How to Create a Text Based Adventure Game in Python

Practice your Python programming with some simple text processing and decision handling to create a playable game.

BY SHAY LYNN KHAN

UPDATED APR 12, 2023

A Python text-based adventure game is a fun project you can undertake if you are learning how to program. You can make a text adventure game using Python, run it in a command line, and change the story based on the text that the player enters.

The Python script for a Python adventure game will cover several kinds of fundamental programming concepts. This includes print statements, if statements, and functions.

How to Create the Python Script for the Text Adventure Game and Add Story Content

You can make a text-based adventure game in Python by creating a script using a standard text file with a .py extension. If you are not familiar with Python syntax, take a look at some basic Python examples that may help you learn it faster.

Note

You can download the full source code for the Python adventure game from this GitHub repository.

You can also look at other useful Python one-liners to perform certain tasks, before starting on the Python adventure game.

In the main function of the Python file, set up your story and welcome message.

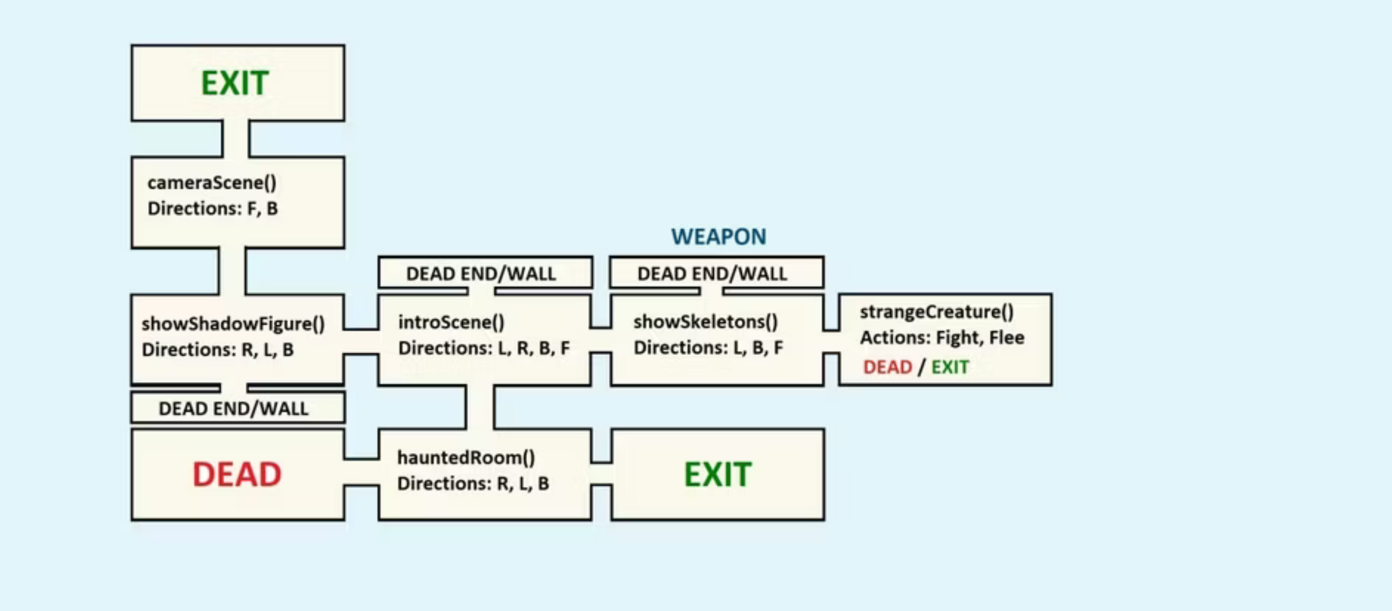
1. Create a new file called "AdventureGame.py", to store the code for your Python adventure game.
2. In the file, add the main starting function. The function will include a brief opening story to welcome the player to the Python text-based adventure game. It will then call another function called **introScene()**.

```
if __name__ == "__main__":
  while True:
    print("Welcome to the Adventure Game!")
    print("As an avid traveler, you have decided to visit the Catacombs of Paris.")
    print("However, during your exploration, you find yourself lost.")
    print("You can choose to walk in multiple directions to find a way out.")
    print("Let's start with your name: ")
    name = input()
    print("Good luck, " +name+ ".")
    introScene()
```

How to Create Multiple Scenes and Options in the Story for the Python Adventure Game

The story for the text-based adventure game will contain several scenes or "rooms". You can create a function for each scene of the Python adventure game so that you can re-use it later if the player ends up entering the same room again.

Each scene will also have different choices of where to go. It is a good idea to map out your story before coding the scenarios, to make sure the story for your text-based adventure game is well organized.



