**General:**

* **Compare to null:**

If variab == None:

* **Break:**while something:

if a == 10:

break

* **Return**

if a == b:

return

* H

**ToDo**

* Find the lambda functions and how they work.
* Sorts
* DataStructures like 2D array, map, tree, list, queue, heap, trie

**Dictionary:**

* **Init dict:**  
  dictnry = {}
* **Contains:**   
  message in self.dic
* **Size of dict:**  
  len(self.dic)
* **Add to dictionary:**self.dic[message] = timestamp

**Heap:**

* **Heapify min and max**  
  import heapq

listForTree = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]

heapq.heapify(listForTree) # for a min heap   
heapq.\_heapify\_max(listForTree) # for a maxheap!!

* **Pop Min vs Max:**heapq.heappop(minheap) # pop from minheap

heapq.\_heappop\_max(maxheap) # pop from maxheap

* **heapq.heappush(closest, (-dist, point[0], point[1]))**

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |