Raspberry Pi for Beginners

LESSON 5

MAKERHOUSE

EMPOWERING MAKERS

TABLE OF CONTENTS

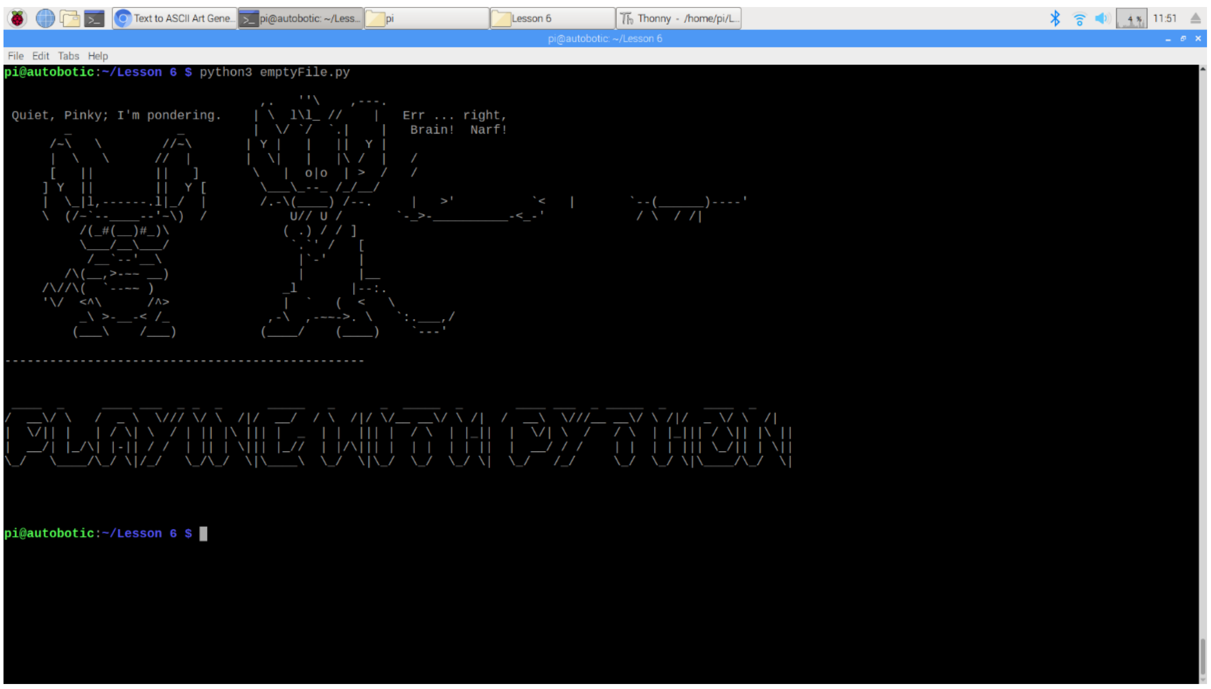
PLAYING WITH PYTHON

1 INTRODUCTION

All Python basics have been teaching in Lesson 5: Getting Starting with Python Basics. For continuity, we will create an original, fun, logical and interactive Python game with some additional information has to be added according to the requirements such as comments, etc.

1.1 GAMES

The fast way to understand and grab the programming skills is undeniably is through practice and implement it in real life. Creating a game is the right choice – logic programming; the computer will continue responding to user input – by making a simple text-based game.



```
pi@autobotic:~/Lesson 6 $ python3 emptyFile.py

 Quiet, Pinky; I'm pondering.                       Err ... right,
                                                     Brain!  Narf!

------------------------------------------------

pi@autobotic:~/Lesson 6 $
```

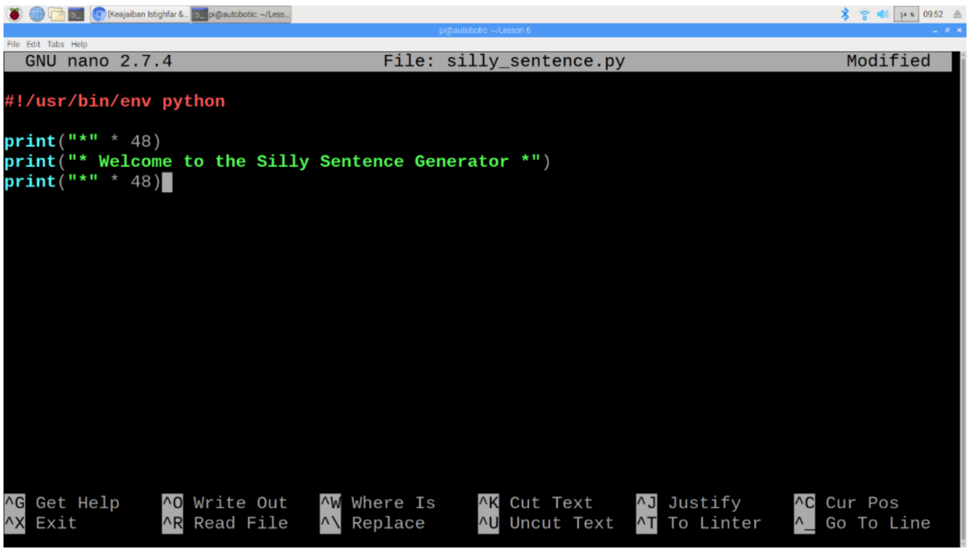
Figure 1: Text based game welcome page

2 SILLY SENTENCE GENERATOR

In interactive programs – "Silly Sentence Generator" – we will develop ridiculously fun word games. The game ideas are to collect information from users and interact with them in the same way they see every day on websites, mobile apps, and games.

2.1 WELCOME MESSAGE

Using the print function, you learned about in Lesson 5, let's make a welcome message:

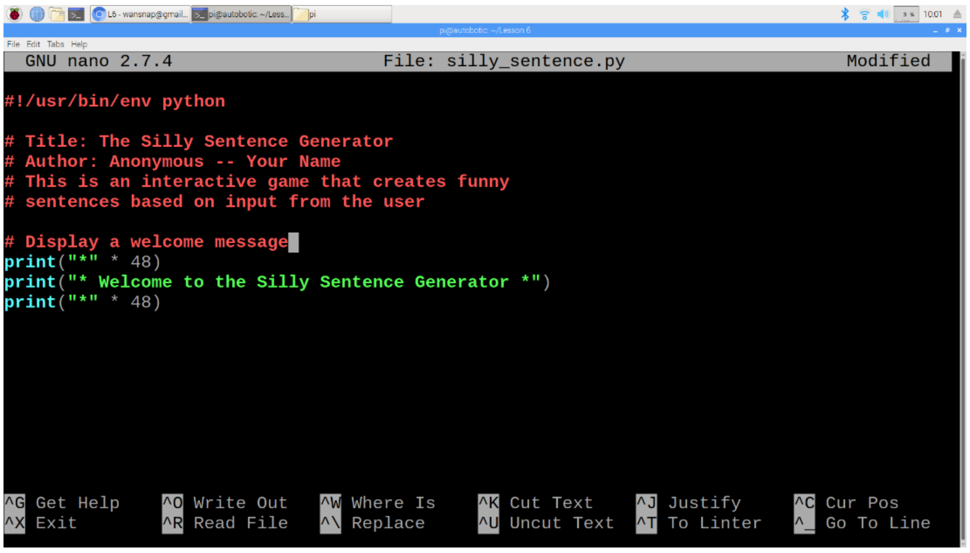


```
  GNU nano 2.7.4              File: silly_sentence.py                   Modified

#!/usr/bin/env python

print("*" * 48)
print("* Welcome to the Silly Sentence Generator *")
print("*" * 48)







^G Get Help     ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify      ^C Cur Pos
^X Exit         ^R Read File    ^\ Replace      ^U Uncut Text   ^T To Linter    ^_ Go To Line
```

Figure 2: Welcoming message

2.1.1 COMMENT (#)

Comic book without words: we know something is happening, but hard to tell without guessing. Comments – hashtags – were used to explain what's happening and for others who may read the code.



```
  GNU nano 2.7.4              File: silly_sentence.py                   Modified

#!/usr/bin/env python

# Title: The Silly Sentence Generator
# Author: Anonymous -- Your Name
# This is an interactive game that creates funny
# sentences based on input from the user

# Display a welcome message
print("*" * 48)
print("* Welcome to the Silly Sentence Generator *")
print("*" * 48)




^G Get Help     ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify      ^C Cur Pos
^X Exit         ^R Read File    ^\ Replace      ^U Uncut Text   ^T To Linter    ^_ Go To Line
```

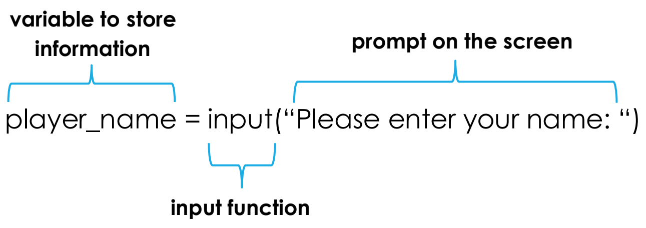
Figure 3: Helpful comment for code understanding

