Raspberry Pi for Beginners

LESSON 8

MAKERHOUSE

EMPOWERING MAKERS

TABLE OF CONTENTS

GOING DEEPER INTO PYTHON

1 INTRODUCTION

Starting from Lesson 5 until 7, we have learned a lot about Python. To add, we have been created a various text-based game using Python. It's such a wonderful feeling, right? We not only grabbed the gist of Python language but also implemented it in fun way through games. Congratulate yourself my dear friends.

Now you have the basics of Python yet become more advanced by going deeper into Python with this lesson; Lesson 8. Be prepared for explore more advanced concepts in the Python language—in particular, object-oriented Python, reading and writing files, handling, exceptions, using modules, and Internet programming

1.1 FORMATTING NUMBERS

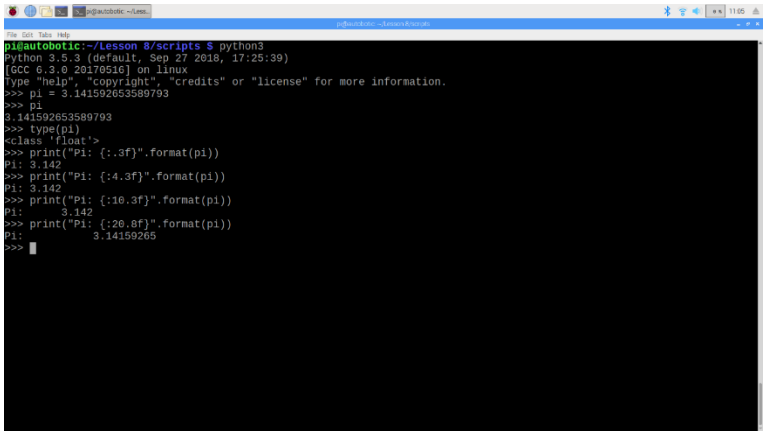


Figure 1: Formatting numbers to a certain number of decimal places.

1.2 FORMATTING DATES

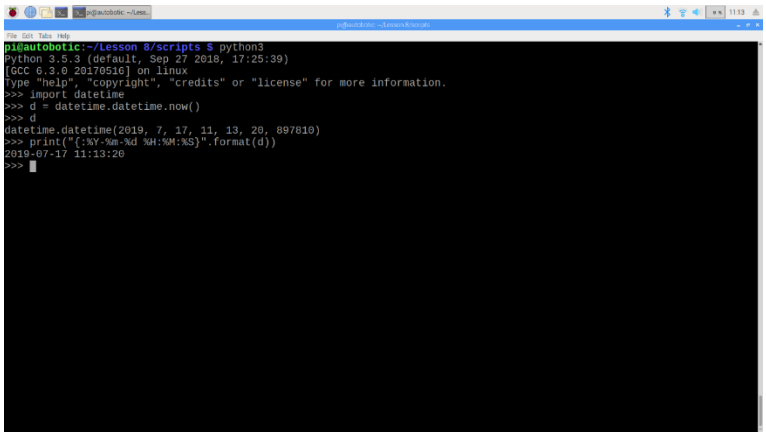
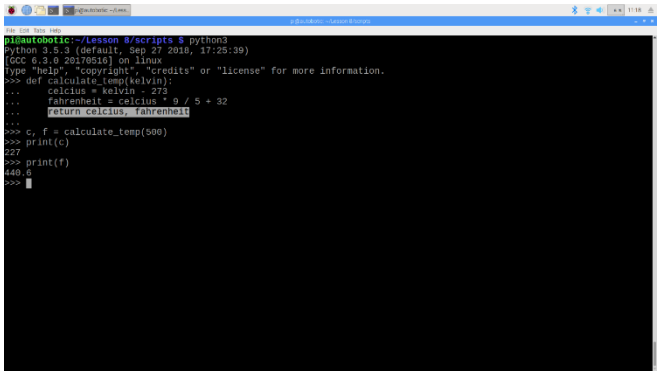


Figure 2: Converting a date into a string and format it in a certain way.

