

```
In [172]: #Q1 Create myTuple tuple with the follwoing values ("NPower", "JDA", "Tuesday", 30, 3
myTuple = ("NPower", "JDA", "Tuesday", 30, 3, 2021)
type(myTuple)
```

Out[172]: tuple

```
In [139]: #Q2 What is the type of myTuple
type(myTuple)
```

Out[139]: tuple

```
In [140]: #Q3 What is the Length of myTuple
len(myTuple)
```

Out[140]: 6

```
In [141]: #Q4 print the values in each index #Use regular indexing
print(myTuple[0])
print(myTuple[1])
print(myTuple[2])
print(myTuple[3])
print(myTuple[4])
print(myTuple[5])
```

NPower
JDA
Tuesday
30
3
2021

```
In [142]: #Q5 print the values in each index #Use negative indexing
print(myTuple[-1])
print(myTuple[-2])
print(myTuple[-3])
print(myTuple[-4])
print(myTuple[-5])
print(myTuple[-6])
```

2021
3
30
Tuesday
JDA
NPower

```
In [20]: #Q6 what is the type of each value
print('Type of 1st element: ',type(myTuple[0]))
print('Type of 2nd element: ',type(myTuple[1]))
print('Type of 3rd element: ',type(myTuple[2]))
print('Type of 4th element: ',type(myTuple[3]))
print('Type of 5th element: ',type(myTuple[4]))
print('Type of 6th element: ',type(myTuple[5]))
```

```
Type of 1st element: <class 'str'>
Type of 2nd element: <class 'str'>
Type of 3rd element: <class 'str'>
Type of 4th element: <class 'int'>
Type of 5th element: <class 'int'>
Type of 6th element: <class 'int'>
```

```
In [145]: #Q7 unpack myTuple in the following variables name,program,dayName,month,day,year
# print the variables
```

```
myTuple = (name, program, dayName, month, day, year)
print(myTuple)
```

NameError

Traceback (most recent call last)

Cell In[145], line 4

```
1 #Q7 unpack myTuple in the following variables name,program,dayName,month,day,year
2 # print the variables
----> 4 myTuple = (name, program, dayName, month, day, year)
      5 print(myTuple)
```

NameError: name 'name' is not defined

```
In [12]: #Q8 unpack myTuple2 in the following variables name,program,dayName.
# What will happen to variables (name,program,dayName) and (month,day,year)
```

```
In [146]: # Note the following
Tuple1=("Jerry",2,89) #This is a tuple with 3 elements
Tuple2=("Ulan",)#This is a tuple with 1 element
test="Leul" #This is a VARIABLE with string value
a,b,c=Tuple1
print("Type a",type(a))
print(a,b,c)
d=Tuple2
print(type(d))
print(d)
e=test
print(e)
```

```
Type a <class 'str'>
Jerry 2 89
<class 'tuple'>
('Ulan',)
Leul
```

```
In [147]: #Tuples are immutable
#we can always make the testTuple variable reference a new tuple in the memory
#and hold a different information
testTuple=(1,2,3)
print(testTuple)
testTuple=(4,5,6)
print(testTuple)
#But we can't change or edit a value for the existing tuple
testTuple[0]=6 #ERROR 'tuple' object does not support item assignment
```

```
(1, 2, 3)
(4, 5, 6)
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In[147], line 9
      7 print(testTuple)
      8 #But we can't change or edit a value for the existing tuple
----> 9 testTuple[0]=6
```

```
TypeError: 'tuple' object does not support item assignment
```

```
In [150]: #Q9 Reverse myTuple, output should look like (2021,3,30,"Tuesday","JDA","NPower")
rev_myTuple = reversed(myTuple)
rev_myTuple
```

```
Out[150]: <reversed at 0x21114ccb640>
```

```
In [151]: #Q10 Create nestedTuple=(("Coursera","course",6),("week",(2,"Lists","Tuple")))

nestedTuple = (('Coursera','Course',6),('Week',(2,'Lists','Tuple')))
nestedTuple
```

```
Out[151]: (('Coursera', 'Course', 6), ('Week', (2, 'Lists', 'Tuple')))
```

```
In [152]: #Q11 What is the output of nestedTuple[1:2]
nestedTuple[1:2]
```

```
Out[152]: (('Week', (2, 'Lists', 'Tuple')),)
```

```
In [153]: #Q12 print each element in the nestedTuple
print(nestedTuple[0][0])
print(nestedTuple[0][1])
print(nestedTuple[0][2])
print(nestedTuple[1][0])
print(nestedTuple[1][1][0])
print(nestedTuple[1][1][1])
print(nestedTuple[1][1][2])
```

```
Coursera
Course
6
Week
2
Lists
Tuple
```

```
In [154]: #Q13 Access (2,"Lists","Tuple") from nestedTuple

nestedTuple[1][1]
```

```
Out[154]: (2, 'Lists', 'Tuple')
```

```
In [155]: #Q14 Access the value "Lists" from nestedTuple
nestedTuple[1][1][1]
```

```
Out[155]: 'Lists'
```

```
In [156]: #Q15 Access the value "Tuple" from nestedTuple
nestedTuple[1][1][2]
```

```
Out[156]: 'Tuple'
```

```
In [157]: #Q16 Access the value "course" from nestedTuple
nestedTuple[0][1]
```

```
Out[157]: 'Course'
```

```
In [174]: #Q17 Concatenate myTuple with nestedTuple
```

```
concat_Tuple = myTuple + nestedTuple
concat_Tuple
```

```
Out[174]: ('NPower',
           'JDA',
           'Tuesday',
           30,
           3,
           2021,
           ('Coursera', 'Course', 6),
           ('Week', (2, 'Lists', 'Tuple')))
```

```
In [177]: #Q18 add your name to the tuple
```

```
name_Tuple = ("Haider", "Rizvi")
name_Tuple = name_Tuple + concat_Tuple
name_Tuple
```

```
Out[177]: ('Haider',
           'Rizvi',
           'NPower',
           'JDA',
           'Tuesday',
           30,
           3,
           2021,
           ('Coursera', 'Course', 6),
           ('Week', (2, 'Lists', 'Tuple')))
```

```
In [178]: #Q19 check whether Coursera exists within myTuple
# NOTE "in" doesn't work properly with nested tuples
```

```
"Coursera" in nestedTuple[0]
```

