DataStructures

January 3, 2017

1 Python main data structures

1.1 Ordered data structures

Lists and tuples are ordered sequences of objects. List and tuples can contain any type of objects.

1.1.1 Lists

Lists are mutable so they can be extended or reduced at will. A list offers a large number of methods which is really handy. A list can be used to create a stack or a queue. The list is the most used data structure, but is not optimized for all usage.

e.g : A stack with a list

```
In [10]: #We put some objects in a list
         stack = ["geography","statistics","biology","linear algebra"]
         print(stack)

         #We append some more objects
         stack.append("physics")
         stack.append("history")
         stack.append("economics")
         print(stack)

         #And we get the lasts which arrived one by one, on the top of the stack

         print(stack.pop())
         print(stack)
         print(stack.pop())
         print(stack)
         print(stack.pop())
         print(stack)
```

```
['geography', 'statistics', 'biology', 'linear algebra']
['geography', 'statistics', 'biology', 'linear algebra', 'physics', 'history', 'eco
economics
['geography', 'statistics', 'biology', 'linear algebra', 'physics', 'history']
history
['geography', 'statistics', 'biology', 'linear algebra', 'physics']
```

```
physics
['geography', 'statistics', 'biology', 'linear algebra']
```

1.1.2 Tuples

Tuples are useful because they :

- are faster than lists (optimzed)
- protect the data, because they are immutable

The counterpart of this optimized structure of data is the small number of methods it offers : index & count The immutable part of the object can be a bad side as well in some cases. Tuples can be used to return multiple variables from a function with unpacking.

```
In [11]: #A sample of unpacking, let's imagine b is the return of a function
         b = ("Bob", 19, "CS")
         (name, age, studies) = b     # tuple unpacking
         print(name)
         print(age)
         print(studies)

Bob
19
CS
```

1.2 Unordered data structures

1.2.1 Dict

A dictionary is a sequence of items. Each item is a pair made of a key and a value. Keys can't be duplicated in a dictionary. Dictionaries are not sorted and there is no concept of order among elements ! The main operations on a dictionary are storing a value with some key and extracting the value given the key.

```
In [12]: dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}

         print ("dict['Name']: ", dict['Name'])
         print ("dict['Age']: ", dict['Age'])

         dict['Age'] = 8; # update existing entry
         dict['School'] = "DPS School"; # Add new entry

dict['Name']:  Zara
dict['Age']:  7
```

1.2.2 Sets

Sets are mutable unordered sequence of unique elements. The unique side brings this structure to be used for set of identifiers and other comparable uses. Set objects also support mathematical operations like union, intersection, difference, and symmetric difference.

```
In [13]: a = set([1, 2, 3, 4])
         b = set([3, 4, 5, 6])
         print(a | b) # Union
         print(a & b) # Intersection
         print(a < b) # Subset
         print(a - b) # Difference
         print(a ^ b) # Symmetric Difference
```

```
{1, 2, 3, 4, 5, 6}
{3, 4}
False
{1, 2}
{1, 2, 5, 6}
```

1.3 Links

- http://www.thomas-cokelaer.info/tutorials/python/
- https://docs.python.org/3.1
- https://welcomedata.wordpress.com/2015/07/03/using-lists-as-stacks-and-queues-in-python/
- http://openbookproject.net/thinkcs/python/english3e/tuples.html
- https://www.tutorialspoint.com/python/python_dictionary.htm

```
In [ ]:
```