***** recommended to explain the program's title, its purpose, and who wrote it!***

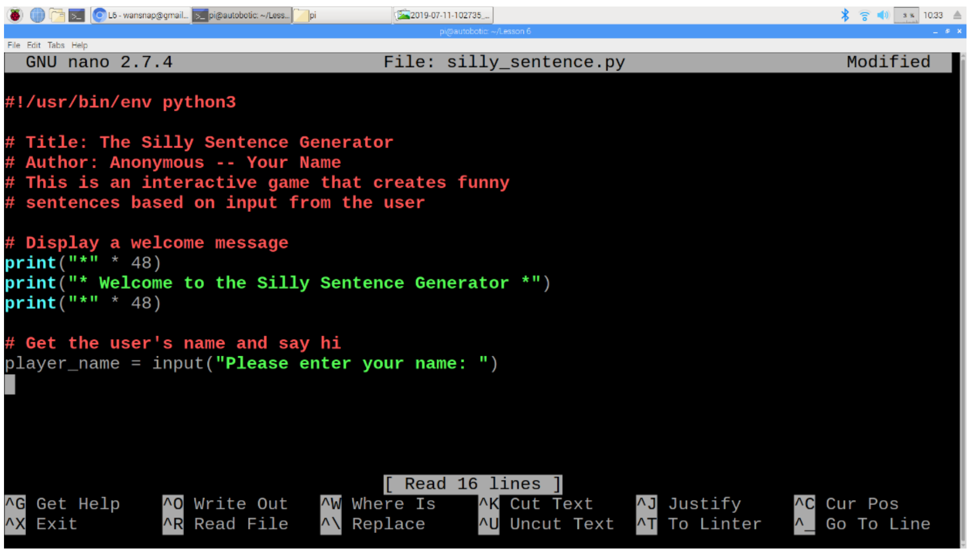
It is very for helpful reading the code; however, comments ignored as the program is executing.

2.2 GETTING AND STORING INFORMATION

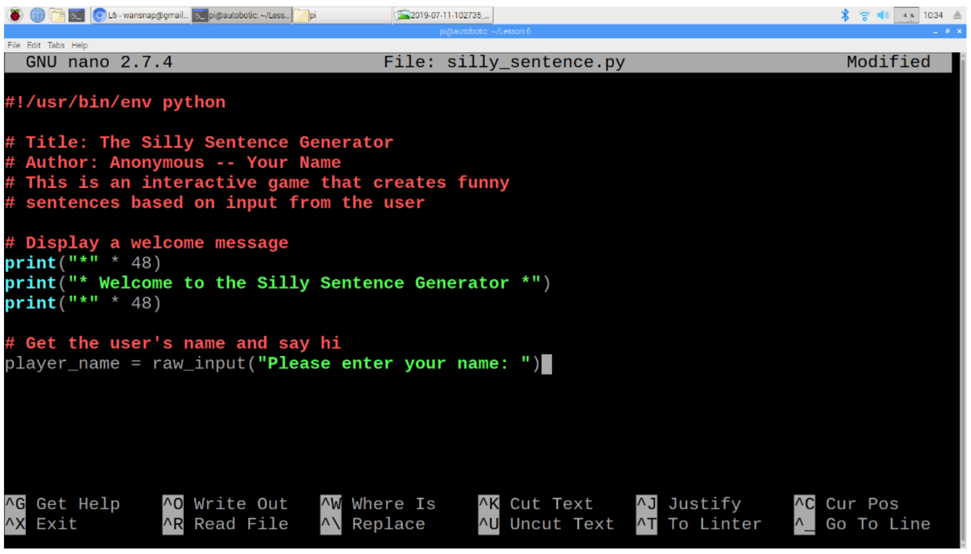
Using the input function, you learned about in Lesson 5 – it displays a prompt and awaits the user's reply. User is required to enters something and presses Enter.



```
player_name = input("Please enter your name: ")
```



```
  GNU nano 2.7.4              File: silly_sentence.py               Modified

#!/usr/bin/env python3

# Title: The Silly Sentence Generator
# Author: Anonymous -- Your Name
# This is an interactive game that creates funny
# sentences based on input from the user

# Display a welcome message
print("*" * 48)
print("* Welcome to the Silly Sentence Generator *")
print("*" * 48)

# Get the user's name and say hi
player_name = input("Please enter your name: ")



                              [ Read 16 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Linter  ^_ Go To Line
```



```
  GNU nano 2.7.4              File: silly_sentence.py               Modified

#!/usr/bin/env python

# Title: The Silly Sentence Generator
# Author: Anonymous -- Your Name
# This is an interactive game that creates funny
# sentences based on input from the user

# Display a welcome message
print("*" * 48)
print("* Welcome to the Silly Sentence Generator *")
print("*" * 48)

# Get the user's name and say hi
player_name = raw_input("Please enter your name: ")



^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Linter  ^_ Go To Line
```

Figure 4: Close looks – user input (input and raw_input) – different in Python 2 and Python 3

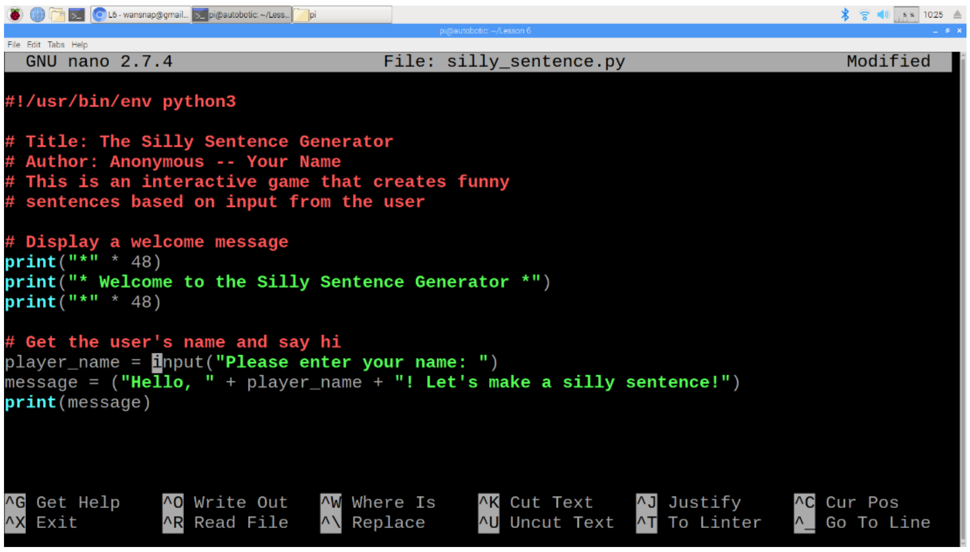
***** information is stored in the variable on the left side of the equals sign***

***** for python 2 replace input → raw_input***

2.3 JOINING STRING

The plus (+) symbol are used to join multiple strings.

$$\overbrace{\text{message}}^{\text{new variable to store information}} = (\text{"Hello, "} + \overbrace{\text{player\_name}}^{\text{Combining previous variable in new variable}} + \text{"! Let's make a silly sentence!"})$$



```
  GNU nano 2.7.4                File: silly_sentence.py                Modified

#!/usr/bin/env python3

# Title: The Silly Sentence Generator
# Author: Anonymous -- Your Name
# This is an interactive game that creates funny
# sentences based on input from the user

# Display a welcome message
print("*" * 48)
print("* Welcome to the Silly Sentence Generator *")
print("*" * 48)

# Get the user's name and say hi
player_name = input("Please enter your name: ")
message = ("Hello, " + player_name + "! Let's make a silly sentence!")
print(message)



^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Linter   ^_ Go To Line
```

Figure 5: Combining user input (strings)

***** string method – upper, lower, or capitalize may be used***

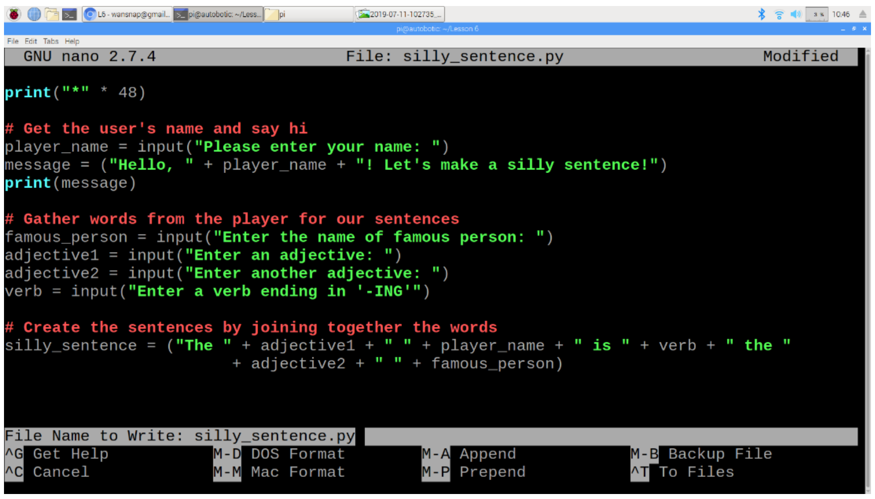
2.4 MORE USER INPUT

Funny sentence will be:

silly_sentence = ("The " + adjective1 + "" + player_name + " is " + verb + " the " + adjective2 + "" + famous_person)

that's means we required more input from user to complete the constructed funny sentence:

1. famous_person = input("Enter the name of a famous person: ")
2. adjective1 = input("Enter an adjective: ")
3. adjective2 = input("Enter another adjective: ")
4. verb = input("Enter a verb ending in –ING: ")

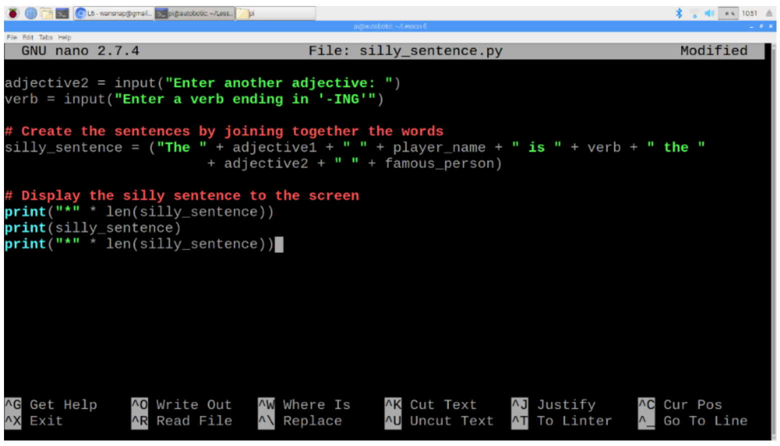


```
  GNU nano 2.7.4              File: silly_sentence.py                Modified

print("*" * 48)

# Get the user's name and say hi
player_name = input("Please enter your name: ")
message = ("Hello, " + player_name + "! Let's make a silly sentence!")
print(message)

# Gather words from the player for our sentences
famous_person = input("Enter the name of famous person: ")
adjective1 = input("Enter an adjective: ")
adjective2 = input("Enter another adjective: ")
verb = input("Enter a verb ending in '-ING'")

# Create the sentences by joining together the words
silly_sentence = ("The " + adjective1 + " " + player_name + " is " + verb + " the "
                        + adjective2 + " " + famous_person)


File Name to Write: silly_sentence.py
^G Get Help          M-D DOS Format       M-A Append           M-B Backup File
^C Cancel            M-M Mac Format       M-P Prepend          ^T To Files
```

Figure 6: More input; more funny!

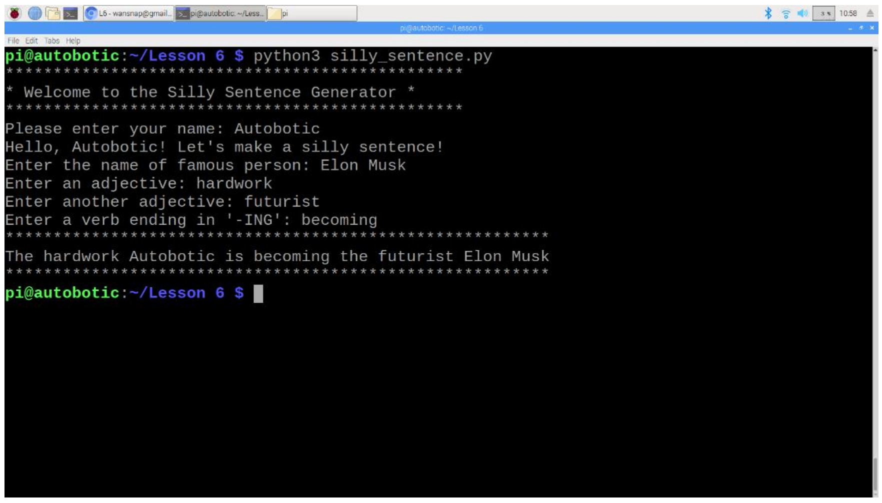
2.5 COMPLETING THE PROGRAM

Let's Python to show the complete sentence made with user own words to the screen.



```
  GNU nano 2.7.4              File: silly_sentence.py                Modified

adjective2 = input("Enter another adjective: ")
verb = input("Enter a verb ending in '-ING'")

# Create the sentences by joining together the words
silly_sentence = ("The " + adjective1 + " " + player_name + " is " + verb + " the "
                        + adjective2 + " " + famous_person)

# Display the silly sentence to the screen
print("*" * len(silly_sentence))
print(silly_sentence)
print("*" * len(silly_sentence))

^G Get Help   ^O Write Out   ^W Where Is   ^K Cut Text   ^J Justify   ^C Cur Pos
^X Exit       ^R Read File   ^\ Replace    ^U Uncut Text ^T To Linter ^_ Go To Line
```

Figure 6: Displaying the output – Silly Sentence output



```
pi@autobotic:~/Lesson 6 $ python3 silly_sentence.py
************************************************
* Welcome to the Silly Sentence Generator *
************************************************
Please enter your name: Autobotic
Hello, Autobotic! Let's make a silly sentence!
Enter the name of famous person: Elon Musk
Enter an adjective: hardwork
Enter another adjective: futurist
Enter a verb ending in '-ING': becoming
**********************************************************
The hardwork Autobotic is becoming the futurist Elon Musk
**********************************************************
pi@autobotic:~/Lesson 6 $
```

Figure 7: The game is ready!

3 CHALLENGE 1: ADDING LOGIC TO PROGRAMS

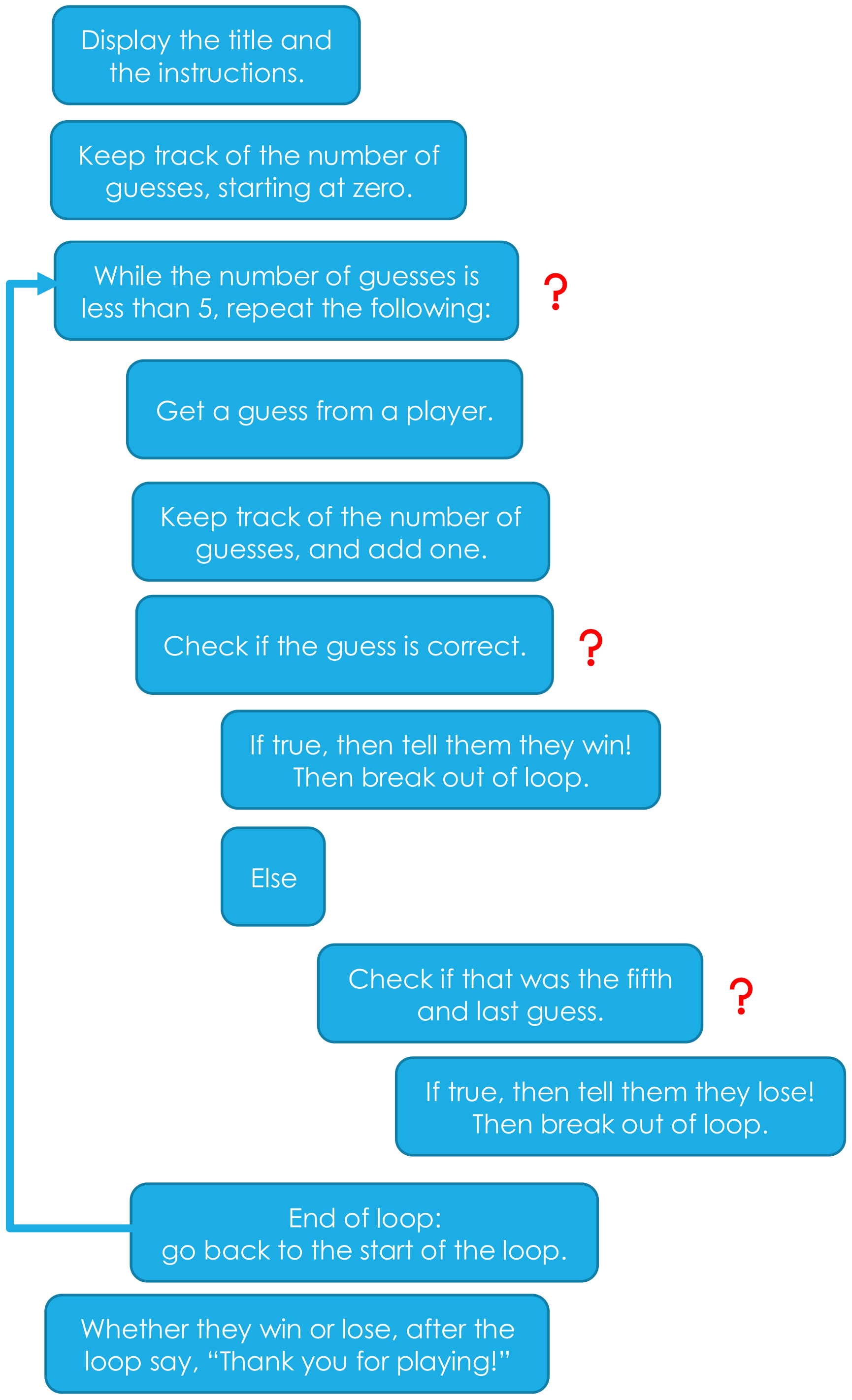
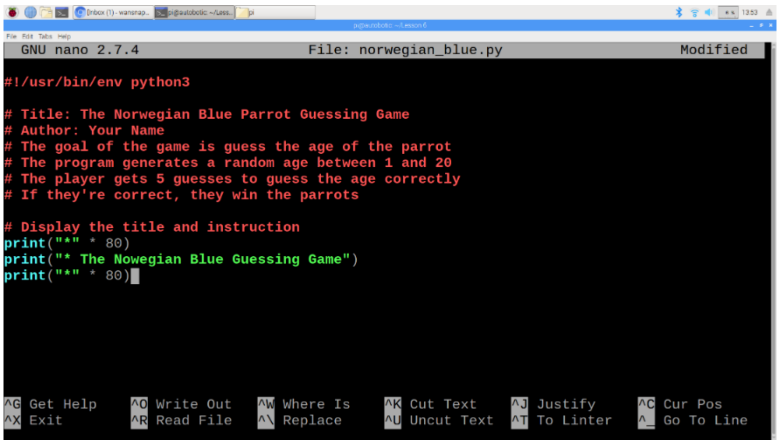