Out[178]: True

```
In [179]: #Q20 check whether 55 exists within testTuple
testTuple=(1,2,33,55,6,55)
55 in testTuple
```

Out[179]: True

```
In [184]: #Q21 Find the index of JDA in myTuple

# Find the index of 'Coursera' in myTuple
# NOTE index doesn't work properly with

myTuple.index('JDA')
```

Out[184]: 1

```
In [185]: #Q22 print index 8 from myTuple

myTuple[8]
```

```
-----
IndexError                                Traceback (most recent call last)
Cell In[185], line 3
      1 #Q22 print index 8 from myTuple
----> 3 myTuple[8]

IndexError: tuple index out of range
```

```
In [35]: #Q23 Get the 4th element from the begining of myTuple and 4th element from last o
```

```
In [187]: #Q24 Find how many times 55 appeared in testTuple [Hint: Use method count()]

55 in testTuple

testTuple.count(55)
```

Out[187]: 2

In []: *## List*

In [71]: *#Q1 Create a tuple with all participant names in the JDA program*

```
JDA = ('Bilkis Abolarinwa', 'Abdul Wasi Ahmadi', 'Uwaila Joy Aiwekhoe', 'Gift Ajayi', 'Azizah Alawusa', 'Derek Anderson',  
      'Hephzhi Andrews', 'Divya Annapureddy', 'Umair Anwar', 'Emmanuel Arogundade', 'Gohar Ayoub', 'Ihuaku Azuma-Awoyo',  
      'Jonathan Banks', 'Khaled Ben Ali', 'Luwam Beyin', 'Bansari Bhatt', 'Biruk Abdissa Biru', 'Omowumi Borokinni',  
      'Sophie Cervera', 'Annie Chaudhary', 'Martino DeLeon', 'Sandhya Deshahalli Ramalingaiah', 'Seethal Elias', 'Mahmoud El  
      'Farha Fatima', 'Alfred Frendo-Cumbo', 'Kamalini Gnanamoorthy', 'Xiangyun Gu', 'Jackson Hemingway', 'Yao Hong', 'Bouza  
      'Mahalakshmi Janarthanan', 'Prakash Kannel', 'Swati Karthik Varadarajan', 'Sandeep Kaur', 'Taranpreet Kaur',  
      'Harkirat Kaushal', 'Abeda Rezwana Khanam', 'Rohit Nitin Kher', 'Akriti Khullar', 'Poonam Kota', 'Snizhana Kshetska',  
      'Archana Kulkarni', 'Rahul Kumar', 'Shino Kuriakose', 'Lu Liu', 'Amber MacMullin', 'Mintu Mathew', 'David Mbiriri',  
      'John Carlo Moreno', 'Ogonna Nwankwo', 'Goodluck Obiakor', 'Chinenye Obiegbusi', 'Oluwaseun Odetola', 'Ayodele Odueke  
      'Joseph Ogor', 'Oluwatosin Ogundipe', 'Daniel Ehinome Okonoyi', 'Olufunmilayo Olaoti', 'Oluwatosin Olawoye', 'Ramon Ol  
      'Ramya Saroja Oruganti', 'Wan Pang', 'Anitha Panneer Selvam', 'Dhara Shaileshkumar Patel', 'Dharati Patel', 'Riyaben  
      'Elena Pellegrino', 'Maria Peskischeva', 'Archana Rajendrakumar', 'Pringa Ramavarman', 'Syed Muhammad Haider Raza Riz  
      'Kyle Schroeder-Bear', 'Salma Shahamiry', 'Bhumika Sharma', 'Sonal Sharma', 'Revathi Singaram', 'Carlos Siso',  
      'Vishwanatha Srinivasa', 'Meach Subaday', 'Gbolahan Sunmonu', 'Loren Taborda', 'VanTam (Tina) Tang', 'Teklit Tesfakin  
      'Jia Von Then', 'Rozina Aamir Vahora', 'Praveena Vijayanand', 'Qingyuan Xue', 'Halit Yildirim')
```

JDA


```
Out[71]: ('Bilkis Abolarinwa',  
         'Abdul Wasi Ahmadi',  
         'Uwaila Joy Aiwekhoe',  
         'Gift Ajayi',  
         'Azizah Alawusa',  
         'Derek Anderson',  
         'Hephzhi Andrews',  
         'Divya Annapureddy',  
         'Umair Anwar',  
         'Emmanuel Arogundade',  
         'Gohar Ayoub',  
         'Ihuaku Azuma-Awoyo',  
         'Jonathan Banks',  
         'Khaled Ben Ali',  
         'Luwam Beyin',  
         'Bansari Bhatt',  
         'Biruk Abdissa Biru',  
         'Omowumi Borokinni',  
         'Sophie Cervera',  
         'Annie Chaudhary',  
         'Martino DeLeon',  
         'Sandhya Deshahalli Ramalingaiah',  
         'Seethal Elias',  
         'Mahmoud Elshal',  
         'Farha Fatima',  
         'Alfred Frendo-Cumbo',  
         'Kamalini Gnanamoorthy',  
         'Xiangyun Gu',  
         'Jackson Hemingway',  
         'Yao Hong',  
         'Bouzaya Imen',  
         'Mahalakshmi Janarthanan',  
         'Prakash Kannel',  
         'Swati Karthik Varadarajan',  
         'Sandeep Kaur',  
         'Taranpreet Kaur',  
         'Harkirat Kaushal',  
         'Abeda Rezwana Khanam',  
         'Rohit Nitin Kher',  
         'Akriti Khullar',  
         'Poonam Kota',  
         'Snizhana Kshetska',  
         'Archana Kulkarni',  
         'Rahul Kumar',  
         'Shino Kuriakose',  
         'Lu Liu',  
         'Amber MacMullin',
```

