Time module

|  |
| --- |
| **import** time  t1 = time.ctime() print(t1) # Sat Apr 23 00:00:52 2022 |

|  |
| --- |
| **import** time  l = time.localtime() print(l)  print(type(l)) # <class 'time.struct\_time'>  # time.struct\_time(tm\_year=2022, tm\_mon=4, tm\_mday=23, tm\_hour=9, tm\_min=53, tm\_sec=5, tm\_wday=5, tm\_yday=113, tm\_isdst=0)  tm\_year = year  tm\_mon = month  tm\_mday = monthday  tm\_hour = hour  tm\_min = minutes  tm\_sec = seconds  tm\_yday = yearday  tm\_isdst = daylight saving time |

|  |
| --- |
| **import** time  l = time.localtime() print(l.tm\_year, l.tm\_mon, l.tm\_mon) # 2022 4 23 print(type(l)) # <class 'time.struct\_time'> |

|  |
| --- |
| **import** time  l = time.localtime() print(l.tm\_isdst) # 0 print(type(l)) # <class 'time.struct\_time'>  # India is not using daylight saving time since 1970 - 2030 # 0 no DST # no dst changes for IST # 1 DST # change time to PST # -1 not known |

|  |
| --- |
| **import** time  l1 = time.localtime() print(type(l1)) # <class 'time.struct\_time'>  # Abbreviated weekday name r1 = time.strftime(**"%a"**, l1) print(r1) # Sat  # Full weekday name r2 = time.strftime(**"%A"**, l1) print(r2) # Saturday  # Full month name r3 = time.strftime(**"%B"**, l1) print(r3) # April  # Abbreviated month name r4 = time.strftime(**"%b"**, l1) print(r4) # Apr |

|  |
| --- |
| **import** time  l1 = time.localtime()  # Hour (24hr) possible values 00 - 23 r1 = time.strftime(**"%H"**, l1) print(r1) # 12  # Hour (12hr) possible values 01 - 12 r2 = time.strftime(**"%I"**, l1) print(r2) # 12  # Day of the year possible values 001 -366 r3 = time.strftime(**"%j"**, l1) print(r3) # 113  # Month of the year possible values 01 - 12 r4 = time.strftime(**"%m"**, l1) print(r4) # 04 |

|  |
| --- |
| **import** time  l1 = time.localtime()  # %p either AM or PM r1 = time.strftime(**"%p"**, l1) print(r1) # PM  # %w Weekday as decimal number , values from 0 to 6 r2 = time.strftime(**"%w"**, l1) print(r2) # 6  # %x for date representation r3 = time.strftime(**"%x"**, l1) print(r3) # 04/23/22  # %X for time representation r4 = time.strftime(**"%X"**, l1) print(r4) # 12:49:36  # %c date and time representation r5 = time.strftime(**"%c"**, l1) print(r5) # Sat Apr 23 12:49:36 2022  # %y year without date representation, values 00 -- 99 r6 = time.strftime(**"%y"**, l1) print(r6) # 22  # %Y year with century as decimal number r7 = time.strftime(**"%Y"**, l1) print(r7) # 2022  # %Z Time Zone Name r8 = time.strftime(**"%Z"**, l1) print(r8) # India Standard Time |

|  |
| --- |
| **import** datetime  # datetime.date(year, month, day)Creates a local date explicitly d = datetime.date(2021, 8, 20) print(d) # 2021-08-20 print(d.year, d.month, d.day) # 2021 8 20  # Today date -- returns the locale date implicitly print(d.today()) # 2022-04-23  2021-08-20  2021 8 20  2022-04-23 |

|  |
| --- |
| **import** datetime  t = datetime.datetime.today() print(t.day, t.month, t.year) # 23 4 2022 print(t.hour, t.minute, t.second) # 13 1 4 print(t.\_\_class\_\_) # <class 'datetime.datetime'> |

|  |
| --- |
| # from datetime import \* **import** datetime  #Set Year, Month, Day d = datetime.date(2021, 8, 30)  #Set Hour, Minute t = datetime.time(12,45) dt = datetime.datetime.combine(d, t) print(dt) # 2021-08-30 12:45:00 |

|  |
| --- |
| **from** datetime **import** date **import** time  d = []  d1 = date(2017, 8, 12) d2 = date(2017, 6, 12) d3 = date(2017, 4, 12) d4 = date(2017, 5, 12)  d.append(d1) d.append(d2) d.append(d3) d.append(d4)  d.sort(reverse=**False**) # Sort in sequence order**for** i **in** d:  print(i) |

|  |
| --- |
| **import** datetime  # Original date t = datetime.datetime.today()  # Replace date start = t.replace(year=1991, month=4, day=13) print(start) # 1991-04-13 14:17:46.042579  end = datetime.datetime(year=2022, month=4, day=30) print(end) # 2022-04-30 00:00:00  d = end - start print(d.days) # days is a property , 11339 |