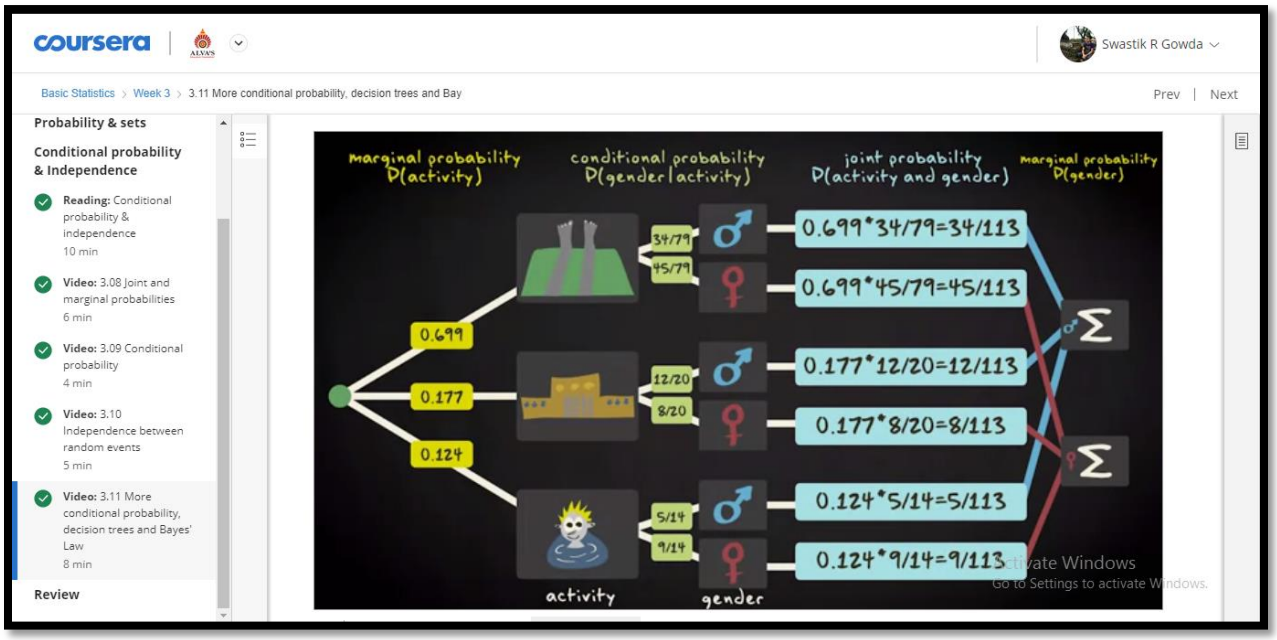
DAILY ASSESSMENT

Date:	22-July-2020	Name:	Swastik R Gowda
Course:	Basic Statistics	USN:	4AL17EC091
Topic:	Week 3	Semester & Section:	6th Sem 'B' Sec
Github Repository:	swastik-gowda		

FORENOON SESSION DETAILS

Image of session



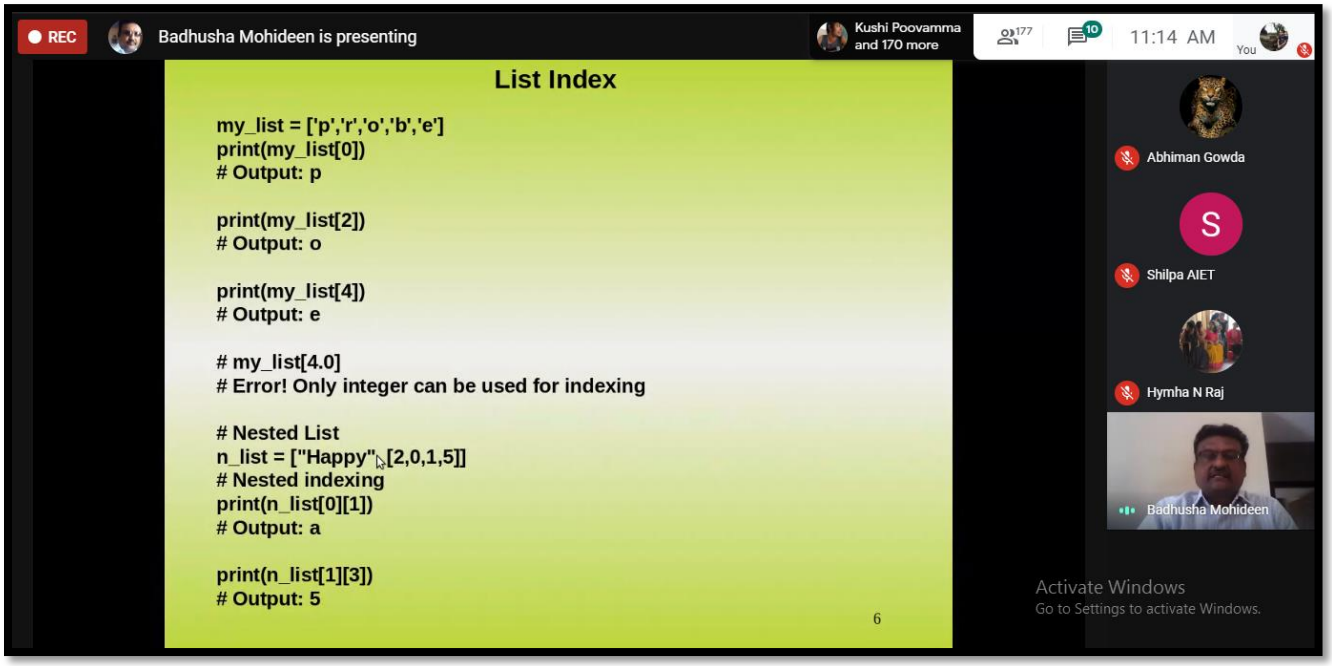
Report – Report can be typed or hand written for up to two pages.