1.3 RETURNING MORE THAN ONE VALUES



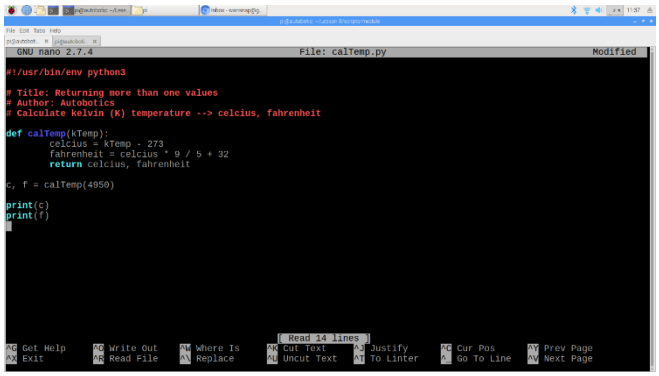
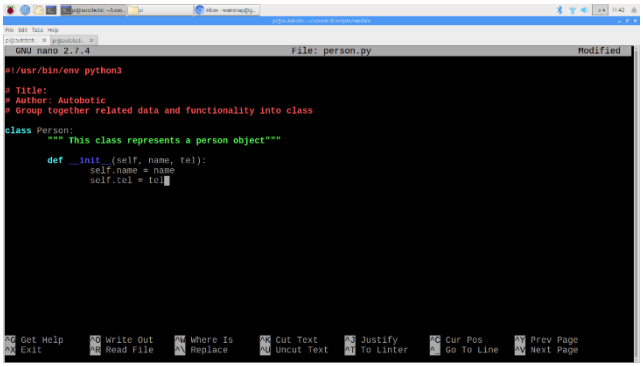


Figure 3: Writing a function that returns more than one value

1.4 DEFINING A CLASS



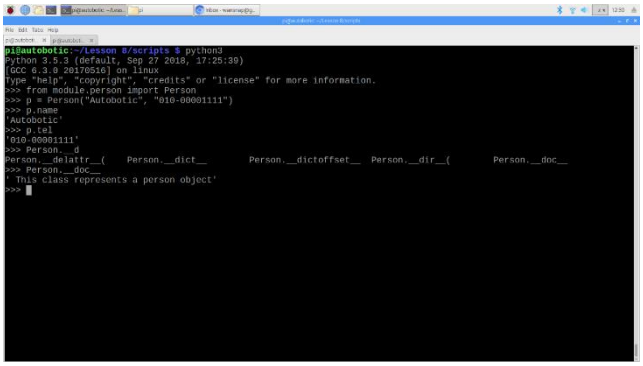


Figure 4: Grouping together related data and functionality into a class.

1.5 DEFINING A METHOD

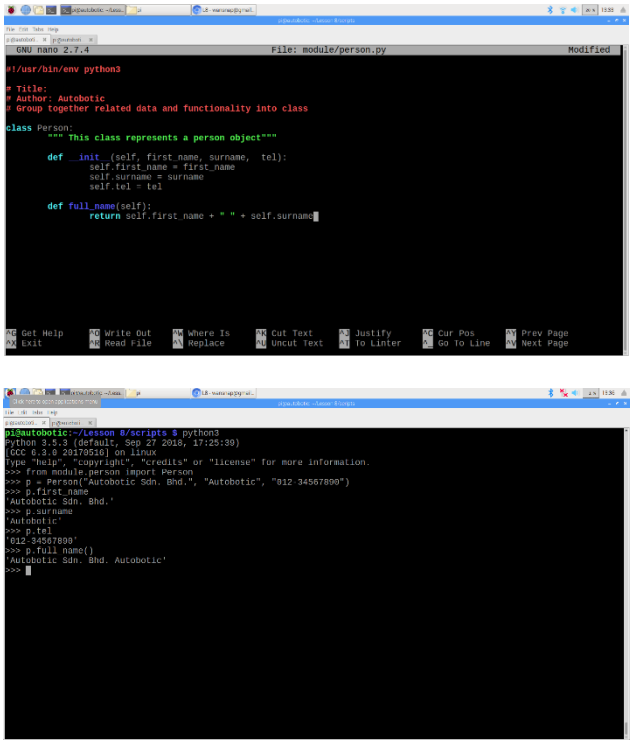


Figure 5: Adding a method to a class.

1.6 INHERITANCE

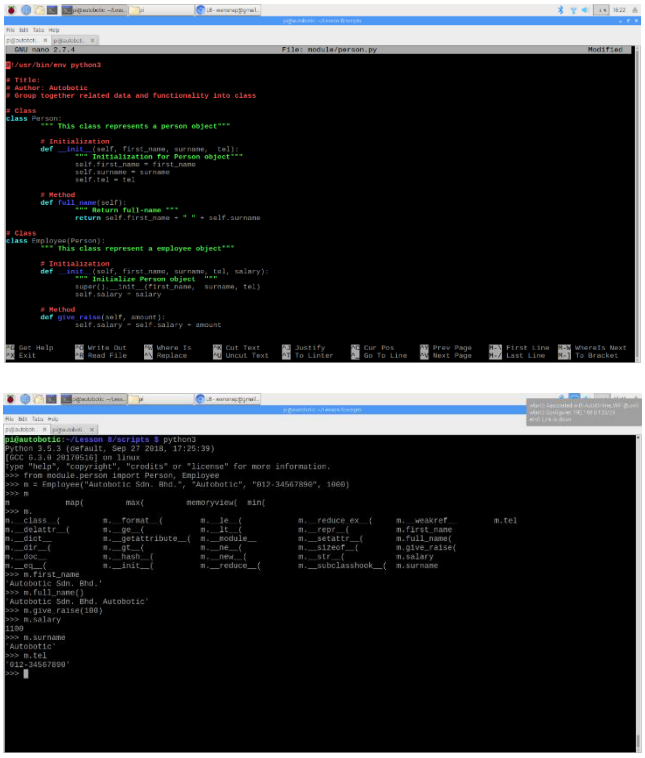


Figure 6: Specialized version of an existing class.

1.7 WRITING TO A FILE

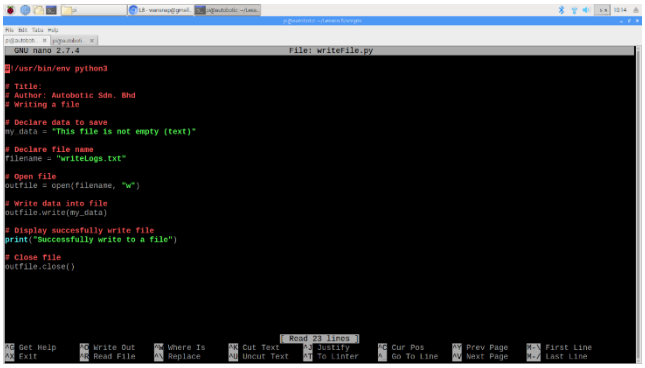


Figure 7: Writing something to a file.

1.8 READING FROM FILE

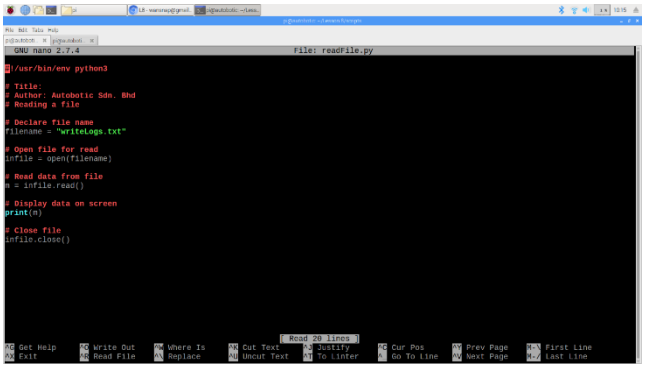


Figure 8: Reading the contents of a file into a string variable.

1.9 PICKLING

Another saving and reading entire contents of a data structure to/from a file -useful for the next time the program is run.

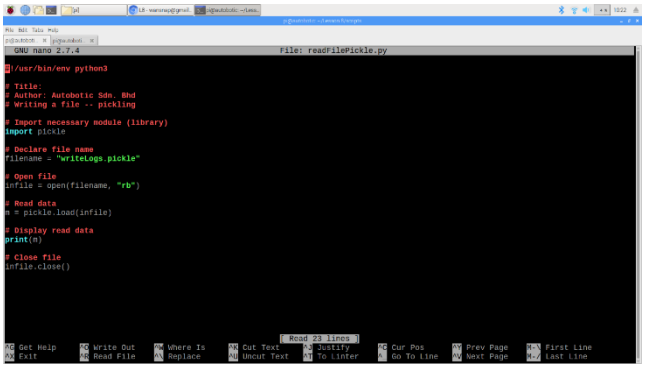
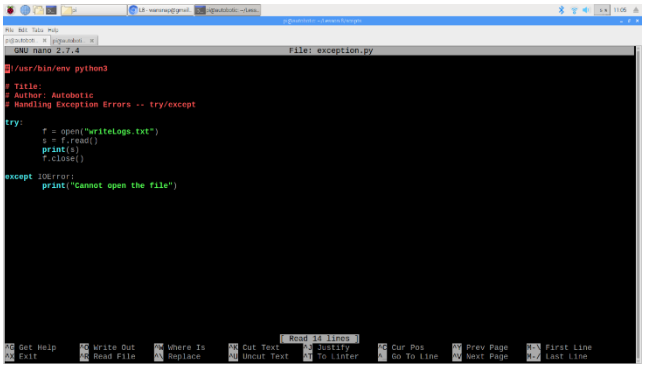
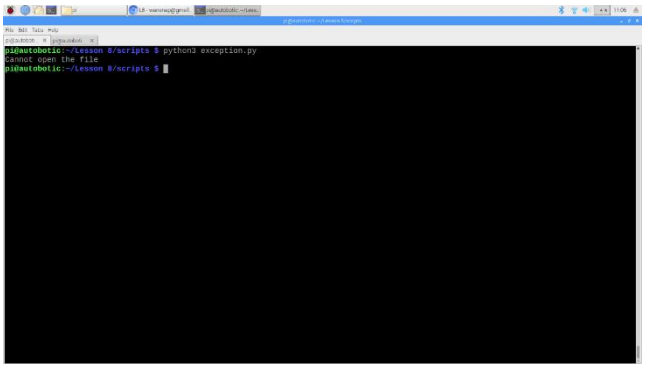


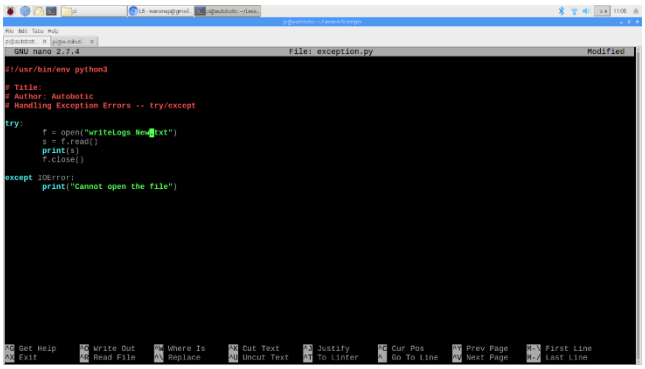
Figure 9: Reading and writing the contents of a file using pickle module.

1.10 HANDLING EXCEPTIONS

To catch the error or exception and display a more user-friendly error message -- use Python's **try/except** construct or may to have **else** and **finally** clauses in the error handling.







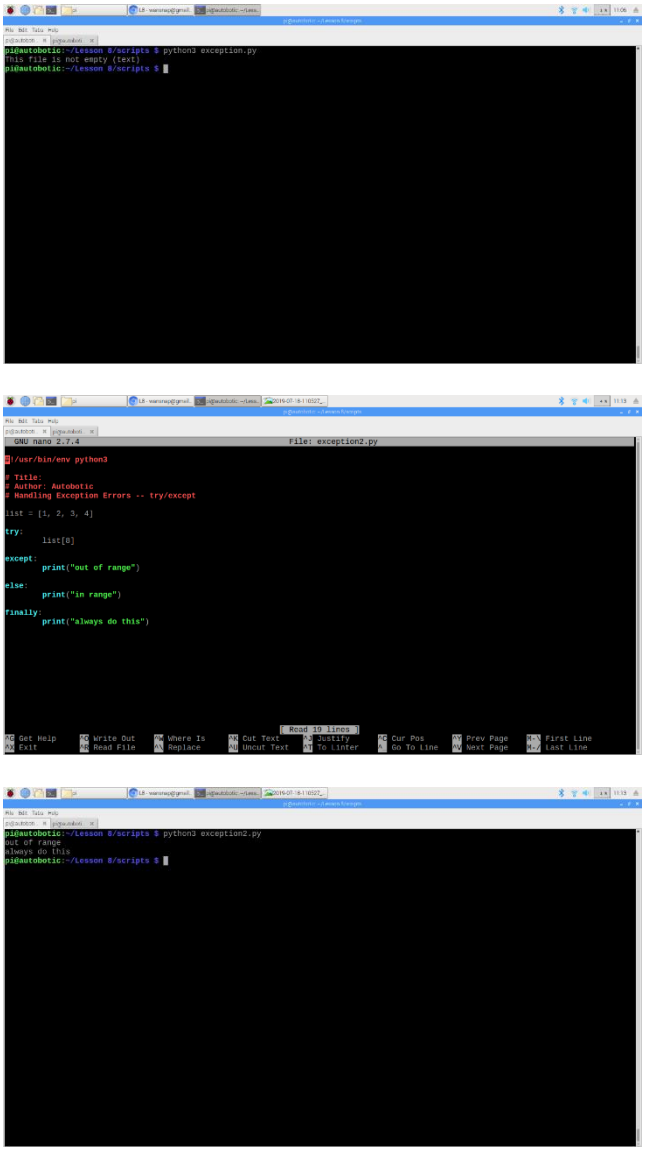
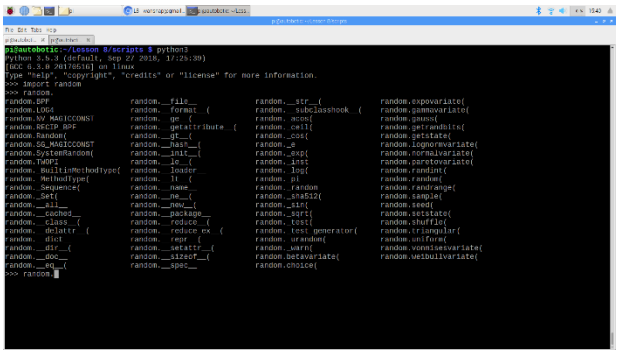


Figure 10: Reading and writing the contents of a file using pickle module.

1.11 USING MODULE

We eventually used various modules (libraries) – turtles, random – by using the **import** command.



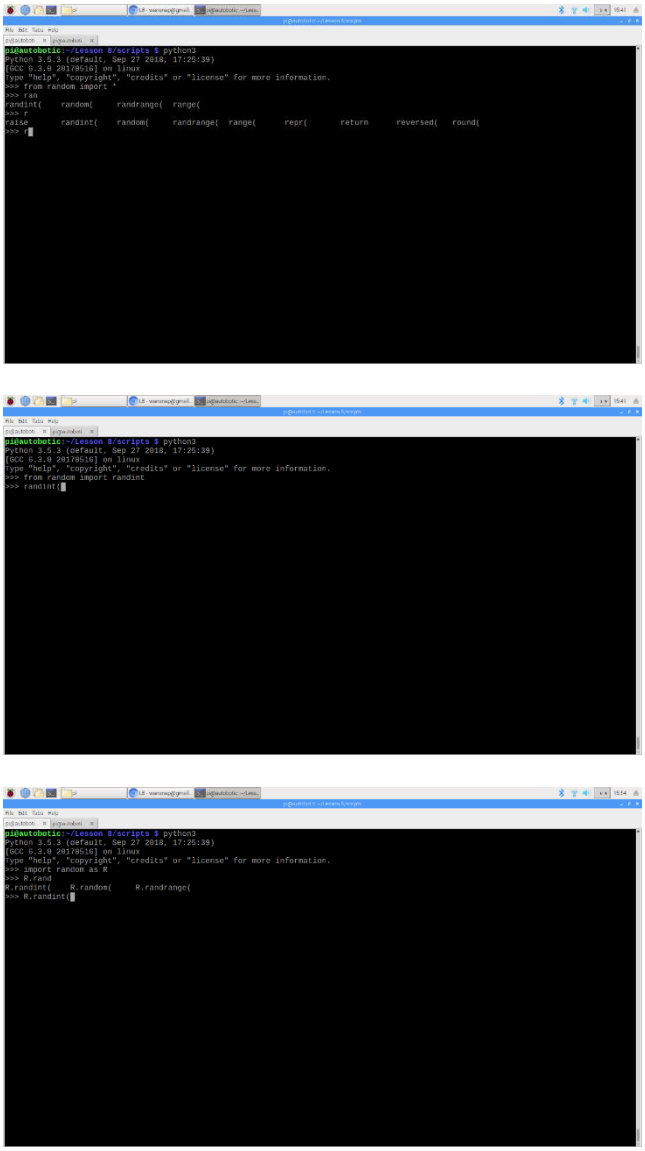


Figure 11: Various way to import (include) modules (libraries).

1.12 RANDOM NUMBERS

Generating a random thing such as number between a range of numbers by importing the random modules (libraries).

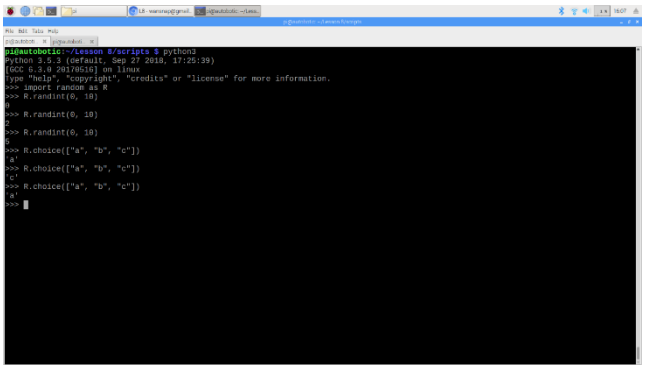


Figure 12: Importing the random module for random numbers or choice of selections.

2 CHALLENGE: A PYTHON GAME OF CAT AND MOUSE

This time we are going to make a game of cat and mouse. The player will control the mouse using the arrow keys and must stay ahead of the cat (controlled by the computer). The longer the mouse stays ahead, the higher the score it gets.

2.1 THE GAMES

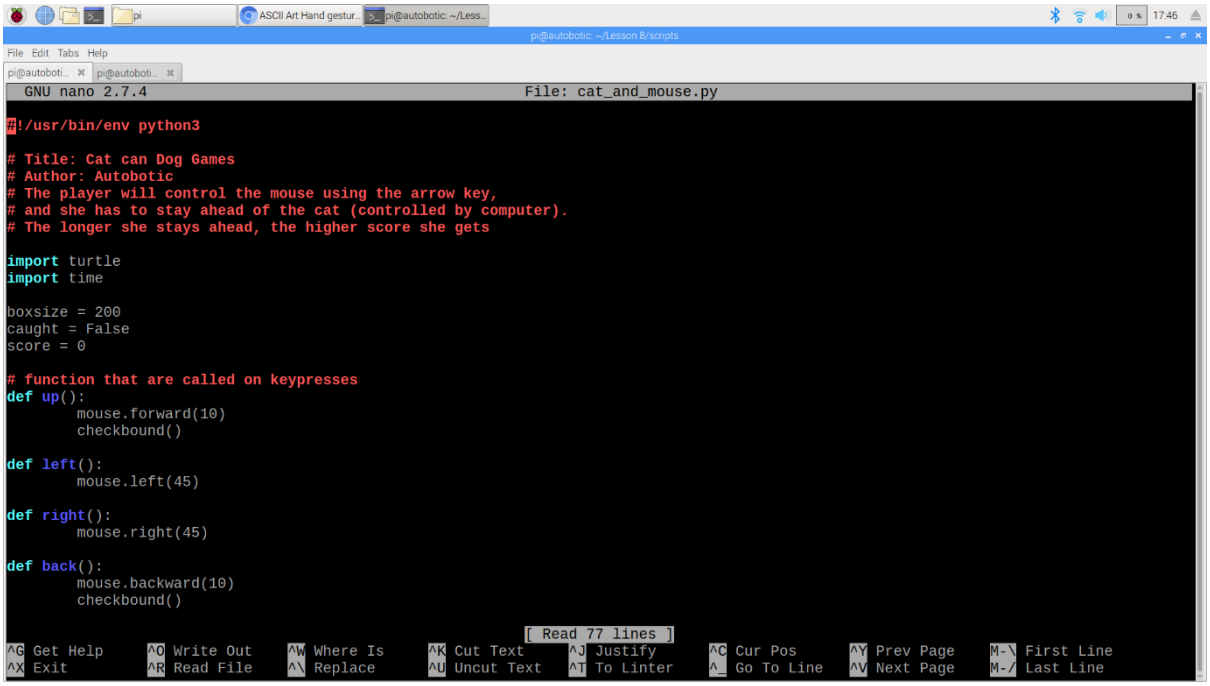


Figure 13: Scripting cat and mouse games in Python

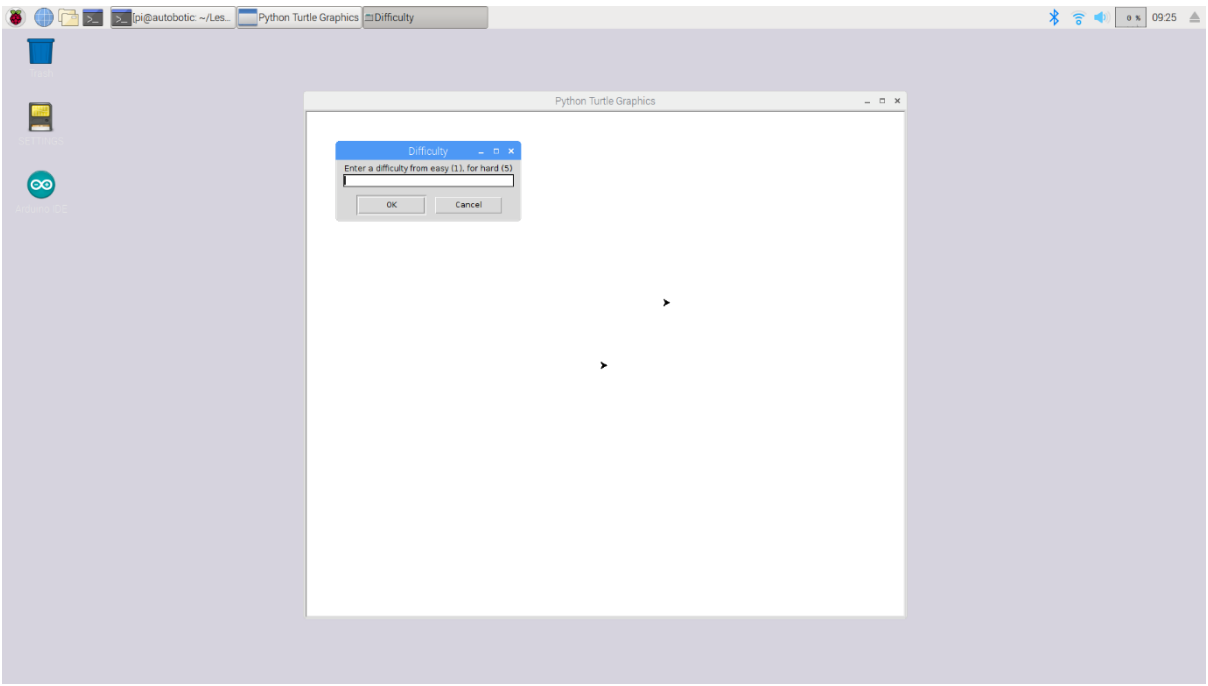


Figure 13: Cat and Dog Game Interface

2.1.1 THE FULL CODES

```
#!/usr/bin/env python3

# Title: Cat can Dog Games
# Author: Autobotic
# The player will control the mouse using the arrow key,
# and she has to stay ahead of the cat (controlled by computer).
# The longer she stays ahead, the higher score she gets

# importing important modules (libraries) – Turtle and time
import turtle
import time

# declare variable
boxsize = 200
caught = False
score = 0

# function that are called on keypresses
def up():
        mouse.forward(10)
        checkbound()

def left():
        mouse.left(45)

def right():
        mouse.right(45)

def back():
        mouse.backward(10)
        checkbound()

def quitTurtles():
        windows.bye()
```



```
# stop the mouse fromn leaving the square set by box size
def checkbound():
        global boxsize
        if mouse.xcor() > boxsize:
                mouse.goto(boxsize, mouse.ycor())
        elif mouse.xcor() < -boxsize:
        mouse.goto(-boxsize, mouse.ycor())
        elif mouse.ycor() > boxsize:
        mouse.goto(mouse.ycor(), boxsize)
        elif mouse.ycor() < -boxsize:
        mouse.goto(mouse.ycor(), -boxsize)

# set up screen
window = turtle.Screen()
mouse = turtle.Turtle()
cat = turtle.Turtle()
mouse.penup()
mouse.penup()
mouse.goto(100, 100)

# add key listeners
window.onkeypress(up, "Up")
window.onkeypress(left, "Left")
window.onkeypress(right, "Right")
window.onkeypress(back, "Down")
window.onkeypress(quitTurtles, "Escape")

difficulty = window.numinput("Difficulty", "Enter a difficulty from easy (1), for hard (5)", minval=1, maxval=5)

window.listen()

# main loop
# note how to changes with difficulty
while not caught:
```



```
        cat.setheading(cat.towards(mouse))
        cat.forward(8 + difficulty)
        score = score + 1

        if cat.distance(mouse) < 5:
                caught = True
        time.sleep(0.2 - (0.01 * difficulty))

window.textinput("GAME OVER", "Well done. You scored:" + str(score * difficulty))
window.bye()
```

Listing 1: Full code of cat and mouse games in Python

