

The Python Time Library

Keeping track of time can be very important when solving problems and creating applications using computers. The **Python time library** will allow you schedule tasks and synchronize elements of your programs easily.



Maybe you want a to create time limit for a video game or get events to occur at a specific time? The Python Time Library can help you with all of this.

Here are some simple examples of things you can do with the **time library** in Python:



Getting the Current Time:

You can use the `time()` function to retrieve the current timestamp, which represents the number of seconds since the **"epoch"** (January 1, 1970, 00:00:00 UTC). *See course page for video explanation. **Very important!!!!***



Exercise#1

Read the following code, **predict** what it might do, and **run** the in Replit to see what each does! Look at the code and output carefully.

```
import time

current_time = time.time()
print("")
print(f"Current timestamp: {current_time}")
print("")
print("What is this number mean? Look up python Epoch time to find out")
print("")
print(time.asctime())
print("")
print("Do you know what asctime() function does? Find out!")
```

Exercise#1 continued...

Read the following code, **predict** what it might do, and **run** the in Replit to see what each does! Look at the code and output carefully.

Sleeping:

The `sleep()` function allows you to pause the execution of your program for a specified number of seconds. You have likely used this one before.

```
import time
print("Before sleep")
time.sleep(3) # Sleep for 3 seconds
print("After sleep")
```

Creating a Timer

Here we will measure how long the user takes to type in their name by subtracting the **start** time from an **end** time.

```
import time
start = time.time()

name=input('give me your name: ')

end = time.time()

count=end-start
print(f'Hey {name} you took {count} seconds to type in your name')
print(f'Hey {name} you took {count:.2f} seconds (rounded to two decimal places) to type in your name')
```

Exercise 2:

Read the following code, **predict** what it might do, and **run** the in Replit to see what each does! Look at the code and output carefully.

Notice how you can extract the day, month, year, hour, minute, and seconds from the current date-time.

```
import time
print("")
a=time.asctime() # Grab the date and time
print(a)
print("")

print(a[-4:])    # the date is a string so grab the last 4 characters
print("")

b=(a.split())   # turn the string into a list
print(b)
print("")

c=(b[3])        # grab the 4th item in the list
print(c)
print("")

d=c.split(":")  # turn the 4th time into a list of it's own
print(d)
print("")

e=(d[0])        # print the first element of the list
print(e)
```

Notice how you can extract the **day, month, year, hour, minute, and seconds from the current time.**

Exercise#2 Continued...

Write a Python program that displays the current **date** `time.asctime()` in the following format:

weekday, day of month, month, year

Example:

```
The Current date is: Wed Oct 11 2023
```

Exercise 3: Birthday Countdown

```
x = input("Enter the future date (e.g., 'Oct 05 2024'): ")
print(x)
y = datetime.strptime(x, '%b %d %Y') # takes a string and turns it into a time object
print (y)
print (y.month) # now we can extract specific elements from our time object
```

Use the code above (or your own method) to help you do the following: Write a Python program that takes a user's birthdate (in the format YYYY-MM-DD) as input and calculates the number of days, hours, and minutes remaining until their next birthday. You can use the `time.asctime()` function to get the current date and time.

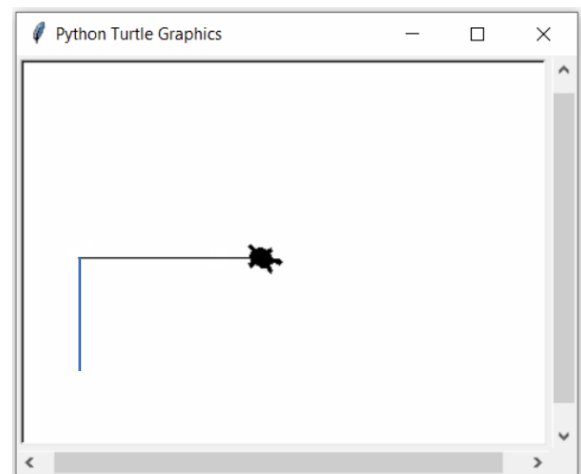


Exercise 3: Countdown Timer

Create a Python program that prompts the user to enter a number between 5 and 10 seconds and then counts down from that number to 0, with a one-second delay between each count. Display a fun message (or emoji or image) when the countdown reaches 0. Use may the `time.sleep()` function to introduce delays. You *must* clear the screen between each number that gets displayed. Have fun with this one.

Exercise 4: Code Execution Timer

Write a Python program that **measures the time taken in seconds for the user to complete a specific task**. You may use code from a previous exercise that allows the user to moving a turtle with arrow keys. Measure and display how long it takes the user to move the turtle from one spot to another on the screen (from A to B). Display the time in *seconds*. Pick another user task if you can think of something even more creative.



Exercise 5: Stopwatch

Create a *user friendly* and *visually appealing* stopwatch program that allows the user to start and stop a stopwatch timer using one of the 3 following ways (choose 1):



- By pressing a key (e.g., "S" for start, "Q" for quit). Display the elapsed time in seconds, updating it in real-time.
- Clicking on turtles that have **start and stop .png images as their shape**.
- If you want to get fancy, you can use the `tkinter` python library to create buttons and display text. Here is some starter code for tkinter:

<https://replit.com/@WSSjw/tkinter-example-stopwatch#main.py>

Exercise 6: Task Scheduler

Design a **task scheduler** that allows users to input tasks with due dates and times. The program should have the following features:

- Allows users to input tasks with due dates and times.
- Display the tasks in chronological order, with the soonest due at the top.
- Highlight tasks that are **overdue** and/or display a message to the user when they become overdue. This should be done by comparing the due date/time with the *actual* date/time.
- Allow the user to mark a task a done. Tasks marked done can't become "overdue" if completed by the due date/time.

See the sample user interface below (you don't have to stick to the following template, you can get as creative as you wish). You may want to use a **list of lists** as a structure for collecting and storing data. You can look this up.

Welcome to Task Scheduler!

Press:

- To see current task list
- To add a task to your list
- To mark a task as completed

Task	Due When	Done?	Alerts
Get gas for car	June 6 th at 10:45am		OVERDUE
Go to Soccer Practice	June 10 th at 9:00am	Done!!!!	
Buy gift for Edward	July 1 st at 3:00pm		