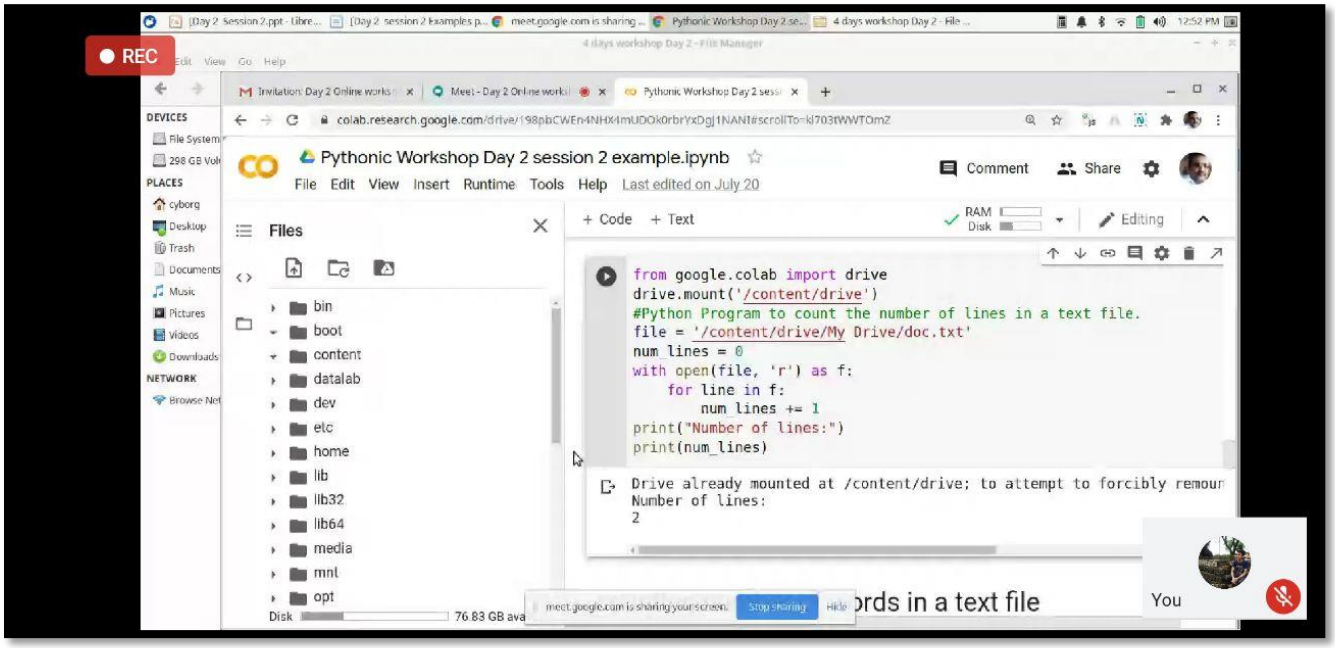
- Recognizing and understanding randomness as well as the ability to reason about it are important skills, not only to apply statistical analysis but also to make sense of things happening around us every day. Here, I will explain that humans are pathologically bad at dealing with randomness. I will use an example along the way.
- Play video starting at 29 seconds and follow transcript
- Imagine you're on a beach watching the waves roll in.
- Play video starting at 33 seconds and follow transcript
- And then your attention is caught by a beautiful shell which is distinctive in shape and larger than its neighboring shells. So you start to search for another one. This will be an unpredictable enterprise. The shells may be distributed at random at this huge beach. Hence, the time it will take you to find another will be uncertain. You may not be able to find a similar at all.
- Play video starting at 1 minute 0 seconds and follow transcript
- The more you think about it, the more you'll realize that randomness is pervasive in everyday life. It is therefore not surprising that we have a rich vocabulary to communicate it,
- Play video starting at 1 minute 12 seconds and follow transcript
- With terms like uncertainty, chance, risk and likelihood.
- Play video starting at 1 minute 17 seconds and follow transcript
- Also, degrees of variability and uncertainty can expressed quite subtly. Consider for example this sequence. Rarely. Seldom. Sometimes. Common. Frequent. Often.
- Play video starting at 1 minute 32 seconds and follow transcript
- Importantly, whether something is random is not just a property of that phenomenon, but very much also a consequence of our knowledge about it.
- Play video starting at 1 minute 43 seconds and follow transcript
- If you'd have been at this beach before, you might have spotted this special shell previously, so that this may change your search strategy and increase your chance of finding more of them.
- Play video starting at 1 minute 53 seconds and follow transcript
- Also the scale of your search matters. While you may not be very certain about finding another shell within a few minutes at a short stretch of beach, the chance to find one when you take a bit more time and cover a larger stretch of beach will increase.
- Play video starting at 2 minutes 11 seconds and follow transcript
- But in spite of many words, our ability to memorize in our daily experience with randomness, we are not very good at assessing it quantitatively at all.
- Play video starting at 2 minutes 23 seconds and follow transcript
- On one hand, we see all sorts of patterns in what is really random data. There's a word for it, apothecia.
- Play video starting at 2 minutes 31 seconds and follow transcript
- And on the other hand, we are unable to make up random data ourselves.

Date:	22-July-2020	Name:	Swastik R Gowda
Course:	How to develop Pythonic coding – Logic Perspective	USN:	4AL17EC091
Topic:	Day 2	Semester & Section:	6th Sem 'B' Sec

AFTERNOON SESSION DETAILS

Image of session





Report – Report can be typed or hand written for up to two pages.

Program to find the largest number in a list.

In [1]:
```
a=[]
n=int(input("Enter number of elements:"))
for i in range(1,n+1):
    b=int(input("Enter element:"))
    a.append(b)
a.sort()
print("Largest element is:",a[n-1])
```

```
Enter number of elements:3
Enter element:9
Enter element:6
Enter element:3
Largest element is: 9
```

Try this More pythonic 2 lines program equivalent to above codings as shown below !

In [3]:
```
x=0
print('The greatest no is',max([int(input(x)) for _ in range(int(input("Enter num: ")))]))
```

```
Enter num: 3
9
6
3
The greatest no is 9
```

Python Program to put the even and odd elements in a list into two different lists.

In [4]:
```
a=[]
n=int(input("Enter number of elements:"))
for i in range(1,n+1):
    b=int(input("Enter element:"))
    a.append(b)
even=[]
odd=[]
for j in a:
    if(j%2==0):
        even.append(j)
    else:
        odd.append(j)
print("The even list",even)
print("The odd list",odd)
```

```
Enter number of elements:3
Enter element:1
Enter element:2
Enter element:3
The even list [2]
The odd list [1, 3]
```

Instead of forementioned 14 lines program, the equivalent 4 lines !

Pythonic program is here ....

In [6]:
```
x=0
l=[int(input(x)) for _ in range(int(input("Enter n: ")))]
print('even list is',[ i for i in l if i%2])
print('odd list is',[i for i in l if not i%2])
```

```
Enter n: 3
1
2
3
even list is [1, 3]
```

Print multiples of 3 upto 200 using a list

In [1]:
```
a=[]
c=1
for i in range(0,200):
    m=3*c
    if m<200:
        a.append(m)
        c=c+1
print(a,end="")
```

```
[3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57, 60, 63, 66, 69, 72, 75, 78, 81, 84, 87, 90, 93, 96, 99, 102, 10
5, 108, 111, 114, 117, 120, 123, 126, 129, 132, 135, 138, 141, 144, 147, 150, 153, 156, 159, 162, 165, 168, 171, 174, 177, 180, 183, 186,
189, 192, 195, 198]
```

In [2]:
```
#The above program can also be written using numpy
import numpy as np
x = np.arange(1,200)
n= x[(x % 3 == 0)]
print(n[:200])
```

```
[  3   6   9  12  15  18  21  24  27  30  33  36  39  42  45  48  51  54
  57  60  63  66  69  72  75  78  81  84  87  90  93  96  99 102 105 108
 111 114 117 120 123 126 129 132 135 138 141 144 147 150 153 156 159 162
 165 168 171 174 177 180 183 186 189 192 195 198]
```

print the nos only divisable by 5 and 7 between 1000 and 2000 using a list(both inclusive)