Display the title and the instructions.
Keep track of the number of guesses, starting at zero.
While the number of guesses is less than 5, repeat the following:
?
Get a guess from a player.
Keep track of the number of guesses, and add one.
Check if the guess is correct.
?
If true, then tell them they win! Then break out of loop.
Else
Check if that was the fifth and last guess.
?
If true, then tell them they lose! Then break out of loop.
End of loop: go back to the start of the loop.
Whether they win or lose, after the loop say, "Thank you for playing!"

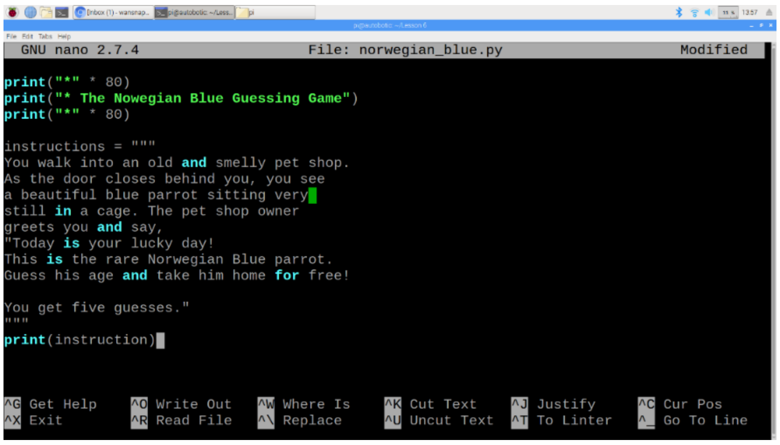
Figure 8: Bird age guessing – winners takes all

This game is about pretending you're visiting a pet shop that has a Norwegian Blue parrot for sale. The shop owner challenges you to guess the age of the parrot. If you guess correctly, then you get to take home the parrot for free.

3.1 GAME WELCOME AND INSTRUCTIONS



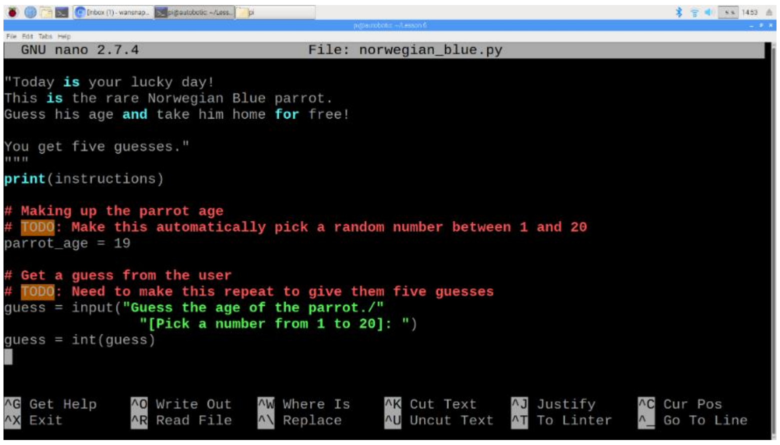
```
  GNU nano 2.7.4              File: norwegian_blue.py                    Modified

#!/usr/bin/env python3

# Title: The Norwegian Blue Parrot Guessing Game
# Author: Your Name
# The goal of the game is guess the age of the parrot
# The program generates a random age between 1 and 20
# The player gets 5 guesses to guess the age correctly
# If they're correct, they win the parrots

# Display the title and instruction
print("*" * 80)
print("* The Nowegian Blue Guessing Game")
print("*" * 80)

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Linter   ^_ Go To Line
```



```
  GNU nano 2.7.4              File: norwegian_blue.py                    Modified

print("*" * 80)
print("* The Nowegian Blue Guessing Game")
print("*" * 80)

instructions = """
You walk into an old and smelly pet shop.
As the door closes behind you, you see
a beautiful blue parrot sitting very
still in a cage. The pet shop owner
greets you and say,
"Today is your lucky day!
This is the rare Norwegian Blue parrot.
Guess his age and take him home for free!

You get five guesses."
"""
print(instruction)

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Linter   ^_ Go To Line
```

Figure 9: Create a welcome page for the guessing game

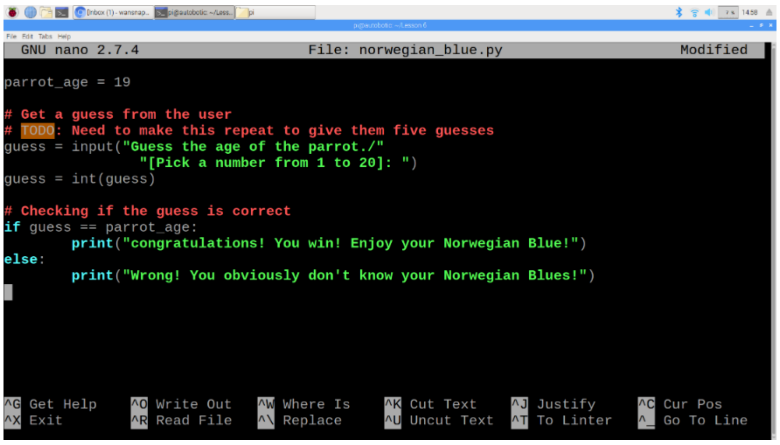
3.2 COLLECTING INPUT FROM PLAYER



```
  GNU nano 2.7.4              File: norwegian_blue.py

"Today is your lucky day!
This is the rare Norwegian Blue parrot.
Guess his age and take him home for free!

You get five guesses."
"""
print(instructions)

# Making up the parrot age
# TODO: Make this automatically pick a random number between 1 and 20
parrot_age = 19

# Get a guess from the user
# TODO: Need to make this repeat to give them five guesses
guess = input("Guess the age of the parrot./"
                "[Pick a number from 1 to 20]: ")
guess = int(guess)

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Linter   ^_ Go To Line
```

Figure 10: User input – guess – required

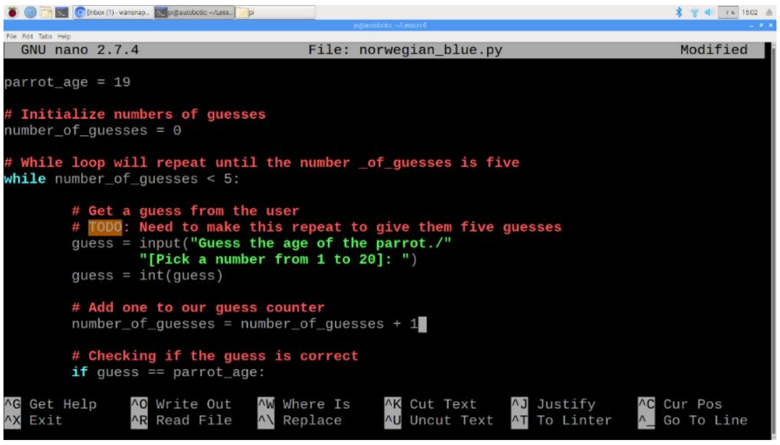
3.3 IF STATEMENTS TO RESPOND TO USERS IN DIFFERENT WAY



```
  GNU nano 2.7.4                 File: norwegian_blue.py                    Modified

parrot_age = 19

# Get a guess from the user
# TODO: Need to make this repeat to give them five guesses
guess = input("Guess the age of the parrot./"
                "[Pick a number from 1 to 20]: ")
guess = int(guess)

# Checking if the guess is correct
if guess == parrot_age:
        print("congratulations! You win! Enjoy your Norwegian Blue!")
else:
        print("Wrong! You obviously don't know your Norwegian Blues!")


^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Linter   ^_ Go To Line
```

Figure 11: Comparing using if condition – True/False

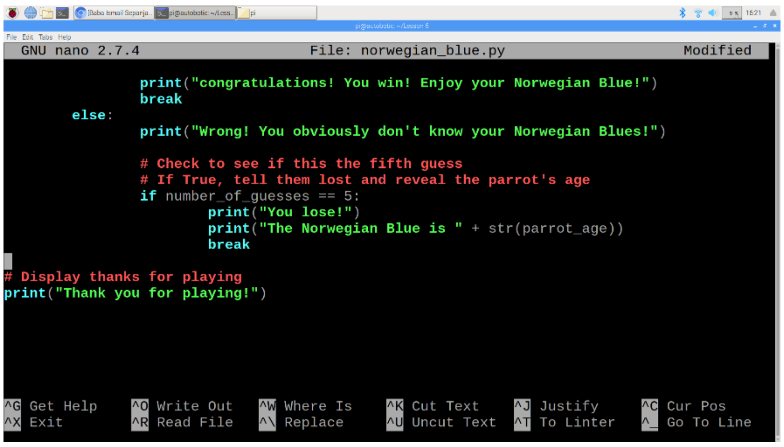
3.4 WHILE LOOPS TO REPEAT THINGS



```
  GNU nano 2.7.4                 File: norwegian_blue.py                    Modified

parrot_age = 19

# Initialize numbers of guesses
number_of_guesses = 0

# While loop will repeat until the number _of_guesses is five
while number_of_guesses < 5:

        # Get a guess from the user
        # TODO: Need to make this repeat to give them five guesses
        guess = input("Guess the age of the parrot./"
                "[Pick a number from 1 to 20]: ")
        guess = int(guess)

        # Add one to our guess counter
        number_of_guesses = number_of_guesses + 1

        # Checking if the guess is correct
        if guess == parrot_age:
^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Linter   ^_ Go To Line
```

Figure 12: Repeating loops - conditionally

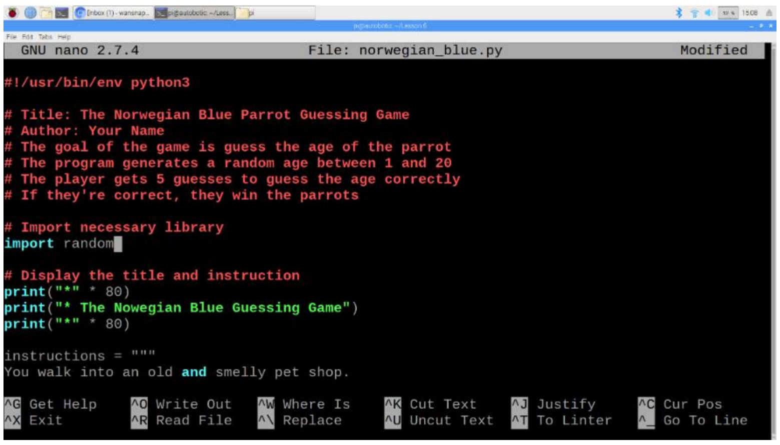
3.4.1 BREAK OUT FROM LOOPS



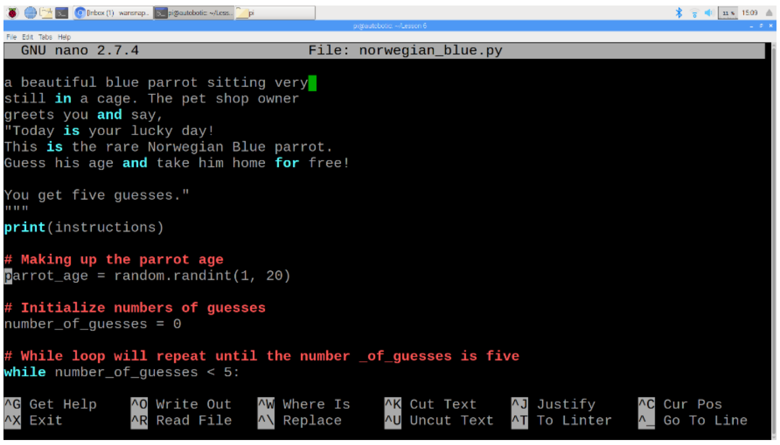
```
  GNU nano 2.7.4                 File: norwegian_blue.py                    Modified

                print("congratulations! You win! Enjoy your Norwegian Blue!")
                break
        else:
                print("Wrong! You obviously don't know your Norwegian Blues!")

                # Check to see if this the fifth guess
                # If True, tell them lost and reveal the parrot's age
                if number_of_guesses == 5:
                        print("You lose!")
                        print("The Norwegian Blue is " + str(parrot_age))
                        break

# Display thanks for playing
print("Thank you for playing!")


^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Linter   ^_ Go To Line
```

Figure 13: Break function – to get out from while loops

3.4.2 RANDOM MODULE



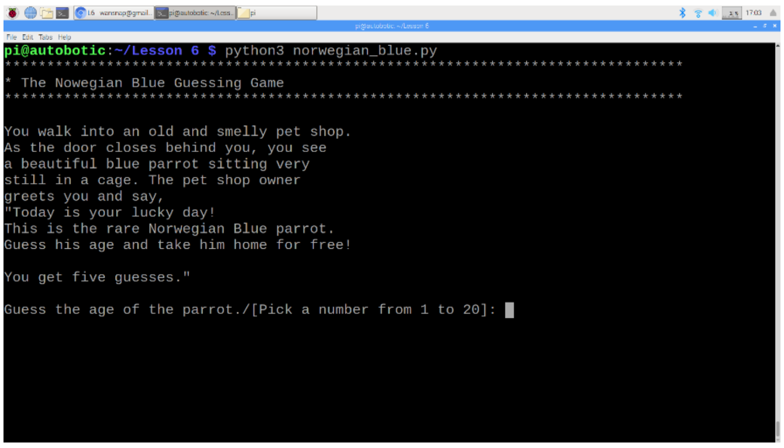
```
GNU nano 2.7.4                    File: norwegian_blue.py                    Modified

#!/usr/bin/env python3

# Title: The Norwegian Blue Parrot Guessing Game
# Author: Your Name
# The goal of the game is guess the age of the parrot
# The program generates a random age between 1 and 20
# The player gets 5 guesses to guess the age correctly
# If they're correct, they win the parrots

# Import necessary library
import random

# Display the title and instruction
print("*" * 80)
print("* The Nowegian Blue Guessing Game")
print("*" * 80)

instructions = """
You walk into an old and smelly pet shop.
^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Linter   ^_ Go To Line
```



```
GNU nano 2.7.4                    File: norwegian_blue.py

a beautiful blue parrot sitting very
still in a cage. The pet shop owner
greets you and say,
"Today is your lucky day!
This is the rare Norwegian Blue parrot.
Guess his age and take him home for free!

You get five guesses."
"""
print(instructions)

# Making up the parrot age
parrot_age = random.randint(1, 20)

# Initialize numbers of guesses
number_of_guesses = 0

# While loop will repeat until the number _of_guesses is five
while number_of_guesses < 5:
^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Linter   ^_ Go To Line
```

Figure 14: Randomize the parrat age each time game start – random module

3.5 COMPLETING THE PROGRAM



```
pi@autobotic:~/Lesson 6 $ python3 norwegian_blue.py
********************************************************************************
* The Nowegian Blue Guessing Game
********************************************************************************

You walk into an old and smelly pet shop.
As the door closes behind you, you see
a beautiful blue parrot sitting very
still in a cage. The pet shop owner
greets you and say,
"Today is your lucky day!
This is the rare Norwegian Blue parrot.
Guess his age and take him home for free!

You get five guesses."

Guess the age of the parrot./[Pick a number from 1 to 20]:
```

Figure 15: Playing the games

4 CHALLENGE 2: RASPI’S CAVE ADVENTURE

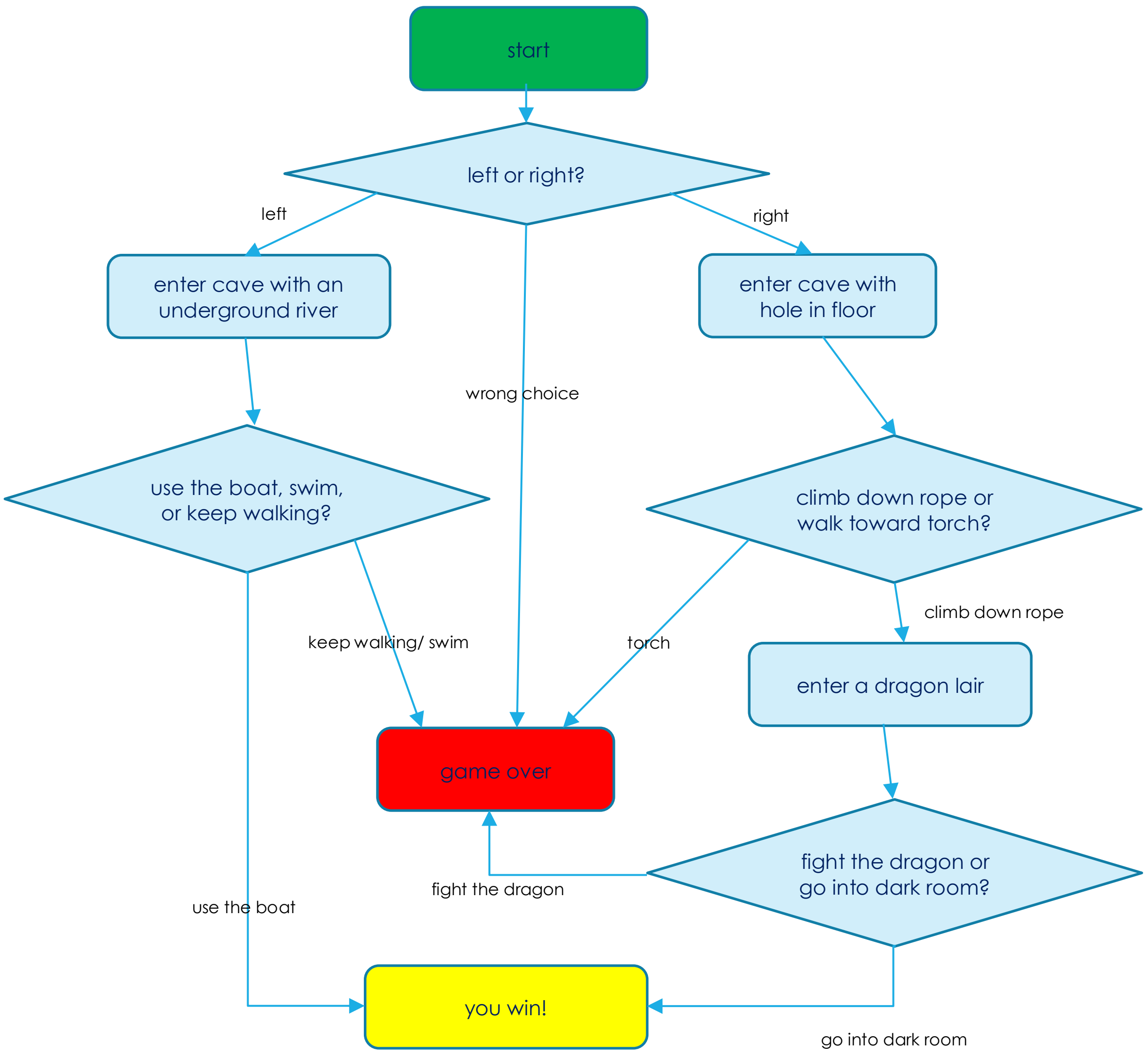

start
left or right?
left
right
enter cave with an underground river
enter cave with hole in floor
wrong choice
use the boat, swim, or keep walking?
climb down rope or walk toward torch?
climb down rope
keep walking/ swim
torch
enter a dragon lair
game over
fight the dragon
fight the dragon or go into dark room?
use the boat
you win!
go into dark room

Figure 16: Raspi’s Cave Adventure flowchart

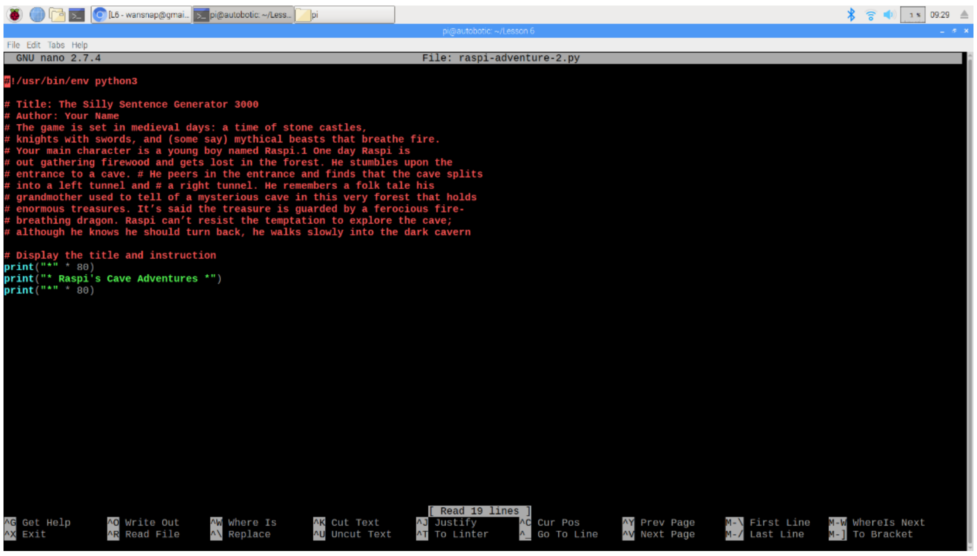
Left cave

If Raspi goes into the left cave, he’ll find himself near an underground river. He’ll need to decide whether to take a boat down the river, swim down the river, or walk along the side of the river. If Raspi decides to take the boat, he’ll soon learn that it has a hole in it, and he’ll sink (game over). Should Raspi choose to avoid the river and walk along its edge, he’ll quickly become distracted by his thoughts, trip on a rock, and hit his head (game over). If Raspi is adventurous and decides to swim in the river, he’ll make it to the other side and find a hidden trea- sure room filled with riches!

Right cave

If Raspi decides to go into the right cave, he’ll need to decide whether to climb down into a hole using a rope or walk toward what appears to be a torch. After walking toward the torch, Raspi will enter a cave full of crystals. The crystal cave sounds promising, but unfortunately a crystal will fall from the ceiling, ending Raspi’s life (game over). Alter- natively, if Raspi uses the rope and goes down the hole, he’ll find him- self in the dragon’s lair with a final choice: whether to fight the dragon or go into a dark room. If Raspi fights the dragon, the dragon will eat him; but if Raspi heads toward the dark room, he'll discover that it's filled with thousands of gold coins, rubies, and diamonds. Raspi is rich and very much alive!

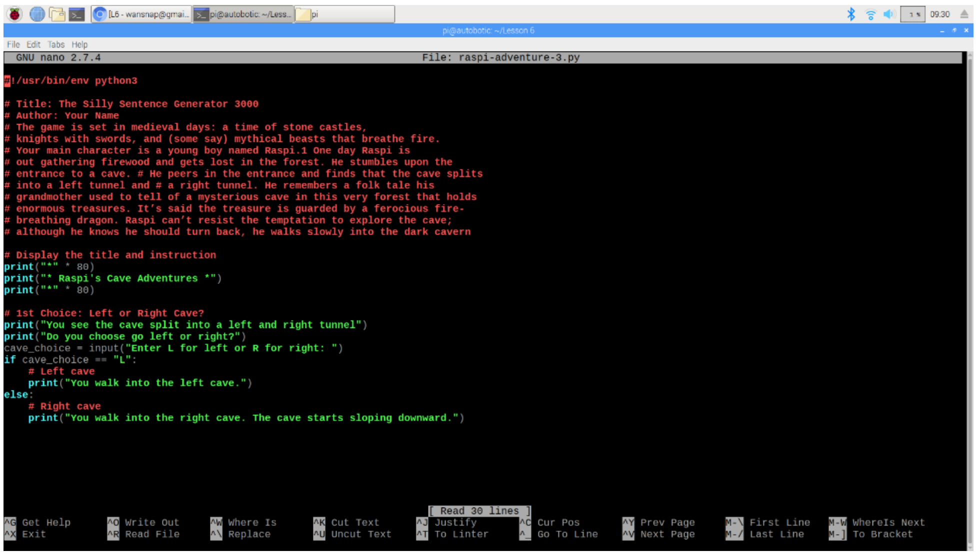
4.1 GAME WELCOME AND INSTRUCTIONS



```
  GNU nano 2.7.4                                     File: raspi-adventure-2.py

#!/usr/bin/env python3

# Title: The Silly Sentence Generator 3000
# Author: Your Name
# The game is set in medieval days: a time of stone castles,
# knights with swords, and (some say) mythical beasts that breathe fire.
# Your main character is a young boy named Raspi.1 One day Raspi is
# out gathering firewood and gets lost in the forest. He stumbles upon the
# entrance to a cave. # He peers in the entrance and finds that the cave splits
# into a left tunnel and # a right tunnel. He remembers a folk tale his
# grandmother used to tell of a mysterious cave in this very forest that holds
# enormous treasures. It’s said the treasure is guarded by a ferocious fire-
# breathing dragon. Raspi can’t resist the temptation to explore the cave;
# although he knows he should turn back, he walks slowly into the dark cavern

# Display the title and instruction
print("*" * 80)
print("* Raspi's Cave Adventures *")
print("*" * 80)

                                          [ Read 19 lines ]
^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos     ^Y Prev Page   M-\ First Line   M-W WhereIs Next
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Linter   ^_ Go To Line  ^V Next Page   M-/ Last Line    M-] To Bracket
```