'Mintu Mathew',
'David Mbiriri',
'John Carlo Moreno',
'Ogonna Nwankwo',
'Goodluck Obiakor',
'Chinenye Obiegbusi',
'Oluwaseun Odetola',
'Ayodele Odueke',
'Joseph Ogor',
'Oluwatosin Ogundipe',
'Daniel Ehinome Okonoyi',
'Olufunmilayo Olaoti',
'Oluwatosin Olawoye',
'Ramon Onel',
'Ramya Saroja Oruganti',
'Wan Pang',
'Anitha Panneer Selvam',
'Dhara Shaileshkumar Patel',
'Dharati Patel',
'Riyaben Patel',
'Elena Pellegrino',
'Maria Peskischeva',
'Archana Rajendrakumar',
'Pringa Ramavarman',
'Syed Muhammad Haider Raza Rizvi',
'Kyle Schroeder-Bear',
'Salma Shahamiry',
'Bhumika Sharma',
'Sonal Sharma',
'Revathi Singaram',
'Carlos Siso',
'Vishwanatha Srinivasa',
'Meach Subaday',
'Gbolahan Sunmonu',
'Loren Taborda',
'VanTam (Tina) Tang',
'Teklit Tesfakiros',
'Jia Von Then',
'Rozina Aamir Vahora',
'Praveena Vijayanand',
'Qingyuan Xue',
'Halit Yildirim')

```
In [72]: #Q2 Convert the tuple into a list called pps_L
pps_L = list(JDA)
pps_L
```

```
Out[72]: ['Bilkis Abolarinwa',  
          'Abdul Wasi Ahmadi',  
          'Uwaila Joy Aiwekhoe',  
          'Gift Ajayi',  
          'Azizah Alawusa',  
          'Derek Anderson',  
          'Hephzhi Andrews',  
          'Divya Annapureddy',  
          'Umair Anwar',  
          'Emmanuel Arogundade',  
          'Gohar Ayoub',  
          'Ihuaku Azuma-Awoyo',  
          'Jonathan Banks',  
          'Khaled Ben Ali',  
          'Luwam Beyin',  
          'Bansari Bhatt',  
          'Biruk Abdissa Biru',  
          'Omowumi Borokinni',  
          'Sophie Cervera',  
          'Annie Chaudhary',  
          'Martino DeLeon',  
          'Sandhya Deshahalli Ramalingaiah',  
          'Seethal Elias',  
          'Mahmoud Elshal',  
          'Farha Fatima',  
          'Alfred Frendo-Cumbo',  
          'Kamalini Gnanamoorthy',  
          'Xiangyun Gu',  
          'Jackson Hemingway',  
          'Yao Hong',  
          'Bouzaya Imen',  
          'Mahalakshmi Janarthanan',  
          'Prakash Kannel',  
          'Swati Karthik Varadarajan',  
          'Sandeep Kaur',  
          'Taranpreet Kaur',  
          'Harkirat Kaushal',  
          'Abeda Rezwana Khanam',  
          'Rohit Nitin Kher',  
          'Akriti Khullar',  
          'Poonam Kota',  
          'Snizhana Kshetska',  
          'Archana Kulkarni',  
          'Rahul Kumar',  
          'Shino Kuriakose',  
          'Lu Liu',  
          'Amber MacMullin',
```

'Mintu Mathew',
'David Mbiriri',
'John Carlo Moreno',
'Ogonna Nwankwo',
'Goodluck Obiakor',
'Chinenye Obiegbusi',
'Oluwaseun Odetola',
'Ayodele Odueke',
'Joseph Ogor',
'Oluwatosin Ogundipe',
'Daniel Ehinome Okonoyi',
'Olufunmilayo Olaoti',
'Oluwatosin Olawoye',
'Ramon Onel',
'Ramya Saroja Oruganti',
'Wan Pang',
'Anitha Panneer Selvam',
'Dhara Shaileshkumar Patel',
'Dharati Patel',
'Riyaben Patel',
'Elena Pellegrino',
'Maria Peskischeva',
'Archana Rajendrakumar',
'Pringa Ramavarman',
'Syed Muhammad Haider Raza Rizvi',
'Kyle Schroeder-Bear',
'Salma Shahamiry',
'Bhumika Sharma',
'Sonal Sharma',
'Revathi Singaram',
'Carlos Siso',
'Vishwanatha Srinivasa',
'Meach Subaday',
'Gbolahan Sunmonu',
'Loren Taborda',
'VanTam (Tina) Tang',
'Teklit Tesfakiros',
'Jia Von Then',
'Rozina Aamir Vahora',
'Praveena Vijayanand',
'Qingyuan Xue',
'Halit Yildirim']

```
In [73]: #Q3 What is the Length of your List  
len(pps_L)
```

Out[73]: 89

```
In [74]: #Q4 Find the values stored in indexes [3],[5],[14],[30],[38]  
print(pps_L[3])  
print(pps_L[5])  
print(pps_L[14])  
print(pps_L[30])  
print(pps_L[38])
```

