Tuple and Sets

January 7, 2021

```
In [1]: # Tuple
        # collection of elements similar to list, uses (), elements can not be changed once it i
        # In tuple, (1,2.5) is same as 1,2.5
        # In tuple, (1,) is same as 1
        # In tuple, ((1,),) is same as (1,),
        # elements can be accessed with index values
        # immutability- elements can not be changed in tuple

In [2]: t=(1,2.5,'student',"college")   #Define tuple

In [3]: t

Out[3]: (1, 2.5, 'student', 'college')

In [4]: t[0]         # first element of tuple (using index)

Out[4]: 1

In [5]: t[3]         # fourth element of tuple as indexing starts from 0,1 and so on from left t

Out[5]: 'college'

In [6]: t[1:5:2]          # prints corresponding slice

Out[6]: (2.5, 'college')

In [7]: t[1:3:2]

Out[7]: (2.5,)

In [8]: t[2]='Hello'      #shows error means elements of tuple can not be changed once assigned

        ---------------------------------------------------------------------------

        TypeError                                 Traceback (most recent call last)

        <ipython-input-8-df773c3d2650> in <module>
    ----> 1 t[2]='Hello'


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

```
In [9]: # swapping values
        # Example: Given a= 5 and b= 7, swap the values of a and b (traditional approach)
        a=5
        b=7

In [10]: a

Out[10]: 5

In [11]: b

Out[11]: 7

In [12]: temp=a
         a=b
         b=temp

In [13]: a

Out[13]: 7

In [14]: b

Out[14]: 5

In [15]: # we swap values of a and b using python
         a=5
         b=7
         a,b=b,a
         a

Out[15]: 7

In [16]: b

Out[16]: 5

In [17]: # we can do swapping for different data type also
         a= 2.5
         b="Hello"
         a,b=b,a

In [18]: a

Out[18]: 'Hello'

In [19]: b

Out[19]: 2.5
```

```
In [20]: # Set
         # unordered collection of unique elements
         # mutable
         # we can add or remove elements from it

In [21]: a_list=[1,2,3,4,5,2]
         a=set(a_list)

In [22]: a                          # duplicates are removed in output and there is a collectio

Out[22]: {1, 2, 3, 4, 5}

In [23]: b={1,2,1,3,2,4,5,2} # direct command

In [24]: b

Out[24]: {1, 2, 3, 4, 5}

In [25]: c=set()    #empty set

In [26]:

Out[26]: set()

In [29]: b_list=[[1],2,1,,6,2]
         b=set(b_list)
         # gives error because sets can contain numbers, tuples and strings but not mutable elem


          File "<ipython-input-29-0a770385b72d>", line 1
        b_list=[[1],2,1,,6,2]
                        ^
    SyntaxError: invalid syntax



In [30]: # Operations on set

In [31]: p=set([1,2,3,5,8])
         q=set([2,3,5,7,11,13])

In [32]: p.add(13)              # add 13 to given set p
         p

Out[32]: {1, 2, 3, 5, 8, 13}

In [34]: p.add(2)       # no effect as 2 is already there in the set
         p

Out[34]: {1, 2, 3, 5, 8, 13}
```

```
In [35]: p.remove(13)    # remove 13 from set p
         p

Out[35]: {1, 2, 3, 5, 8}

In [36]: p.remove(20)     # gives error as 20 is not member of p
         p


        ---------------------------------------------------------------------------

        KeyError                                  Traceback (most recent call last)

        <ipython-input-36-038e610b9a43> in <module>
    ----> 1 p.remove(20)
          2 p


        KeyError: 20


In [37]: # Union of two sets p and q
         # syntax: p|q or p.union(q)

In [38]: p|q

Out[38]: {1, 2, 3, 5, 7, 8, 11, 13}

In [39]: p.union(q)

Out[39]: {1, 2, 3, 5, 7, 8, 11, 13}

In [40]: # Intersection of two sets
         # syntax: p&q or p.intersection(q)

In [41]: p&q

Out[41]: {2, 3, 5}

In [42]: p.intersection(q)

Out[42]: {2, 3, 5}

In [43]: p-q   # difference of two sets

Out[43]: {1, 8}

In [44]: p.difference(q)   # difference of two sets

Out[44]: {1, 8}
```

```
In [45]: # symmetric difference
         # syntax: p^q or p.symmetric_difference(q)

In [46]: p^q

Out[46]: {1, 7, 8, 11, 13}

In [47]: p.symmetric_difference(q)

Out[47]: {1, 7, 8, 11, 13}

In [49]: #checking for subsets
         a={1,2,3,4}
         b={1,2}
         b<=a

Out[49]: True

In [50]: b.issubset(a)

Out[50]: True

In [51]: a<=b

Out[51]: False

In [52]: a.issubset(b)

Out[52]: False

In [53]: # checking for supersets
         a>=a

Out[53]: True

In [54]: a>=b

Out[54]: True

In [55]: b>=a

Out[55]: False

In [56]: # length or cardinality of a set
         len(a)

Out[56]: 4

In [57]: len(b)

Out[57]: 2
```

```
In [58]: 1 in a    # shows true means 1 is member of set a

Out[58]: True

In [59]: 7 in a     # shows false as 7 is not member of a

Out[59]: False

In [60]: a(2)      # sets do not support indexing hence slicing and striding is not possible


        ---------------------------------------------------------------------------

        TypeError                                 Traceback (most recent call last)

        <ipython-input-60-36b638668758> in <module>
    ----> 1 a(2)


        TypeError: 'set' object is not callable


In [ ]:
```