Figure 17: Welcome to the adventure

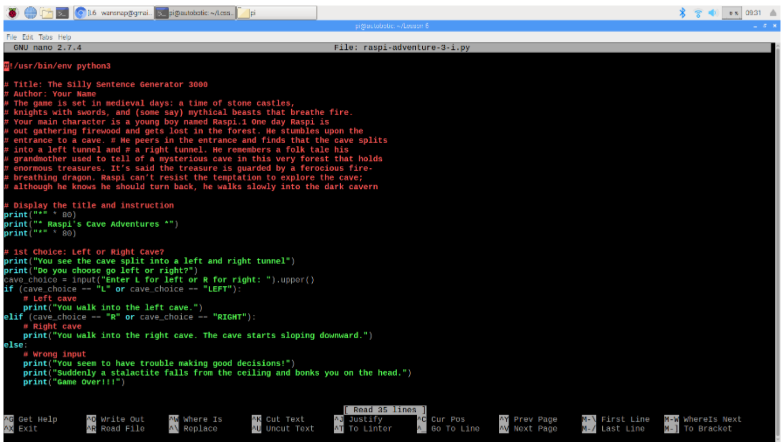
4.2 LEFT OR RIGHT CAVE?



```
  GNU nano 2.7.4                                     File: raspi-adventure-3.py

#!/usr/bin/env python3

# Title: The Silly Sentence Generator 3000
# Author: Your Name
# The game is set in medieval days: a time of stone castles,
# knights with swords, and (some say) mythical beasts that breathe fire.
# Your main character is a young boy named Raspi.1 One day Raspi is
# out gathering firewood and gets lost in the forest. He stumbles upon the
# entrance to a cave. # He peers in the entrance and finds that the cave splits
# into a left tunnel and # a right tunnel. He remembers a folk tale his
# grandmother used to tell of a mysterious cave in this very forest that holds
# enormous treasures. It’s said the treasure is guarded by a ferocious fire-
# breathing dragon. Raspi can’t resist the temptation to explore the cave;
# although he knows he should turn back, he walks slowly into the dark cavern

# Display the title and instruction
print("*" * 80)
print("* Raspi's Cave Adventures *")
print("*" * 80)

# 1st Choice: Left or Right Cave?
print("You see the cave split into a left and right tunnel")
print("Do you choose go left or right?")
cave_choice = input("Enter L for left or R for right: ")
if cave_choice == "L":
    # Left cave
    print("You walk into the left cave.")
else:
    # Right cave
    print("You walk into the right cave. The cave starts sloping downward.")

                                          [ Read 30 lines ]
^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos     ^Y Prev Page   M-\ First Line   M-W WhereIs Next
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Linter   ^_ Go To Line  ^V Next Page   M-/ Last Line    M-] To Bracket
```

Figure 18: Your choice?

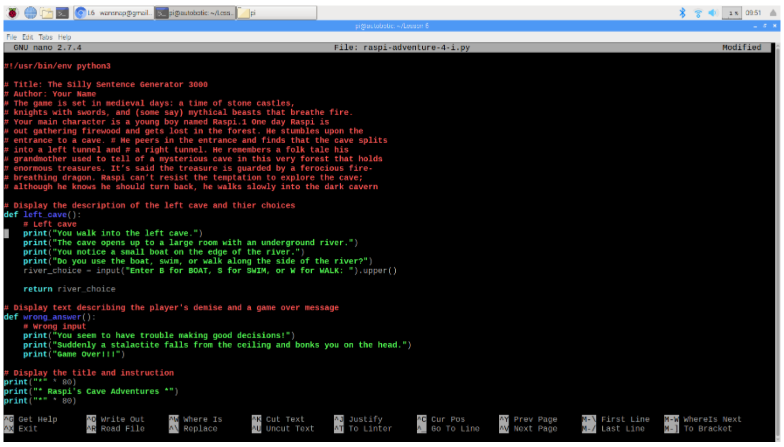
4.2.1 WRONG INPUT



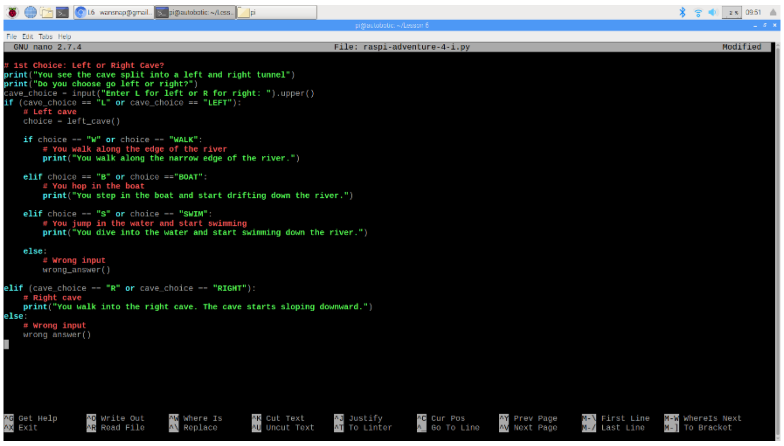
```
  GNU nano 2.7.4                    File: raspi-adventure-3-i.py

#!/usr/bin/env python3

# Title: The Silly Sentence Generator 3000
# Author: Your Name
# The game is set in medieval days: a time of stone castles,
# knights with swords, and (some say) mythical beasts that breathe fire.
# Your main character is a young boy named Raspi.1 One day Raspi is
# out gathering firewood and gets lost in the forest. He stumbles upon the
# entrance to a cave. # He peers in the entrance and finds that the cave splits
# into a left tunnel and # a right tunnel. He remembers a folk tale his
# grandmother used to tell of a mysterious cave in this very forest that holds
# enormous treasures. It's said the treasure is guarded by a ferocious fire-
# breathing dragon. Raspi can't resist the temptation to explore the cave;
# although he knows he should turn back, he walks slowly into the dark cavern

# Display the title and instruction
print("*" * 80)
print("* Raspi's Cave Adventures *")
print("*" * 80)

# 1st Choice: Left or Right Cave?
print("You see the cave split into a left and right tunnel")
print("Do you choose go left or right?")
cave_choice = input("Enter L for left or R for right: ").upper()
if (cave_choice == "L" or cave_choice == "LEFT"):
    # Left cave
    print("You walk into the left cave.")
elif (cave_choice == "R" or cave_choice == "RIGHT"):
    # Right cave
    print("You walk into the right cave. The cave starts sloping downward.")
else:
    # Wrong input
    print("You seem to have trouble making good decisions!")
    print("Suddenly a stalactite falls from the ceiling and bonks you on the head.")
    print("Game Over!!!")

                                [ Read 35 lines ]
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text    ^J Justify   ^C Cur Pos     ^Y Prev Page  M-\ First Line  M-W WhereIs Next
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text  ^T To Linter ^_ Go To Line  ^V Next Page  M-/ Last Line   M-] To Bracket
```

Figure 19: User accidentally entering wrong input – different from available selection

4.3 FUNCTION



```
  GNU nano 2.7.4                    File: raspi-adventure-4-i.py                    Modified

#!/usr/bin/env python3

# Title: The Silly Sentence Generator 3000
# Author: Your Name
# The game is set in medieval days: a time of stone castles,
# knights with swords, and (some say) mythical beasts that breathe fire.
# Your main character is a young boy named Raspi.1 One day Raspi is
# out gathering firewood and gets lost in the forest. He stumbles upon the
# entrance to a cave. # He peers in the entrance and finds that the cave splits
# into a left tunnel and # a right tunnel. He remembers a folk tale his
# grandmother used to tell of a mysterious cave in this very forest that holds
# enormous treasures. It's said the treasure is guarded by a ferocious fire-
# breathing dragon. Raspi can't resist the temptation to explore the cave;
# although he knows he should turn back, he walks slowly into the dark cavern

# Display the description of the left cave and thier choices
def left_cave():
    # Left cave
    print("You walk into the left cave.")
    print("The cave opens up to a large room with an underground river.")
    print("You notice a small boat on the edge of the river.")
    print("Do you use the boat, swim, or walk along the side of the river?")
    river_choice = input("Enter B for BOAT, S for SWIM, or W for WALK: ").upper()

    return river_choice

# Display text describing the player's demise and a game over message
def wrong_answer():
    # Wrong input
    print("You seem to have trouble making good decisions!")
    print("Suddenly a stalactite falls from the ceiling and bonks you on the head.")
    print("Game Over!!!")

# Display the title and instruction
print("*" * 80)
print("* Raspi's Cave Adventures *")
print("*" * 80)
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text    ^J Justify   ^C Cur Pos     ^Y Prev Page  M-\ First Line  M-W WhereIs Next
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text  ^T To Linter ^_ Go To Line  ^V Next Page  M-/ Last Line   M-] To Bracket
```



```
  GNU nano 2.7.4                    File: raspi-adventure-4-i.py                    Modified

# 1st Choice: Left or Right Cave?
print("You see the cave split into a left and right tunnel")
print("Do you choose go left or right?")
cave_choice = input("Enter L for left or R for right: ").upper()
if (cave_choice == "L" or cave_choice == "LEFT"):
    # Left cave
    choice = left_cave()

    if choice == "W" or choice == "WALK":
        # You walk along the edge of the river
        print("You walk along the narrow edge of the river.")

    elif choice == "B" or choice =="BOAT":
        # You hop in the boat
        print("You step in the boat and start drifting down the river.")

    elif choice == "S" or choice == "SWIM":
        # You jump in the water and start swimming
        print("You dive into the water and start swimming down the river.")

    else:
        # Wrong input
        wrong_answer()

elif (cave_choice == "R" or cave_choice == "RIGHT"):
    # Right cave
    print("You walk into the right cave. The cave starts sloping downward.")
else:
    # Wrong input
    wrong answer()

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text    ^J Justify   ^C Cur Pos     ^Y Prev Page  M-\ First Line  M-W WhereIs Next
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text  ^T To Linter ^_ Go To Line  ^V Next Page  M-/ Last Line   M-] To Bracket
```