Gift Ajayi
Derek Anderson
Luwam Beyin
Bouzaya Imen
Rohit Nitin Kher

```
In [75]: #Q5 Check if "Sarah" is in the List  
'Sarah' in pps_L
```

Out[75]: False

```
In [77]: #Q6 Add "Sarah" to your List  
pps_L.append('Sarah')  
pps_L[-1]
```

Out[77]: 'Sarah'

```
In [78]: #Q7 Check if "Bashir" is in the List  
'Bashir' in pps_L
```

Out[78]: False

```
In [79]: #Q9 Add "Bashir" to your List  
pps_L.append('Bashir')  
pps_L[-1]
```

Out[79]: 'Bashir'

```
In [80]: #Q10 Find the index of the element "Sarah" in the List  
pps_L.index('Sarah')
```

Out[80]: 89

```
In [86]: #Q11 Change the value of the index in Q10 from "Sarah" to "Instructor"
pps_L[pps_L.index('Sarah')] = 'Instructor'
pps_L[89]
```

```
Out[86]: ['Bilkis Abolarinwa',
          'Abdul Wasi Ahmadi',
          'Uwaila Joy Aiwekhoe',
          'Gift Ajayi',
          'Azizah Alawusa',
          'Derek Anderson',
          'Hephzhi Andrews',
          'Divya Annapureddy',
          'Umair Anwar',
          'Emmanuel Arogundade',
          'Gohar Ayoub',
          'Ihuaku Azuma-Awoyo',
          'Jonathan Banks',
          'Khaled Ben Ali',
          'Luwam Beyin',
          'Bansari Bhatt',
          'Biruk Abdissa Biru',
          'Omowumi Borokinni',
          'Sophie Cervera',
          'Annie Chaudhary',
          'Martino DeLeon',
          'Sandhya Deshahalli Ramalingaiah',
          'Seethal Elias',
          'Mahmoud Elshal',
          'Farha Fatima',
          'Alfred Frendo-Cumbo',
          'Kamalini Gnanamoorthy',
          'Xiangyun Gu',
          'Jackson Hemingway',
          'Yao Hong',
          'Bouzaya Imen',
          'Mahalakshmi Janarthanan',
          'Prakash Kannel',
          'Swati Karthik Varadarajan',
          'Sandeep Kaur',
          'Taranpreet Kaur',
          'Harkirat Kaushal',
          'Abeda Rezwana Khanam',
          'Rohit Nitin Kher',
          'Akriti Khullar',
          'Poonam Kota',
          'Snizhana Kshetska',
          'Archana Kulkarni',
          'Rahul Kumar',
          'Shino Kuriakose',
          'Lu Liu',
          'Amber MacMullin',
```


'Mintu Mathew',
'David Mbiriri',
'John Carlo Moreno',
'Ogonna Nwankwo',
'Goodluck Obiakor',
'Chinenye Obiegbusi',
'Oluwaseun Odetola',
'Ayodele Odueke',
'Joseph Ogor',
'Oluwatosin Ogundipe',
'Daniel Ehinome Okonoyi',
'Olufunmilayo Olaoti',
'Oluwatosin Olawoye',
'Ramon Onel',
'Ramya Saroja Oruganti',
'Wan Pang',
'Anitha Panneer Selvam',
'Dhara Shaileshkumar Patel',
'Dharati Patel',
'Riyaben Patel',
'Elena Pellegrino',
'Maria Peskischeva',
'Archana Rajendrakumar',
'Pringa Ramavarman',
'Syed Muhammad Haider Raza Rizvi',
'Kyle Schroeder-Bear',
'Salma Shahamiry',
'Bhumika Sharma',
'Sonal Sharma',
'Revathi Singaram',
'Carlos Siso',
'Vishwanatha Srinivasa',
'Meach Subaday',
'Gbolahan Sunmonu',
'Loren Taborda',
'VanTam (Tina) Tang',
'Teklit Tesfakiros',
'Jia Von Then',
'Rozina Aamir Vahora',
'Praveena Vijayanand',
'Qingyuan Xue',
'Halit Yildirim',
'Instructor',
'Bashir']

```
In [88]: #Q12 Find the index of the element "Bashir" in the list
pps_L.index('Bashir')
#pps_L
```

```
Out[88]: 90
```

```
In [89]: #Q13 Change the value of the index in Q12 from "Bashir" to "Peer-Mentor"
pps_L[pps_L.index('Bashir')] = 'Peer-mentor'
pps_L
```

```
Out[89]: ['Bilkis Abolarinwa',
          'Abdul Wasi Ahmadi',
          'Uwaila Joy Aiwekhoe',
          'Gift Ajayi',
          'Azizah Alawusa',
          'Derek Anderson',
          'Hephzhi Andrews',
          'Divya Annapureddy',
          'Umair Anwar',
          'Emmanuel Arogundade',
          'Gohar Ayoub',
          'Ihuaku Azuma-Awoyo',
          'Jonathan Banks',
          'Khaled Ben Ali',
          'Luwam Beyin',
          'Bansari Bhatt',
          'Biruk Abdissa Biru',
          'Omowumi Borokinni',
          'Sophie Cervera',
          'Annie Chaudhary',
          'Martino DeLeon',
          'Sandhya Deshahalli Ramalingaiah',
          'Seethal Elias',
          'Mahmoud Elshal',
          'Farha Fatima',
          'Alfred Frendo-Cumbo',
          'Kamalini Gnanamoorthy',
          'Xiangyun Gu',
          'Jackson Hemingway',
          'Yao Hong',
          'Bouzaya Imen',
          'Mahalakshmi Janarthanan',
          'Prakash Kannel',
          'Swati Karthik Varadarajan',
          'Sandeep Kaur',
          'Taranpreet Kaur',
          'Harkirat Kaushal',
          'Abeda Rezwana Khanam',
          'Rohit Nitin Kher',
          'Akriti Khullar',
          'Poonam Kota',
          'Snizhana Kshetska',
          'Archana Kulkarni',
          'Rahul Kumar',
          'Shino Kuriakose',
          'Lu Liu',
          'Amber MacMullin',
```