In [3]:
```
a=[]
b=[]
for i in range(1000,2000):
    if i%5==0:
        a.append(i)
    if i%7==0:
        b.append(i)
print("The numbers which are divisible by 5 are:",a,end="")
print("\n")
print("The numbers which are divisible by 7 are:",b,end="")
```

```
The numbers which are divisible by 5 are: [1000, 1005, 1010, 1015, 1020, 1025, 1030, 1035, 1040, 1045, 1050, 1055, 1060, 1065, 1070, 107
5, 1080, 1085, 1090, 1095, 1100, 1105, 1110, 1115, 1120, 1125, 1130, 1135, 1140, 1145, 1150, 1155, 1160, 1165, 1170, 1175, 1180, 1185, 11
90, 1195, 1200, 1205, 1210, 1215, 1220, 1225, 1230, 1235, 1240, 1245, 1250, 1255, 1260, 1265, 1270, 1275, 1280, 1285, 1290, 1295, 1300, 1
305, 1310, 1315, 1320, 1325, 1330, 1335, 1340, 1345, 1350, 1355, 1360, 1365, 1370, 1375, 1380, 1385, 1390, 1395, 1400, 1405, 1410, 1415,
1420, 1425, 1430, 1435, 1440, 1445, 1450, 1455, 1460, 1465, 1470, 1475, 1480, 1485, 1490, 1495, 1500, 1505, 1510, 1515, 1520, 1525, 1530,
1535, 1540, 1545, 1550, 1555, 1560, 1565, 1570, 1575, 1580, 1585, 1590, 1595, 1600, 1605, 1610, 1615, 1620, 1625, 1630, 1635, 1640, 1645,
1650, 1655, 1660, 1665, 1670, 1675, 1680, 1685, 1690, 1695, 1700, 1705, 1710, 1715, 1720, 1725, 1730, 1735, 1740, 1745, 1750, 1755, 1760,
1765, 1770, 1775, 1780, 1785, 1790, 1795, 1800, 1805, 1810, 1815, 1820, 1825, 1830, 1835, 1840, 1845, 1850, 1855, 1860, 1865, 1870, 1875,
1880, 1885, 1890, 1895, 1900, 1905, 1910, 1915, 1920, 1925, 1930, 1935, 1940, 1945, 1950, 1955, 1960, 1965, 1970, 1975, 1980, 1985, 1990,
1995]

The numbers which are divisible by 7 are: [1001, 1008, 1015, 1022, 1029, 1036, 1043, 1050, 1057, 1064, 1071, 1078, 1085, 1092, 1099, 110
6, 1113, 1120, 1127, 1134, 1141, 1148, 1155, 1162, 1169, 1176, 1183, 1190, 1197, 1204, 1211, 1218, 1225, 1232, 1239, 1246, 1253, 1260, 12
67, 1274, 1281, 1288, 1295, 1302, 1309, 1316, 1323, 1330, 1337, 1344, 1351, 1358, 1365, 1372, 1379, 1386, 1393, 1400, 1407, 1414, 1421, 1
428, 1435, 1442, 1449, 1456, 1463, 1470, 1477, 1484, 1491, 1498, 1505, 1512, 1519, 1526, 1533, 1540, 1547, 1554, 1561, 1568, 1575, 1582,
1589, 1596, 1603, 1610, 1617, 1624, 1631, 1638, 1645, 1652, 1659, 1666, 1673, 1680, 1687, 1694, 1701, 1708, 1715, 1722, 1729, 1736, 1743,
1750, 1757, 1764, 1771, 1778, 1785, 1792, 1799, 1806, 1813, 1820, 1827, 1834, 1841, 1848, 1855, 1862, 1869, 1876, 1883, 1890, 1897, 1904,
1911, 1918, 1925, 1932, 1939, 1946, 1953, 1960, 1967, 1974, 1981, 1988, 1995]
```

Add the n number of names in a list and print them alphabetically and reverse alphabetically.

In [5]:
```
my_str = input("Enter a string: ")

# breakdown the string into a list of words
```

- A tuple is a collection of objects which ordered and immutable. Tuples are sequences, just like lists. The differences between tuples and lists are, the tuples cannot be changed unlike lists and tuples use parentheses, whereas lists use square brackets.
- Tuples are immutable which means you cannot update or change the values of tuple elements. You are able to take portions of existing tuples to create new tuples
- The most basic data structure in Python is the sequence. Each element of a sequence is assigned a number - its position or index.
- The first index is zero, the second index is one, and so forth. Python hassix built-in types of sequences, but the most common ones are lists and tuples, which we would see in this tutorial.
- There are certain things you can do with all sequence types.
- These operations include indexing, slicing, adding, multiplying, and checking for membership. In addition, Python has built-in functions for finding the length of a sequence and for finding its largest and smallest elements.