In [144]:

```
#Q1 Create myTuple tuple with the follwoing values ("NPower","JDA","Tuesday",30,3,2021)
myTuple =("NPower","JDA","Tuesday",30,3,2021)
myTuple
```

Out[144]:

```
('NPower', 'JDA', 'Tuesday', 30, 3, 2021)
```

In [2]:

```
#Q2 What is the type of myTuple
type(myTuple)
```

Out[2]:

```
tuple
```

In [3]:

```
#Q3 What is the length of myTuple
len(myTuple)
```

Out[3]:

```
6
```

In [5]:

```
#Q4 print the values in each index #Use regular indexing
print(myTuple[0:6])
```

```
('NPower', 'JDA', 'Tuesday', 30, 3, 2021)
```

In [25]:

```
#Q5 print the values in each index #Use negative indexing
print(myTuple[-6:-1])
```

```
('NPower', 'JDA', 'Tuesday', 30, 3)
```

In [29]:

```
#Q6 what is the type of each value
print(type(myTuple[0]))
print(type(myTuple[1]))
print(type(myTuple[2]))
print(type(myTuple[3]))
print(type(myTuple[4]))
```

```
<class 'str'>
<class 'str'>
<class 'str'>
<class 'int'>
<class 'int'>
```

In [36]:

```
#Q7 unpack myTuple in the follwoeing variables name,program,dayName,month,day,year accord
ingly
# print the variables
(name,program,dayName,month,day,year)=myTuple
print(name)
print(program)
print(dayName)
print(month)
print(day)
print(year)
```

```
NPower
JDA
Tuesday
30
3
2021
```

In [ ]:

```
#Q8 unpack myTuple2 in the follwoeing variablesname,program,dayName.
# What will happen to variables (name,program,dayName) and (month,day,year)
```

In [37]:

```
# 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 'str'>
Ulan
Leul
```

In [ ]:

```
#Tuples are immutable
#we can always make the testTuple variable reference a new tuple in the memory
#and holding 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 assignmentmy
```

In [145]:

```
#Q9 Reverse myTuple, output should looks like ("NPower","JDA","Tuesday",30,3,2021)


myTuple = tuple(reversed(myTuple))
myTuple
```

Out[145]:

```
(2021, 3, 30, 'Tuesday', 'JDA', 'NPower')
```

In [122]:

```
#Q10 Create nestedTuple=(("Coursera","course",6),("week",(2,"Lists","Tuple")))
nestedTuple=(("Coursera","course",6),("week",(2,"Lists","Tuple")))
nestedTuple
```

Out[122]:

```
(('Coursera', 'course', 6), ('week', (2, 'Lists', 'Tuple')))
```

```
(('Coursera', 'course', 6), ('week', (2, 'Lists', 'Tuple')))
```

In [101]:

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

Out[101]:

```
(('week', (2, 'Lists', 'Tuple')),)
```

In [111]:

```
#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 [113]:

```
#Q13 Access (2,"Lists","Tuple") from nestedTuple
nestedTuple[1][1]
```

Out[113]:

```
(2, 'Lists', 'Tuple')
```

In [115]:

```
#Q14 Access "Lists" from nestedTuple
nestedTuple[1][1][1]
```

Out[115]:

```
'Lists'
```

In [116]:

```
#Q15 Access "Tuple" from nestedTuple
nestedTuple[1][1][2]
```

Out[116]:

```
'Tuple'
```

In [117]:

```
#Q16 Access "course" from nestedTuple
nestedTuple[0][1]
```

Out[117]:

```
'course'
```

In [132]:

```
#Q17 Concatenate myTuple with nestedTuple
myTuple=myTuple+nestedTuple
myTuple
```

In [140]:

```python
#Q18 add your name to the tuple
myTuple=myTuple+("santhiya",)
myTuple
```

Out[140]:

```
('NPower',
 'JDA',
 'Tuesday',
 30,
 3,
 2021,
 ('Coursera', 'course', 6),
 ('week', (2, 'Lists', 'Tuple')),
 'santhiya')
```

In [137]:

```python
#Q19 check whether Coursera exists within a myTuple

# NOTE in doesn't work properly with nested tuples # Wrong output
"Coursera" in myTuple
```

Out[137]:

```
False
```

In [139]:

```python
#Q20 check whether an element exists within a testTuple
testTuple=("san", 1, "adhith")
print("adhith" in testTuple)
print(2 in testTuple)
print("san" in testTuple)
```

```
True
False
True
```

In [130]:

```python
#Q21 Find the index of JDA in myTuple


# Find the index of 'Coursera' in myTuple
# NOTE index doesn't work properly with nested tuples # Wrong output

myTuple.index("JDA")
```

Out[130]:

```
1
```

In [141]:

```python
#Q22 print index 8 from myTuple
myTuple[8]
```

Out[141]:

```
'santhiya'
```

In [125]:

```python
#Q23 Get the 4th element and 4th element from last of a myTuple
print(myTuple[3])
print(myTuple[-4])
```

```
30
('Coursera', 'course', 6)
```

In [124]:

```python
[illegible]
```

```
#Q24 Find how many times 27 appeared in the tuple [Hint: Use method cou
print(myTuple.count(27))
```

```
0
```

In [ ]:

```

```