1. Fetching keystroke using pyhook and pythoncom
2. Passing each event to separate thread with lock
   1. Key up and key down event passes separately
   2. Key press and release time (epoch time in millisecond accuracy - 1455091739.100591), status, and key name are the arguments
3. Initial variable declaration
   1. Counter = 0
   2. Dictionary – dict\_order – {name:counter} – {‘space’:0}
   3. Dictionary – dict\_value – {counter:[key down time, key up time]} – {0:[0.0, 0.0]}
   4. Main tree dictionary
   5. Key list which is storing words and will be emptied when words pass to dictionary creation function
4. Thread listener accept the arguments
   1. \_\_init\_\_ start the thread, no explicit calls to start() required
   2. Printing event name and time
   3. Checking the event type using status
   4. If key down
      1. Finding epoch time using time stamp of event generation
      2. Incrementing counter
      3. Storing {name:counter} to dict\_order
      4. Storing {counter:[key down time]} to dict\_value
   5. If key up
      1. Finding epoch time using time stamp of event generation
      2. Count = dict\_order[name] – name is from key up event, both key up and key down name should be same for a particular key
      3. Appending value to dict\_Value list – dict\_value[count].append(key up time)