  
  
((1, 2),)  
(((1, 2),),)  
((((1, 2),),),)
```

```
In [6]: print(tuple1[1])
```

```
vidyalaya
```

```
In [12]: tuple5 = tuple("sravani")
print(tuple5)
print(tuple5[1:])
print(tuple5[::-2])
print(tuple5[3:5])
```

```
('s', 'r', 'a', 'v', 'a', 'n', 'i')
('r', 'a', 'v', 'a', 'n', 'i')
('i', 'a', 'a', 's')
('v', 'a')
```

```
In [15]: all(tuple1)
```

```
Out[15]: True
```

```
In [16]: any(tuple1)
```

```
Out[16]: True
```

```
In [17]: len(tuple1)
```

```
Out[17]: 2
```

```
In [18]: enumerate(tuple1)
```

```
Out[18]: <enumerate at 0x26798695480>
```

```
In [19]: max(tuple1)
```

```
Out[19]: 'vidyalaya'
```

```
In [20]: min(tuple1)
```

```
Out[20]: 'kendriya'
```

ARRAYS



```
In [42]: import array as arr
```

```
In [43]: a = arr.array('i', [1, 2, 3, 4])  
         for i in range (0,4):  
             print(a[i], end = " ")  
1 2 3 4
```

```
In [44]: a.insert(4, 5)  
         for i in range (0,5):  
             print(a[i], end = " ")  
1 2 3 4 5
```

```
In [45]: b = arr.array('d', [1.1, 2.2, 3.3])  
         b.append(4.4)  
         for i in (b):  
             print(i)  
         print()  
1.1  
2.2  
3.3  
4.4
```

```
In [46]: print(a[1])  
2
```

```
In [47]: print(a.pop(2))  
3
```

```
In [48]: for i in range (0, 4):  
         print(a[i], end = " ")
```

1 2 4 5

```
In [49]: a.remove(2)  
         for i in range(0, 3):  
             print(a[i], end = " ")
```

1 4 5

```
In [51]: sliced_array = b[1:3]  
         print(sliced_array)
```

array('d', [2.2, 3.3])

```
In [52]: print(a.index(4))
```

1

```
In [54]: a[2] = 7  
         for i in range (0, 3):  
             print(a[i])
```

1
4
7

DICTIONARIES



```
In [1]: Dict = {1: 'sravani', 2: 'karthik', 3: ' akshesha', 4: 'akshay', 5: 'aishwitha'}  
print(Dict)
```

```
{1: 'sravani', 2: 'karthik', 3: ' akshesha', 4: 'akshay', 5: 'aishwitha'}
```

```
In [2]: t = {}  
print(t)
```

```
{}
```

```
In [3]: t = dict({1: 'sravani', 2: 'karthik', 3: ' akshesha'})  
print(t)
```

```
{1: 'sravani', 2: 'karthik', 3: ' akshesha'}
```

```
In [4]: s = dict([(1, 'akshay'), (2, 'aishwitha')])  
print(s)
```

```
{1: 'akshay', 2: 'aishwitha'}
```

```
In [5]: a = {1: 'cadbury', 2: '5star', 3: {'x': 'asha', 'y': 'mangobite'}}  
print(a)
```

```
{1: 'cadbury', 2: '5star', 3: {'x': 'asha', 'y': 'mangobite'}}
```

```
In [6]: lst = {}  
print(lst)  
lst[0] = 'sudden'  
lst[1] = 'change'  
print(lst)  
lst['value_set'] = 1, 2, 3, 4  
print(lst)
```

```
lst[1] = 'rain'
print(lst)

{}
{0: 'sudden', 1: 'change'}
{0: 'sudden', 1: 'change', 'value_set': (1, 2, 3, 4)}
{0: 'sudden', 1: 'rain', 'value_set': (1, 2, 3, 4)}
```

```
In [7]: print(Dict[1])
```

sravani

```
In [8]: print(Dict.get(4))
```

akshay

```
In [9]: z = {'A': {1: 'LFJC'}, 'B': {2: 'KV'}}
print(z['A'][1])
```

LFJC

```
In [10]: del Dict[5]
print(Dict)
del z['A'][1]
print(z)
```

{1: 'sravani', 2: 'karthik', 3: 'akshesha', 4: 'akshay'}
{'A': {}, 'B': {2: 'KV'}}

```
In [11]: oop = Dict.pop(3)
print(oop)
```

akshesha

```
In [12]: print(Dict)
```

{1: 'sravani', 2: 'karthik', 4: 'akshay'}


```
In [13]: ppt = Dict.popitem()
         print(ppt)
         print(Dict)
         (4, 'akshay')
         {1: 'sravani', 2: 'karthik'}
```

```
In [14]: Dict.clear()
         print(Dict)
         {}
```

FUNCTIONS



```
In [8]: def evenOdd(x):  
        if (x % 2 == 0):  
            print("even")  
        else:  
            print("odd")  
evenOdd(5)  
evenOdd(8)
```

```
odd  
even
```

```
In [9]: def math(x):  
        x[1] = 15  
        lst = [1, 5, 10, 20, 25]  
        math(lst);  
        print(lst)
```

```
[1, 15, 10, 20, 25]
```

```
In [10]: def sra(x, y = 15):  
          print("x:", x)  
          print("y", y)
```

```
sra(20)
```

```
x: 20  
y 15
```

```
In [14]: def student(firstname, lastname):  
          print(firstname, lastname)  
          student(firstname = 'sravani', lastname = 'mahankali')  
          student(firstname = 'akshesha', lastname = 'oraon')  
          student(lastname = 'kurumeti', firstname = 'akshay')
```

```
sravani mahankali  
akshesha oraon
```

akshay kurumeti

```
In [15]: cube = lambda x: x*x*x  
print(cube(15))
```

3375

```
In [17]: import math  
a = 5.6  
print(math.ceil(a)) ### smallest integral value ###  
print(math.floor(a)) ### greatest integral value ###
```

6

5

```
In [20]: x = -25  
y = 10  
print(math.fabs(x)) ### ABSOLUTE VALUE ###  
print(math.factorial(y)) ### factorial of number ###
```

25.0

3628800

```
In [21]: p = - 10  
q = 5  
r = -25  
s = 16  
print(math.copysign(p, q)) ### value of a with sign of b ###
```

10.0

```
In [22]: print(math.gcd(r, s)) ### greatest common divisor of two numbers ###
```

1

```
In [23]: print(math.exp(5)) ### exponential ###  
print(math.log(3, 5)) ### log 3 with base 5 ###
```

```
148.4131591025766  
0.6826061944859854
```

```
In [24]: print(math.log2(5)) ### log of 5 with base 2 ###  
        print(math.log10(5)) ### log of 5 with base 10 ###
```

```
2.321928094887362  
0.6989700043360189
```

```
In [25]: print(math.pow(7, 8)) ### 7 raised to the power 8 ###  
        print(math.sqrt(80)) ### square root of 80 ###
```

```
5764801.0  
8.944271909999916
```

```
In [27]: a = math.pi/3  
        print(math.sin(a))  
        print(math.cos(a))  
        print(math.tan(a))
```

```
0.8660254037844386  
0.5000000000000001  
1.7320508075688767
```

```
In [28]: b = 3  
        c = 5  
        print(math.hypot(b, c)) ### hypotenuse ###
```

```
5.830951894845301
```

```
In [29]: print(math.degrees(a))
```

```
59.99999999999999
```

```
In [30]: d = 90  
        print(math.radians(d))
```

```
In [31]: print(math.gamma(b))
```

```
2.0
```

```
In [32]: print(math.pi)  
print(math.e)
```

```
3.141592653589793  
2.718281828459045
```

```
In [34]: math.isinf(90)
```

```
Out[34]: False
```

```
In [37]: math.isnan(56)
```

```
Out[37]: False
```

```
In [39]: def me(*a):  
          for i in a:  
              print(i)  
me("my", "name", "is", "sravani")
```

```
my  
name  
is  
sravani
```

IF ELSE



```
In [2]: i = 80
if (i > 100):
    print('true')
print('false')
```

false

```
In [3]: i = 15
if (i < 20):
    print('true')
else:
    print('false')
```

true

```
In [4]: i = 100
if (i == 100):
    if (i < 200):
        print("i is smaller")
    else:
        print("i is greater")
```

i is smaller

```
In [5]: i = 50
if (i == 100):
    print(100)
elif (i == 200):
    print(200)
elif (i == 50):
    print(50)
else:
    print('false')
```

50

LOOPS



```
In [1]: count = 5
        while (count<10):
            count = count+2
            print("yes")
```

