Python time sleep(): Examples to add DELAY to your code

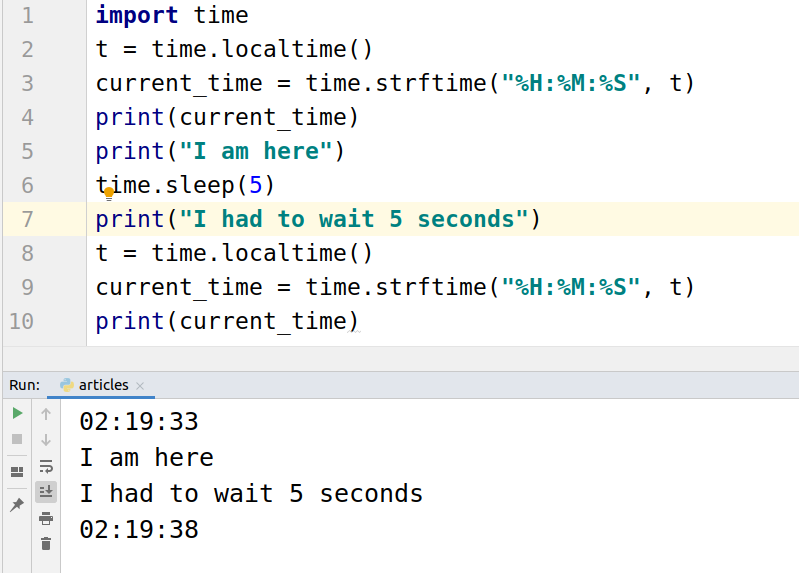
Sometimes you need to add delay to your specific code. Python has a built-in function sleep for such functionality. The sleep() function is part of the time module.

## Using sleep() function in Python

Let’s take a look at how to use the sleep function in Python. We will use the current time to check the time difference.

|  |
| --- |
| import time t = time.localtime() current\_time = time.strftime("%H:%M:%S", t) print(current\_time) print("I am here") time.sleep(5) print("I had to wait 5 seconds") t = time.localtime() current\_time = time.strftime("%H:%M:%S", t) print(current\_time) |

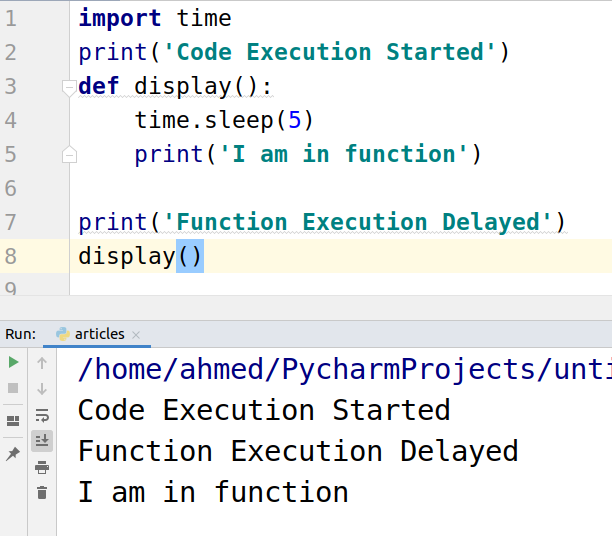
The following is the output.



## Delay function execution

Use the time.sleep function in the user-defined function to delay the execution of the function.

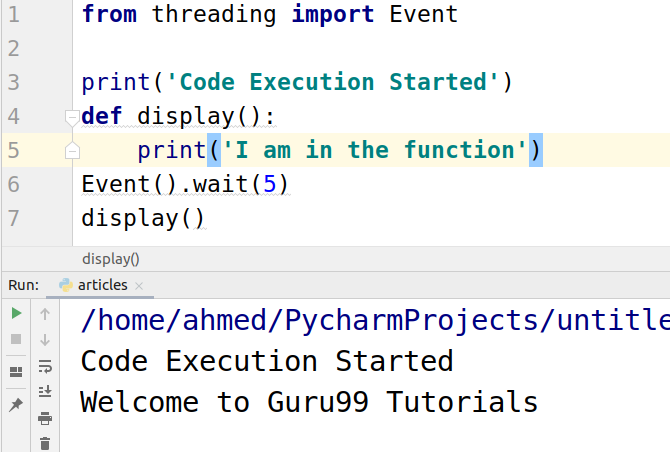
|  |
| --- |
| import time print('Code Execution Started') def display():  time.sleep(5)  print('I am in function')   print('Function Execution Delayed') display() |

]

## Adding delay using **Event().wait** function

In the threading library there is a function called event.wait() that can be used to add delay in the code.

|  |
| --- |
| from threading import Event  print('Code Execution Started') def display():  print('I am in the function') Event().wait(5) display() |



## Using Timer for delay

We can also use timer function to add delay to code.

|  |
| --- |
| from threading import Timer print('Code Execution Started') def display():  print('I am in the function') t = Timer(5, display) t.start() |