'Mintu Mathew',
'David Mbiriri',
'John Carlo Moreno',
'Ogonna Nwankwo',
'Goodluck Obiakor',
'Chinenye Obiegbusi',
'Oluwaseun Odetola',
'Ayodele Odueke',
'Joseph Ogor',
'Oluwatosin Ogundipe',
'Daniel Ehinome Okonoyi',
'Olufunmilayo Olaoti',
'Oluwatosin Olawoye',
'Ramon Onel',
'Ramya Saroja Oruganti',
'Wan Pang',
'Anitha Panneer Selvam',
'Dhara Shaileshkumar Patel',
'Dharati Patel',
'Riyaben Patel',
'Elena Pellegrino',
'Maria Peskischeva',
'Archana Rajendrakumar',
'Pringa Ramavarman',
'Syed Muhammad Haider Raza Rizvi',
'Kyle Schroeder-Bear',
'Salma Shahamiry',
'Bhumika Sharma',
'Sonal Sharma',
'Revathi Singaram',
'Carlos Siso',
'Vishwanatha Srinivasa',
'Meach Subaday',
'Gbolahan Sunmonu',
'Loren Taborda',
'VanTam (Tina) Tang',
'Teklit Tesfakiros',
'Jia Von Then',
'Rozina Aamir Vahora',
'Praveena Vijayanand',
'Qingyuan Xue',
'Halit Yildirim',
'Instructor',
'Peer-mentor']

```
In [50]: ## Dictionary
```

```
In [102]: #Q1 Create a dictionary for all participants in the JDA program.
#Keys should be the first letter in their names, Values should be their names.
#[Hint] dic={"L":(Leul),"J":(Jerry,Jainam).....}

dic = {'A': 'Alfred', 'D': 'David', 'H': 'Haider', 'I': ('Ihuaku', 'Ifenna'), 'S': 'Salma', 'V': 'Vishwanatha'}
dic
```

```
Out[102]: {'A': 'Alfred',
'D': 'David',
'H': 'Haider',
'I': ('Ihuaku', 'Ifenna'),
'S': 'Salma',
'V': 'Vishwanatha'}
```

```
In [92]: #Q2 What are the keys of your dictionary
dic.keys()
```

```
Out[92]: dict_keys(['A', 'D', 'H', 'I', 'S', 'V'])
```

```
In [93]: #Q3 What are the values of your dictionary
dic.values()
```

```
Out[93]: dict_values(['Alfred', 'David', 'Haider', ('Ihuaku', 'Ifenna'), 'Salma', 'Vishwanatha'])
```

```
In [99]: #Q4 Find participant's names that start with J,I,M,C,A
```

```
print(dic['I'])
print(dic['A'])

('Ihuaku', 'Ifenna')
Alfred
```

```
In [110]: #Q5 Add the following to your dictionary
# Key<(K,PD)>, Value<(Kristina)>
```

```
dic[("K","PD")] = "Kristina"
dic
```

```
Out[110]: {'A': 'Alfred',
'D': 'David',
'H': 'Haider',
'I': ('Ihuaku', 'Ifenna'),
'S': 'Salma',
'V': 'Vishwanatha',
('K', 'PD'): 'Kristina'}
```

```
In [111]: #Q6 Find the Length of your dictionary
len(dic)
dic
```

```
Out[111]: {'A': 'Alfred',
'D': 'David',
'H': 'Haider',
'I': ('Ihuaku', 'Ifenna'),
'S': 'Salma',
'V': 'Vishwanatha',
('K', 'PD'): 'Kristina'}
```

```
In [112]: #Q7 Add the following to your dictionary
# Key<(S,Instructor)>, Value<(Sarah)>
```

```
dic[("S","Instructor")] = "Sarah"
dic
```

```
Out[112]: {'A': 'Alfred',
'D': 'David',
'H': 'Haider',
'I': ('Ihuaku', 'Ifenna'),
'S': 'Salma',
'V': 'Vishwanatha',
('K', 'PD'): 'Kristina',
('S', 'Instructor'): 'Sarah'}
```

```
In [113]: #Q8 Find the Length of your dictionary
len(dic)
```

```
Out[113]: 8
```