```
yes
yes
yes
```

```
In [3]: count = 0
        while (count <= 5):
            count = count+1
            print("no")
        else:
            print("yes")
```

```
no
no
no
no
no
no
yes
```

```
In [4]: ### iteration ###
        l = ["sravani", "is", "a", "good", "girl"]
        for i in l:
            print(i)
```

```
sravani
is
a
good
girl
```

```
In [5]: t = (1, 2, 3, 4, 5)
        for i in t:
            print(i)
```

```
1
2
3
4
5
```

```
In [7]: s = "little flower"
        for i in s:
            print(i)
```

```
l
i
t
t
l
e

f
l
o
w
e
r
```

```
In [13]: d = dict()
          d[1] = 'sravani'
          d[2] = 'akshay'
          for i in d:
              print((i, d[i]))
```

```
(1, 'sravani')
(2, 'akshay')
```

```
In [16]: lst = [1, 2, 3, 4]
          for index in range(len(lst)):
```

```
print(lst[index])
```

```
1  
2  
3  
4
```

```
In [18]: for i in range (1, 10):  
         for j in range(i):  
             print(i, end=' ')  
         print()
```

```
1  
2 2  
3 3 3  
4 4 4 4  
5 5 5 5 5  
6 6 6 6 6 6  
7 7 7 7 7 7 7  
8 8 8 8 8 8 8 8  
9 9 9 9 9 9 9 9 9
```

```
In [20]: for letter in 'kendriyavidyalaya':  
         if letter == 's' or letter == 'a':  
             continue  
         print("letter:", letter)
```

```
letter: k  
letter: e  
letter: n  
letter: d  
letter: r  
letter: i  
letter: y  
letter: v  
letter: i  
letter: d  
letter: y
```

```
letter: l  
letter: y
```

```
In [21]: for letter in 'kendriyavidyalaya':  
         if letter == 's' or letter == 'a':  
             break  
         print("letter:", letter)
```

```
letter: k  
letter: e  
letter: n  
letter: d  
letter: r  
letter: i  
letter: y
```

```
In [25]: for letter in 'kendriyavidyalaya':  
         pass  
         print("letter:", letter)
```

```
letter: a
```

```
In [ ]:
```

SETS



```
In [1]: set1 = {}  
print(set1)  
  
{}
```

```
In [3]: set2 = set("karthik")  
print(set2)  
  
{ 'r', 't', 'h', 'k', 'a', 'i' }
```

```
In [4]: set3 = set(["zindagi", "na", "milegi", "dobara"])  
print(set3)  
  
{ 'na', 'zindagi', 'milegi', 'dobara' }
```

```
In [1]: set4 = {1, 2, 3, 4, 5, 6}  
print(set4)  
  
{1, 2, 3, 4, 5, 6}
```

```
In [2]: set4.add(7)  
set4.add(8)  
print(set4)  
  
{1, 2, 3, 4, 5, 6, 7, 8}
```

```
In [4]: for i in range(1,25):  
        set4.add(i)  
print(set4)  
  
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,  
21, 22, 23, 24}
```

```
In [3]: set5 = set([6, 5, (11, 7), 9])  
set5.update([1, 2])
```

```
print(set5)  
{(11, 7), 1, 2, 5, 6, 9}
```

```
In [8]: for x in set5:  
        print(x)
```

```
(11, 7)  
1  
2  
5  
6  
9
```

```
In [9]: set5.remove(1)  
print(set5)
```

```
{(11, 7), 2, 5, 6, 9}
```

```
In [10]: set5.discard(2)  
print(set5)
```

```
{(11, 7), 5, 6, 9}
```

```
In [11]: for i in range(5, 7):  
        set5.remove(i)  
print(set5)
```

```
{(11, 7), 9}
```

```
In [12]: set5.pop()  
print(set5)
```

```
{9}
```

```
In [13]: set5.clear()  
print(set5)
```



```
set()
```

```
In [15]: digit = {1, 2, 3, 4, 5, 6, 7, 8, 9}
digit.add(10)
print(digit)
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

```
In [16]: ### if the set is frozen we cannot add elements ###
frozen_digit = frozenset(digit)
frozen_digit.add(11)
```

```
-----
AttributeError
```

```
Traceback (most recent call l
```

```
ast)
```

```
<ipython-input-16-b79b7fe47a1d> in <module>
```

```
1 frozen_digit = frozenset(digit)
```

```
----> 2 frozen_digit.add(11)
```

```
AttributeError: 'frozenset' object has no attribute 'add'
```

