

Write an alarm clock program in Python

If you are new to Python programming, writing an alarm program is a simple example for you to practice.

There are many types of exercises where you can improve your Python programming skills. One of them is the exercise of writing a simple alarm program. After writing, you can run the alarm program with the command.

Write an alarm clock program in Python

First, you must create a Python script and add commands to display a question asking when the user wants to set the alarm. If you're unfamiliar with some Python syntax, you can look at some Python examples to understand the fundamentals.

1. Create a new file and name it alarm.py.
2. Open the Python script with any text editor, such as Notepad++.
3. At the top of the file, import both the datetime and time modules. The program will use these modules to calculate how long it has to wait for the alarm to ring. You can use Python's time module to delay execution as well as other actions.

```
import datetime import time
```

1. Add a while loop, requiring the user to enter a valid alarm ringing time, in the correct [hour:minute] format. The while loop will repeat if the user enters an incorrect number.

```
invalid = True
while(invalid):
    # Get a valid user input for the alarm time
    print("Set a valid time for the alarm (Ex. 06:30)")
    userInput = input(">> ")
```

1. In the while loop, convert the user input into an array that separates the hour value from the minute value.

```
## For example, this will convert 6:30 to an array of [6, 30].
alarmTime = [int(n) for n in userInput.split(":")]
```

1. Still in the while loop, check the hour value and minute value. The hour value must be between 0 and 23, and the minute value must be between 0 and 59. If these conditions are not met, the while loop is repeated, requiring the user to enter a value new.

```
# Validate the time entered to be between 0 and 24 (hours) or 0 and 60 (minutes)
if alarmTime[0] >= 24 or alarmTime[0] < 0:
    invalid = True
elif alarmTime[1] >= 60 or alarmTime[1] < 0:
    invalid = True
else:
    invalid = False
```

How to count the waiting time until the alarm rings

You create a counter for how many seconds the program will have to wait until the alarm goes off.

1. Below the while loop, convert the alarm time to the number of seconds in the day. For example, a day has 86,400 seconds. If the user enters 00:01 (one minute after midnight), the alarm time in seconds will be 60. If the user enters 23:59, the alarm time in seconds will be 86,340.

```
# Number of seconds in an Hour, Minute, and Second seconds_hms = [3600, 60, 1] #
```

1. Use the `datetime.now()` function to determine the current time. Convert current time to seconds:

```
now = datetime.datetime.now() currentTimeInSeconds = sum([a*b for a,b in zip(seconds_hms, now.time().split(':'))])
```

1. Calculate the number of seconds to wait until the alarm sounds:

```
secondsUntilAlarm = alarmSeconds - currentTimeInSeconds
```

1. If the difference between two time periods is negative, the alarm will be set for the next day:

```
if secondsUntilAlarm < 0: ????  
secondsUntilAlarm += 86400 # number of seconds in a day
```

1. Display a message to let the user know the alarm has been successfully set:

```
print("Alarm is set!") print("The alarm will ring at %s" % datetime.datetime.now().time())
```

How to make the alarm sound

To ring the alarm, wait for the last seconds to pass before printing the message "Wake up!" for users.

1. Use `time.sleep` to set the number of seconds to wait before the alarm sounds:

```
time.sleep(secondsUntilAlarm)
```

1. Show a notification to the user when the alarm goes off:

```
print("Ring Ring. time to wake up!")
```

How to add a countdown timer before the alarm goes off

1. Replace the `time.sleep` line. Add a for loop for each second until the bell rings and display the remaining seconds to the user:

```
for i in range(0, secondsUntilAlarm): ???time.sleep(1) ???  
secondsUntilAlarm -= 1 ???  
print(datetime.datetime.now().time())
```

How to run the alarm program

You run the alarm program by navigating to where the file is saved using the command line. Use the Python command to run the program and enter the alarm time.

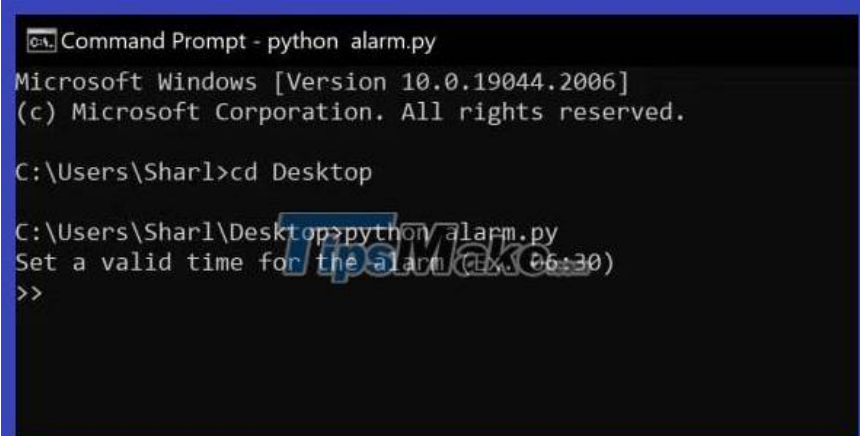
1. Open Command Prompt or Terminal. Navigate to where you saved the alarm.py file. For example, if you save a Python file on the Desktop, you use the cd Desktop command.

```
cd Desktop
```

1. Use Python command to run Python program:

```
python alarm.py
```

1. Set the exact timer time in [hour:minute] format, for example 4:30.



```
Command Prompt - python alarm.py
Microsoft Windows [Version 10.0.19044.2006]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sharl>cd Desktop

C:\Users\Sharl\Desktop>python alarm.py
Set a valid time for the alarm (EX: 06:30)
>>
```

TipsMake

1. The alarm clock will be set and start counting down every second until the bell rings. Wait until the alarm clock completes the countdown.

```
Command Prompt - python alarm.py
Microsoft Windows [Version 10.0.19044.2006]
(c) Microsoft Corporation. All rights reserved.


C:\Users\Sharl>cd Desktop

C:\Users\Sharl\Desktop>python alarm.py
Set a valid time for the alarm (Ex. 06:30)
>> 4:22
Alarm is set!
The alarm will ring at 0:00:19
0:00:18
0:00:17
0:00:16
0:00:15
0:00:14
0:00:13
```

TipsMake

1. When the countdown ends, the program will display a message letting you know that the bell is ringing.

```
Command Prompt
C:\Users\Shar1>cd Desktop
C:\Users\Shar1\Desktop>python alarm.py
Set a valid time for the alarm (Ex. 06:30)
>> 4:22
Alarm is set!
The alarm will ring at 0:00:19
0:00:18
0:00:17
0:00:16
0:00:15
0:00:14
0:00:13
0:00:12
0:00:11
0:00:10
0:00:09
0:00:08
0:00:07
0:00:06
0:00:05
0:00:04
0:00:03
0:00:02
0:00:01
0:00:00
Ring Ring... time to wake up!
```



TipsMake

Writing an alarm clock program in Python is a simple exercise that will help you improve your Python programming skills. TipsMake.com hopes that this article will be useful to you.

You finished reading the article "**Write an alarm clock program in Python**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.