Figure 20: Transform the long list script (repeated) into function

4.4 COMPLETING THE PROGRAM

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

# Title: The Silly Sentence Generator 3000
# Author: Your Name
# The game is set in medieval days: a time of stone castles,
# knights with swords, and (some say) mythical beasts that breathe fire.
# Your main character is a young boy named Raspi.1 One day Raspi is
# out gathering firewood and gets lost in the forest. He stumbles upon the
# entrance to a cave. # He peers in the entrance and finds that the cave splits
# into a left tunnel and # a right tunnel. He remembers a folk tale his
# grandmother used to tell of a mysterious cave in this very forest that holds
# enormous treasures. It's said the treasure is guarded by a ferocious fire-
# breathing dragon. Raspi can't resist the temptation to explore the cave;
# although he knows he should turn back, he walks slowly into the dark cavern

# Display the description of the left cave and thier choices
def left_cave():
        # Left cave
        print("You walk into the left cave.")
        print("The cave opens up to a large room with an underground river.")
        print("You notice a small boat on the edge of the river.")
        print("Do you use the boat, swim, or walk along the side of the river?")
        river_choice = input("Enter B for BOAT, S for SWIM, or W for WALK: ").upper()

        return river_choice

# You walk along the edge of the river
def walk():
        print("You walk along the narrow edge of the river.")

        # Wrong choice
        wrong_answer()

def boat():
        print("You step in the boat and start drifting down the river.")

        # You found the treasurer
        correct_answer()

# You jump in the water and start swimming
def swim():
        print("You dive into the water and start swimming down the river.")

        # Wrong choice
        wrong_answer()

# You found the treasurer
def correct_answer():
        print("You seem have making good decisions!")
        print("Suddenly a stalactite falls from the ceiling and open small treasurer box, full of gold!")
        print("Congratulation!!!")

# Display text describing the player's demise and a game over message
def wrong_answer():
        # Wrong input
        print("You seem to have trouble making good decisions!")
        print("Suddenly a stalactite falls from the ceiling and bonks you on the head.")
        print("Game Over!!!")

# Display the description of the right cave and thier choices
def right_cave():
        # Right cave
        print("You walk into the right cave. The cave starts sloping downward.")
        print("The cave opens up to a large room with a hole in the floor.")
        print("You notice a torch in distance.")
        print("Do you use the rope to climb down hole, or walk toward the torch?")
        river_choice = input("Enter T for TORCH, or R for ROPE: ").upper()
        return river_choice

# You walk toward the torch light
def torch():
        print("You walk toward the torch")

        # Wrong input
        wrong_answer()

# You climb down the rope
def hole():
        print("You climb down the rope.")
        print("Entering the Dragon Lairs")
        print("You have a choice; slay the dragon, or get into the room (hiding)")
        action_choice = input("Enter S for SLAY, or R for hiding into the rooms nearby: ").upper()

        return action_choice

# You try to slay the dragon
def slay():
        print("You try to fight the dragon")

        # Wrong answer
        wrong_answer()

# You enter the dark room
def room():
        print("You entering the dark room inside the dragon lairs")

        # You found the treasurer
        correct_answer()


# Display the title and instruction
print("*" * 80)
print("* Raspi's Cave Adventures *")
print("*" * 80)

# 1st Choice: Left or Right Cave?
print("You see the cave split into a left and right tunnel")
print("Do you choose go left or right?")
cave_choice = input("Enter L for left or R for right: ").upper()
if cave_choice == "L" or cave_choice == "LEFT":
        # Left cave
        choice = left_cave()

        if choice == "W" or choice == "WALK":
                # You walk along the edge of the river
                walk()

        elif choice == "B" or choice =="BOAT":
                # You hop in the boat
                boat()

        elif choice == "S" or choice == "SWIM":
                # You jump in the water and start swimming
                swim()
        else:
                # Wrong input
                wrong_answer()

elif cave_choice == "R" or cave_choice == "RIGHT":
        # Right cave
        choice = right_cave()

        if choice == "T" or choice == "TORCH":
                # You walk toward the torch light
                torch()

        elif choice == "R" or choice =="ROPE":
                # You climb down the rope
                choice = hole()

                if choice == "S" or choice == "SLAY":
                    # You try to slay the dragon
                    slay()

                elif choice == "R" or choice == "ROOM":
                    # You enter the dark room
                    room()

                else:
                    # Wrong input
                    wrong_input()

        else:
        # Wrong input
        wrong_answer()
else:
        # Wrong input
        wrong_answer()
```

Listing 1: Complete Raspi's Cave Adventure

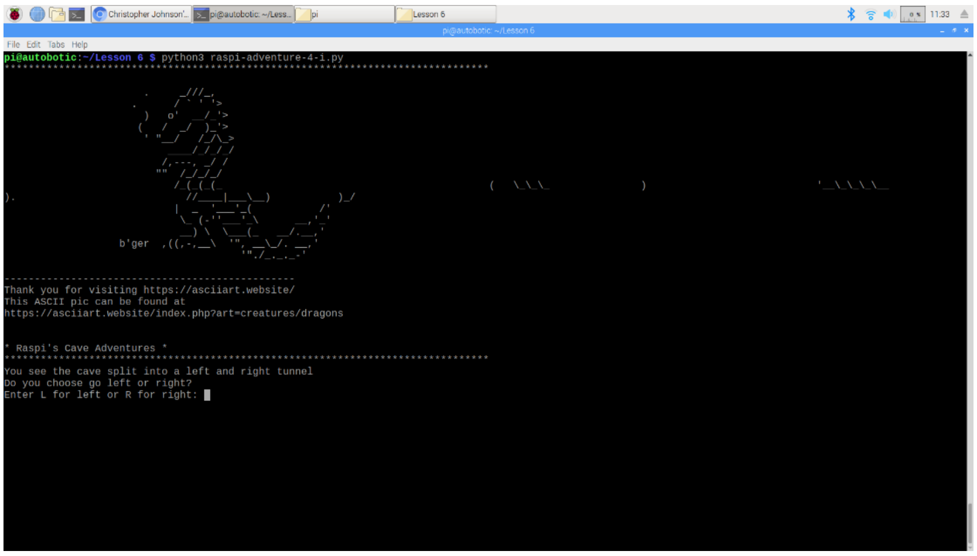
5 EXTRA: STYLING

Before desktop operating systems (OSs) and games had high-end graphics, computers had limited display capabilities. Computer users and programmers invented a new type of art called ASCII art that uses text characters to make images.

To get the ASCII art – there are online website can done it for you. Take a looks:

1. www.chris.com/ascii—A huge collection of ASCII art, sorted by topics
2. http://patorjk.com/software/taag—A text-to-ASCII art generator (TAAG). You type in words, and it automatically creates ASCII art for you.
3. http://picascii.com—A tool that converts pictures to ASCII art

5.1 ASCII ART: RASPI'S CAVE ADVENTURE



```
pi@autobotic:~/Lesson 6 $ python3 raspi-adventure-4-i.py
********************************************************************************

                     .      _///_,
                    .      / ` ' '>
                      )   o'  __/_'>
                     (   /  _/  )_'>
                      ' "__/   /_/\_>
                          ____/_/_/_/
                          /,---, _/ /
                        ""  /_/_/_/
                           /_(_(_(_                                      (   \_\_\_                )                        '__\_\_\_\__
).                           //____|___\__)           )_/
                            |  _  '___'_(           /'
                             \_ (-'' ___'_\      __,'_'
                             __) \  \___(_   __/.__,'
                  b'ger  ,((,-,__\  '", __\_/. __,'
                                       '"./_._._-'

------------------------------------------------
Thank you for visiting https://asciiart.website/
This ASCII pic can be found at
https://asciiart.website/index.php?art=creatures/dragons


* Raspi's Cave Adventures *
********************************************************************************
You see the cave split into a left and right tunnel
Do you choose go left or right?
Enter L for left or R for right:
```

Figure 21: Fashioned Raspi's Cave Adventure