Python methods

- **Python built-in data types (arrays)**
  - **List**
    - **Characteristic - Ordered, Changeable, Allow duplicate values**
    - **Functions**
      - len(list) - return the length of the list
      - type(list) - <class 'list'>
      - list() constructor - myList = list(("a", "b", "c"))
      - thislist[2:5] - access list from index 2, 3, 4 (starting from 0) 不包含range末尾的position
      - if "sth" in thislist: - check if item exists
      - thislist.insert(index, sth) - insert items at the specified index
      - thislist.append(sth) - add item to the end of the list
      - thislist.extend(anotherlist/anothertuple) - thislist + anotherlist/anothertuple
      - thislist.remove(item) - remove the specified item value
      - thislist.pop(index) - remove the specified index 没有index就pop掉末尾
      - del thislist[idx] - remove specified index
      - del thislist - delete the whole list
      - thislist.clear() - delete the whole list, but the list still remains
    - **Loops functions**
      - for loops
        - for x in thislist: print(x)
        - for i in range(len(thislist)):
      - while loops
        - i = 0; while i < len(thislist): ... i+= 1
      - print(x) for x in thislist
      - for x in thislist: if "a" in x: print True
    - **List Comprehensions**
      - **filter**
        - newlist = [x for x in fruits if "a" in x]
        - **newlist = [*expr* for *item* in *iterable* if *condition == True*] - if和for的位置可以调换**
        - expr可以是很多种，x.upper(), x.lower(), etc.

          newlist = ['hello' for x in fruits]

        - newlist = [x if x != 'banana' else 'orange' for x in fruits] - 用orange替换banana，其他照旧 - 可以用来写替换replace method
    - **Sort Lists**
        - thislist.sort() - sort alphabetically/numerically
        - thislist.sort(reverse = True) - sort Descending alphabet/numeric order
        - **Customize sort**
            - **thislist.sort(key = myfunc)** - 类似于sorting constructor，例子里是到50的距离，从小到大排序

              ```
              def myfunc(n):
                return abs(n-50)
              ```
            - thislist.sort(key = str.lower)要求case-insensitive sort
        - thislist.reverse() - 直接reverse current sorting, 和alphabet无关
    - **Copy Lists**
        - mylist = thislist.copy() - copy value而不是reference，直接等于的话会copy reference，value同时修改
        - mylist = list(thislist) - copy value
    - **Join Lists**
        - list3 = list1 + list2
        - for x in list2: list1.append(x)
        - list1.extend(list2)
    - **List Built-in methods Summary**
        - append() - adds an element at the end of the list
        - clear() - removes all the elements from the list
        - copy() - returns a copy of the list
        - count() - returns the number of elements with the specified value
            - count = list.count("value") - 回这个list有多少个这个value
        - extend() - add the elements of a list (or any iterable) to the end of the current list
        - **index() - returns the index of the first element with the specified value**
            - 可以用来找value存在的第一个index，比如说CS50 python vanity plate.py中，list(str)后找第一个零的位置
        - insert() - adds an element at the specified position
        - pop() - remove the element at the specified position **idx**
        - remove() - remove the item with the specified value **value**
        - reverse() - reverse the order of the list
        - sort() - sort the list
- **Tuple**
    - **Characteristic - Ordered, Unchangeable, Allow duplicate**

- **Functions**
  - len(tuple) - return the number of items in the tuple
  - **thistuple = ("apple" ,) 必须加逗号，不加逗号type不是tuple是string**
  - thistuple = tuple(("apple", "banana")) - tuple constructor
  - thistuple[1/-1] - access tuple items by referring to the index
  - if "apple" in thistuple: return True - contains method
  - **Change Tuple Values - immutable as tuple, mutable as list**
    - mylist = list(mytuple)

      mylist[1] = "change"

      mytuple = tuple(mylist)
    - 用list 来append items
    - **可以add tuple to a tuple**

      thistuple += anothertuple
    - 用list来remove items
    - del thistuple - delete it completely
- **Unpacking tuples**
  - **(a,b,c) = tuple - 两者数量要一样，不然就是(a, *b) - *代表剩下的value是list**
- **Loop Tuple**
  - for loop
    - for x in thistuple:
    - for i in range(len(thistuple)):
  - while loop
    - i = 0; while i < len(thistuple): ... i += 1
- **Join Tuples**
  - tuple3 = tuple1 + tuple2
  - mytuple = tuple1 * 2 - 单纯后面复制粘贴
- **Tuple Built-in Methods**
  - count() - return the number of times a specified value occurs in a tuple

    tuple.count(value)
  - index() - search the tuple for a specified value and returns the position of where it was found

    tuple.index(value)
- **Set**
  - **Characteristic - Unordered, Unchangeable 但是可以add/remove items, Unindexed, NO duplicate**
  - **True = 1, False = 0; Set cannot have duplicate values**
  - **Functions**