STRINGS



```
In [34]: string1 = 'my name is sravani'
string2 = "I believe that life brings surprises when needed"
string3 = '''accepting things and changing the way we think is a challenge'''
print(string1)
print(string2)
print(string3)
```

```
my name is sravani
I believe that life brings surprises when needed
accepting things and changing the way we think is a challenge
```

```
In [35]: print(string1[4])
print(string2[-2])
```

```
a
e
```

```
In [36]: print(string1[2:9])
print(string2[-5:-2])
```

```
name i
eed
```

```
In [37]: ### updating a string ###
string1 = "i am sravani"
print(string1)
```

```
i am sravani
```

```
In [38]: print(string1)
print(string2)
print(string3)
```

I believe that life brings surprises when needed
accepting things and changing the way we think is a challenge

```
In [41]: ### only entire string can be deleted ###  
del(string1)  
print(string1)
```

```
-----  
-----  
NameError                                Traceback (most recent call l  
ast)  
<ipython-input-41-89edbd8d570a> in <module>  
    1 ### only entire string can be deleted ###  
----> 2 del(string1)  
    3 print(string1)  
  
NameError: name 'string1' is not defined
```

```
In [42]: ### escape sequencing ###  
string4 = '''i'm pursuing business analytics'''  
string5 = 'i\'m a kind person'  
string6 = "it has a name\"pen\""  
string7 = "c:\\drive\\photos"  
print(string4)  
print(string5)  
print(string6)  
print(string7)
```

```
i'm pursuing business analytics  
i'm a kind person  
it has a name"pen"  
c:\drive\photos
```

```
In [43]: ### formatting of strings ###
```

```
In [44]: string8 = "{} {} {} {}".format('sravani', 'is', 'an', 'analyst')  
print(string8)
```

```
sravani is an analyst
```

```
In [45]: string8 = "{0} {2} {3} {1}".format('sravani', 'is', 'an', 'analyst')
print(string8)
```

```
sravani an analyst is
```

```
In [46]: string8 = "{a} {b} {c} {d}".format(a='sravani', b='is', c='an', d='analyst')
print(string8)
```

```
sravani is an analyst
```

```
In [47]: string9 = "{0:b}".format(18)
print(string9)
string9 = "{0:e}".format(200.6879)
print(string9)
string9 = "{0:.3f}".format(22/56)
print(string9)
```

```
10010
2.006879e+02
0.393
```

```
In [48]: string10 = "|{:>10}|{: ^5}|{: ^5}|{:<10}|".format('sravani', 'is', 'an', 'analyst')
print(string10)
```

```
| sravani| is | an |analyst |
```

```
In [52]: int1 = 89.36874521
print('value is %.2f' %int1)
print('value is %.4f' %int1)
```

```
value is 89.37
value is 89.3687
```

DATA FRAMES



```
Position      C
Age           27
Height        6-11
Weight        265
College       Texas A&M
Salary        1.9689e+07
Name: 98, dtype: object
```

```
In [26]: t = pd.read_csv(r'D:\nba.csv', index_col = "Name")
```

```
second = t.loc["Avery Bradley"]
print(second)
```

```
Team      Boston Celtics
Number      0
Position   PG
Age        25
Height     6-2
Weight     180
College    Texas
Salary     7.73034e+06
Name: Avery Bradley, dtype: object
```

```
In [29]: q = pd.read_csv(r'D:\nba.csv', index_col = "Name")
```

```
third = q["Age"]
print(third)
```

```
Name
Avery Bradley      25.0
Jae Crowder        25.0
John Holland       27.0
R.J. Hunter        22.0
Jonas Jerebko      29.0
Amir Johnson       29.0
Jordan Mickey      21.0
Kelly Olynyk       25.0
Terry Rozier       22.0
Marcus Smart       22.0
```

```
In [9]: import pandas as pd
df = pd.DataFrame()
print(df)
```

Empty DataFrame
Columns: []
Index: []

```
In [10]: s = ['sravani', 1, 2, 3.3]
df = pd.DataFrame(s)
print(df)
```

