|  |  |
| --- | --- |
| **import** time print(time.gmtime(0)) *#1970* print() print(time.localtime()) print() print(time.time()) *#no of time the epoc has started* print(**"\*"**\*50) print(time.gmtime(0)) time\_here=time.localtime() print(time\_here) print(**"year : "**, time\_here[0], time\_here.tm\_year) print(**"month : "**, time\_here[1], time\_here.tm\_mon) print(**"day : "**, time\_here[2], time\_here.tm\_mday)  print(**"\*"**\*50) **import** time **from** time **import** time **as** my\_timer **import** random  input(**"press enter"**) wait\_tm=random.randint(1,6) time.sleep(wait\_tm) start\_tm= my\_timer() input(**"enter mar wara"**) end\_tm=my\_timer()  print(**"stsrted at : "** + time.strftime(**"%X"**, time.localtime(start\_tm))) *# strftime converts a string to date* print(**"ended at : "** + time.strftime(**"%X"**,time.localtime(end\_tm)))  print(**"elapsed : {}"**.format(end\_tm-start\_tm)) | time.struct\_time(tm\_year=1970, tm\_mon=1, tm\_mday=1, tm\_hour=0, tm\_min=0, tm\_sec=0, tm\_wday=3, tm\_yday=1, tm\_isdst=0)  time.struct\_time(tm\_year=2018, tm\_mon=2, tm\_mday=13, tm\_hour=9, tm\_min=6, tm\_sec=43, tm\_wday=1, tm\_yday=44, tm\_isdst=0)  1518502003.9003594  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  time.struct\_time(tm\_year=1970, tm\_mon=1, tm\_mday=1, tm\_hour=0, tm\_min=0, tm\_sec=0, tm\_wday=3, tm\_yday=1, tm\_isdst=0)  time.struct\_time(tm\_year=2018, tm\_mon=2, tm\_mday=13, tm\_hour=9, tm\_min=6, tm\_sec=43, tm\_wday=1, tm\_yday=44, tm\_isdst=0)  year : 2018 2018  month : 2 2  day : 13 13  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  press enter  enter mar wara  stsrted at : 09:08:53  ended at : 09:08:54  elapsed : 1.386667251586914 |
| **from** datetime **import** datetime s=datetime.today().second print(s) | 34 |
| **import** time **from** time **import** process\_time **as** timerN *#counts the time that is used to count the cpu process time* **import** random  input(**"press enter"**) wt=random.randint(1,4) time.sleep(wt) bt=timerN() input(**"Press Enter"**) et=timerN()  print(**"started at : "**+time.strftime(**"%X"**, time.localtime(bt))) print(**"ended at : "**+time.strftime(**"%X"**, time.localtime(et))) print(**"elapsed : {}"**.format(et-bt)) |  |
| **import** time **from** time **import** perf\_counter **as** timerN *#perf\_counter is always refered* **import** random *#* input(**"press enter"**) wt=random.randint(1,4) time.sleep(wt) bt=timerN() input(**"Press Enter"**) et=timerN()  print(**"started at : "**+time.strftime(**"%X"**, time.localtime(bt))) print(**"ended at : "**+time.strftime(**"%X"**, time.localtime(et))) print(**"elapsed : {}"**.format(et-bt)) |  |
| **import** time **from** time **import** monotonic **as** timerN **import** random  input(**"press enter"**) wt=random.randint(1,4) time.sleep(wt) bt=timerN() input(**"Press Enter"**) et=timerN()  print(**"started at : "**+time.strftime(**"%X"**, time.localtime(bt))) print(**"ended at : "**+time.strftime(**"%X"**, time.localtime(et))) print(**"elapsed : {}"**.format(et-bt)) |  |
| **import** time print(**"time:\t\t\t"**, time.get\_clock\_info(**'time'**)) print(**"perf\_counter:\t"**, time.get\_clock\_info(**'perf\_counter'**)) print(**"process\_time:\t"**, time.get\_clock\_info(**'process\_time'**)) print(**"monotonic:\t\t"**, time.get\_clock\_info(**'monotonic'**))  time: namespace(adjustable=True, implementation='GetSystemTimeAsFileTime()', monotonic=False, resolution=0.015625)  perf\_counter: namespace(adjustable=False, implementation='QueryPerformanceCounter()', monotonic=True, resolution=5.699551559283316e-07)  process\_time: namespace(adjustable=False, implementation='GetProcessTimes()', monotonic=True, resolution=1e-07)  monotonic: namespace(adjustable=False, implementation='GetTickCount64()', monotonic=True, resolution=0.015625) | |