- len(thisset) - number of items
- thisset = set((1, 2, 3)) - set constructor
- for x in thisset: - access set items
- thisset.add("value") - add new item to the set
- **thisset.update(anotherset/anotherlist...) - add elements from another set to the set**
- **thisset.remove("value") - remove value 如果value不存在会有error**
- **thisset.discard("value") - remove value 如果value不存在不会有error**
- x = thisset.pop() - **RANDOM** value in the set
- thisset.clear() - empty the set
- del thisset - delete the set completely

- **Loop Items**
  - for loop - for x in thisset:
- **！！！ Join Sets**
  - **set3 = set1.union(set2) - exclude duplicate items**
  - **set1.update(set2) - exclude duplicate items**
  - **set1.intersection_update(set2) - set 1 keep ONLY Duplicates**
  - **set3 = set1.intersection(set2) - 保存intersection到new set里**
  - **set1.symmetric_difference_update(set2) - Keep only elements that are NOT present in both set == OUTER JOIN - INNERJOIN == A并B - A交B**
  - **set3 = set1.symmetric_difference(set2) - 保存非intersection part到new set里**
- **Set Built-in Methods**
  - **和自身有关的都update，return new set都没有update**
  - add() - adds an element to the set
  - clear() - removes all the elements from the set
  - copy() - returns a copy of the set
  - **difference() - returns a set containing the difference between two or more sets**

    x.difference(y) = x-y
  - difference_update() - removes the items in this set that are also included in another, specified set
  - discard() - remove the specified item NO ERROR
  - intersection() - return a set that is the intersection of two other sets
  - intersection_update() - removes the items in this set that are not present in other, specified set
  - **isdisjoint()** - returns whether two sets have a intersection or not

- **issubset()** - returns whether another set contains this set or not
- **issuperset()** - returns whether this set contains another set or not
- pop() - removes a RAMDOM element from the set
- remove() - remove the specified element WILL HAVE ERROR IF NOT EXISTED
- symmetric_difference() - return a set with the symmetric difference of two sets
- symmetric_difference_update() - insert the symmetric difference from this set and another
- union() - return a set containing the union of sets
- update() - update the set with the union of this set and others

- **Dictionary**
  - **Characteristic - Ordered (NOT index but has a defined order), Changeable, NO duplicate**
  - **Referred to by using Key name**
  - **Functions**
    - len(thisdict) - number of items
    - **thisdict = dict(name='john', age=36, country='norway') - dict constructor**
    - **x = thisdict.get("key")** - access the item by using the key
    - **keyList = thisdict.keys()** - return a list of keys - **实时变化，如果本身dict key-value pair增减，keylist也会直接改变**
    - **valueList = thisdict.values()** - return a list of values - **实时变化，如果本身dict key-value pair增减，valuelist也会直接改变**
    - **itemList = thisdict.items() - return a list of key:value pairs - Tuples in a List 实时变化**

      [('1', 'one'), ('2', 'two)]
    - **thisdict["key"] = newValue - change value/add new key:value pair**
    - **thisdict.update({"key" : newValue})** - update the dictionary with the given items
    - thisdict.pop("key") - remove the item with the specified key name
    - thisitem.popitem() - remove the last inserted item
    - del thisdict["key"] - remove the item with the specified key name
    - del thisdict - delete the dictionary completely
    - thisdict.clear() - empty the dictionary
  - **Loop Dictionary**
    - for loop
      - for x in thisdict:
        - print(x) - keys
        - print(thisdict[x]) - values

        - for x in thisdict.values():
        - for x in thisdict.keys():
        - **for x, y in thisdict.items(): - both keys and values**
- **Copy Dictionary**
    - mydict = thisdict.copy()
    - mydict = dict(thisdict)
- **Nested Dictionary**
    - myLargeDict = {'child1' : dict1, 'child2' : dict2}
    - print(myLargeDict['child1'][nestedKeyName])
- **Dictionary Built-in Methods**
    - clear() - remove all the elements from the dictionary
    - copy() - returns a copy of the dictionary
    - **fromkeys() - return a dictionary with the specified keys and value**

        thisdict = dict.fromkeys(keys, value)
        keys values可以是任意值，keys一般是list/tuple，value是单个value，全部assign同一个value
        Default value = None
    - get() - return the value of the specified key
    - items() - return a list containing a tuple for each key value pair
    - keys() - return a list containing the dictionary's keys
    - pop() - remove the element with the specified key
    - popitem() - remove the last inserted key-value pair
    - **setdefault() - return the value of the specified key. If the key does not exist: insert the key, with the specified value**

        x = car.setdefault('key', 'value')- 如果key存在直接return value，key不存在就set key:value pair
    - update() - update the dictionary with the specified key-value pair
    - values() - return a list of all the values in the dictionary