	0
0	sravani
1	1
2	2
3	3.3

```
In [11]: data = {'name':["sravani", "akshesha", "karthik", "akshay"], 'age':[21,
22, 29, 22]}
df = pd.DataFrame(data)
print(df)
```

	name	age
0	sravani	21
1	akshesha	22
2	karthik	29
3	akshay	22

```
In [30]: s = pd.read_csv(r'D:\nba.csv')
first = s.iloc[98]
print(first)
```

Name	DeAndre Jordan
Team	Los Angeles Clippers
Number	6

Jared Sullinger	24.0
Isaiah Thomas	27.0
Evan Turner	27.0
James Young	20.0
Tyler Zeller	26.0
Bojan Bogdanovic	27.0
Markel Brown	24.0
Wayne Ellington	28.0
Rondae Hollis-Jefferson	21.0
Jarrett Jack	32.0
Sergey Karasev	22.0
Sean Kilpatrick	26.0
Shane Larkin	23.0
Brook Lopez	28.0
Chris McCullough	21.0
Willie Reed	26.0
Thomas Robinson	25.0
Henry Sims	26.0
Donald Sloan	28.0
Thaddeus Young	27.0
...	
Al-Farouq Aminu	25.0
Pat Connaughton	23.0
Allen Crabbe	24.0
Ed Davis	27.0
Maurice Harkless	23.0
Gerald Henderson	28.0
Chris Kaman	34.0
Meyers Leonard	24.0
Damian Lillard	25.0
C.J. McCollum	24.0
Luis Montero	23.0
Mason Plumlee	26.0
Brian Roberts	30.0
Noah Vonleh	20.0
Trevor Booker	28.0
Trey Burke	23.0
Alec Burks	24.0
Dante Exum	20.0

```

Derrick Favors      24.0
Rudy Gobert         23.0
Gordon Hayward      26.0
Rodney Hood         23.0
Joe Ingles          28.0
Chris Johnson       26.0
Trey Lyles          20.0
Shelvin Mack        26.0
Raul Neto           24.0
Tibor Pleiss        26.0
Jeff Withey         26.0
NaN                NaN
Name: Age, Length: 458, dtype: float64

```

```

In [32]: ### working with missing data ###
import pandas as pd
import numpy as np
g = {'first score':[5, 6, 9, np.nan],
      'second score':[96, np.nan, 85, 78],
      'third score': [75, 44, np.nan, 326]}
df = pd.DataFrame(g)
df.isnull()

```

```

Out[32]:

```

	first score	second score	third score
0	False	False	False
1	False	True	False
2	False	False	True
3	True	False	False

```

In [33]: df.fillna(0) ### filling missing values with 0 ###

```

```

Out[33]:

```

	first score	second score	third score
0	5.0	96.0	75.0

	first score	second score	third score
1	6.0	0.0	44.0
2	9.0	85.0	0.0
3	0.0	78.0	326.0

In [38]: `df.dropna() ### dropping missing values ###`

Out[38]:

	first score	second score	third score
0	5.0	96.0	75.0

In [40]: `import pandas as pd`

```
h = {'name': ["sravani", "akshay", "akshesha", "harisha"],
      'degree': ["b pharm", "b tech", " b Tech", "bds"],
      'score': [90, 40, 80, 98]}
```

```
df = pd.DataFrame(h)
```

```
# iterating over rows using iterrows() function
```

```
for i, j in df.iterrows():
    print(i, j)
    print()
```

```
0 name      sravani
  degree    b pharm
  score         90
Name: 0, dtype: object
```

```
1 name      akshay
  degree    b tech
  score         40
Name: 1, dtype: object
```

```
2 name      akshesha
  degree    b Tech
```

```
score      80  
Name: 2, dtype: object
```

```
3 name      harisha  
degree      bds  
score      98  
Name: 3, dtype: object
```

```
In [42]: columns = list(df)  
for i in columns:  
    print (df)
```

```
   name degree score  
0  sravani  b pharm   90  
1   akshay   b tech   40  
2 akshesha  b Tech   80  
3   harisha    bds   98  
   name degree score  
0  sravani  b pharm   90  
1   akshay   b tech   40  
2 akshesha  b Tech   80  
3   harisha    bds   98  
   name degree score  
0  sravani  b pharm   90  
1   akshay   b tech   40  
2 akshesha  b Tech   80  
3   harisha    bds   98
```

THANK YOU