In [114]: *#Q7 Delete the following keys from your dictionary*

```
#Key<(S,Instructor)>  
#Key<J>
```

```
del(dic[('S', 'Instructor')])  
dic
```

Out[114]: {'A': 'Alfred',
 'D': 'David',
 'H': 'Haider',
 'I': ('Ihuaku', 'Ifenna'),
 'S': 'Salma',
 'V': 'Vishwanatha',
 ('K', 'PD'): 'Kristina'}

In [115]: *#Q9 Find the Length of your dictionary*

```
len(dic)
```

Out[115]: 7

In [116]: *#Q10 Create a new dictionary called synonyms_dic*

```
synonyms_dic = {}  
synonyms_dic
```

Out[116]: {}

In [118]: *#Q11 Add the synonyms of the following words to your dictionary*

```
#Bad, Good, Awesome, Cold, Easy, Hard, Big, Small
```

```
synonyms_dic = {'Bad': 'Wrong', 'Good': 'Acceptable', 'Awesome': 'Amazing', 'Cold': 'Chilly', 'Easy': 'Straighfoward',  
                  'Hard': 'Difficult', 'Big': 'Huge', 'Small': 'Little'}  
synonyms_dic
```

Out[118]: {'Bad': 'Wrong',
 'Good': 'Acceptable',
 'Awesome': 'Amazing',
 'Cold': 'Chilly',
 'Easy': 'Straighfoward',
 'Hard': 'Difficult',
 'Big': 'Huge',
 'Small': 'Little'}


```
In [121]: #Q12 Use your dictionary to find the synonym of the following words
# Awesome, Easy, Small, Cold
print(synonyms_dic['Awesome'])
print(synonyms_dic['Easy'])
print(synonyms_dic['Small'])
print(synonyms_dic['Cold'])
```

Amazing
Straighfoward
Little
Chilly

```
In [189]: #Q13 Delete everything from synonyms_dic

synonyms_dic.clear()
```

```
In [190]: #Q14 Print all values in synonyms_dic

print(synonyms_dic.values())

dict_values([])
```

```
In [122]: #Q15 Convert the below two lists into dictionary
Keys=["True", "False"]
Values=[1,0]
dict = {Keys[0]:Values[0], Keys[1]:Values[1]}
dict
```

```
Out[122]: {'True': 1, 'False': 0}
```

```
In [123]: #Q16 Sum all the values in the following dictionary
income={'Jan':1000, 'Feb':1500, 'Mar':980, 'April':1000}
sum(income.values())
```

```
Out[123]: 4480
```

```
In [69]: #Q17 Create the following dictionary encryption={10:"hungry", 101:"am", 110:"I",
# What should be the decryption of the following code "110 101 10 1011 110 1001 1
```

```
In [ ]:
```