Each scene for the Python text game will have a list of valid directions, and an if-statement for the multiple paths the player can take. Depending on the path the player takes in your Python adventure game, the script will call the next scene.

Create functions for the scenes that will occur in your Python adventure game.

1. Create the **introScene()** function above the main function. Add a message and the directions that the player can walk in.

```
def introScene():
  directions = ["left", "right", "forward"]
  print("You are at a crossroads, and you can choose to go down any of the four hallways. Where would you like to go?")
  userInput = ""
  while userInput not in directions:
    print("Options: left/right/backward/forward")
    userInput = input()
    if userInput == "left":
      showShadowFigure()
    elif userInput == "right":
      showSkeletons()
    elif userInput == "forward":
      hauntedRoom()
    elif userInput == "backward":
      print("You find that this door opens into a wall.")
    else:
      print("Please enter a valid option for the adventure game.")
```

2. Depending on the user's input, your Python adventure game will call another scene. For example, if the player types "left", the adventure game will display the scene **showShadowFigure()** to the player. From this room, if the player goes backward, the Python adventure game will take them back to the intro scene. If they go left or right, they will either enter another room or hit a dead end.

```
def showShadowFigure():
  directions = ["right", "backward"]
  print("You see a dark shadowy figure appear in the distance. You are creeped out. Where would you like to go?")
  userInput = ""
  while userInput not in directions:
    print("Options: right/left/backward")
    userInput = input()
    if userInput == "right":
      cameraScene()
    elif userInput == "left":
      print("You find that this door opens into a wall.")
    elif userInput == "backward":
      introScene()
    else:
      print("Please enter a valid option for the adventure game.")
```

3. Add the camera scene to handle the case where the player turns right. This is where they can find one of the exits. Call the **quit()** function to end the Python text-based adventure game. The player can also still choose to move backward to the previous scene.

```
def cameraScene():
  directions = ["forward", "backward"]
  print("You see a camera that has been dropped on the ground. Someone has been here recently. Where would you like to go?")
  userInput = ""
  while userInput not in directions:
    print("Options: forward/backward")
    userInput = input()
    if userInput == "forward":
      print("You made it! You've found an exit.")
      quit()
    elif userInput == "backward":
      showShadowFigure()
    else:
      print("Please enter a valid option for the adventure game.")
```

4. Back to the beginning of the adventure game, you will still need to add the functions for the remaining scenes. Add the **hauntedRoom()** scene for the case where the player chooses to move forward. This will also end the Python adventure game depending on the player's choice.

```
def hauntedRoom():
  directions = ["right", "left", "backward"]
  print("You hear strange voices. You think you have awoken some of the dead. Where would you like to go?")
  userInput = ""
  while userInput not in directions:
    print("Options: right/left/backward")
    userInput = input()
    if userInput == "right":
      print("Multiple goul-like creatures start emerging as you enter the room. You are killed.")
      quit()
    elif userInput == "left":
      print("You made it! You've found an exit.")
      quit()
    elif userInput == "backward":
      introScene()
    else:
      print("Please enter a valid option for the adventure game.")
```

5. You can also add more interesting content to your Python text-based adventure game. Create a global variable, at the very top of the file, called "weapon". It will either be true or false depending on if the player finds it.

```
weapon = False
```

6. In one of the rooms, set the weapon variable to true if the player finds it. The player can use it in the next room if needed.

```
def showSkeletons():
  directions = ["backward", "forward"]
  global weapon
  print("You see a wall of skeletons as you walk into the room. Someone is watching you. Where would you like to go?")
  userInput = ""
  while userInput not in directions:
    print("Options: left/backward/forward")
    userInput = input()
    if userInput == "left":
      print("You find that this door opens into a wall. You open some of the drywall to discover a knife.")
      weapon = True
    elif userInput == "backward":
      introScene()
    elif userInput == "forward":
      strangeCreature()
    else:
      print("Please enter a valid option for the adventure game.")
```

7. If the player finds the weapon, they can kill the enemy in the next room, and find another exit. Otherwise, the enemy will kill them.

```
def strangeCreature():
  actions = ["fight", "flee"]
  global weapon
  print("A strange goul-like creature has appeared. You can either run or fight it. What would you like to do?")
  userInput = ""
  while userInput not in actions:
    print("Options: flee/fight")
    userInput = input()
    if userInput == "fight":
      if weapon:
        print("You kill the goul with the knife you found earlier. After moving forward, you find one of the exits. Congrats!")
      else:
        print("The goul-like creature has killed you.")
      quit()
    elif userInput == "flee":
      showSkeletons()
    else:
      print("Please enter a valid option for the adventure game.")
```

How to Run the Python Script for the Text-Based Adventure Game

You can run the script for your Python text-based game using a terminal or command prompt. As you enter input into the terminal, the story will continue to move forward to the next scene of the Python adventure game.

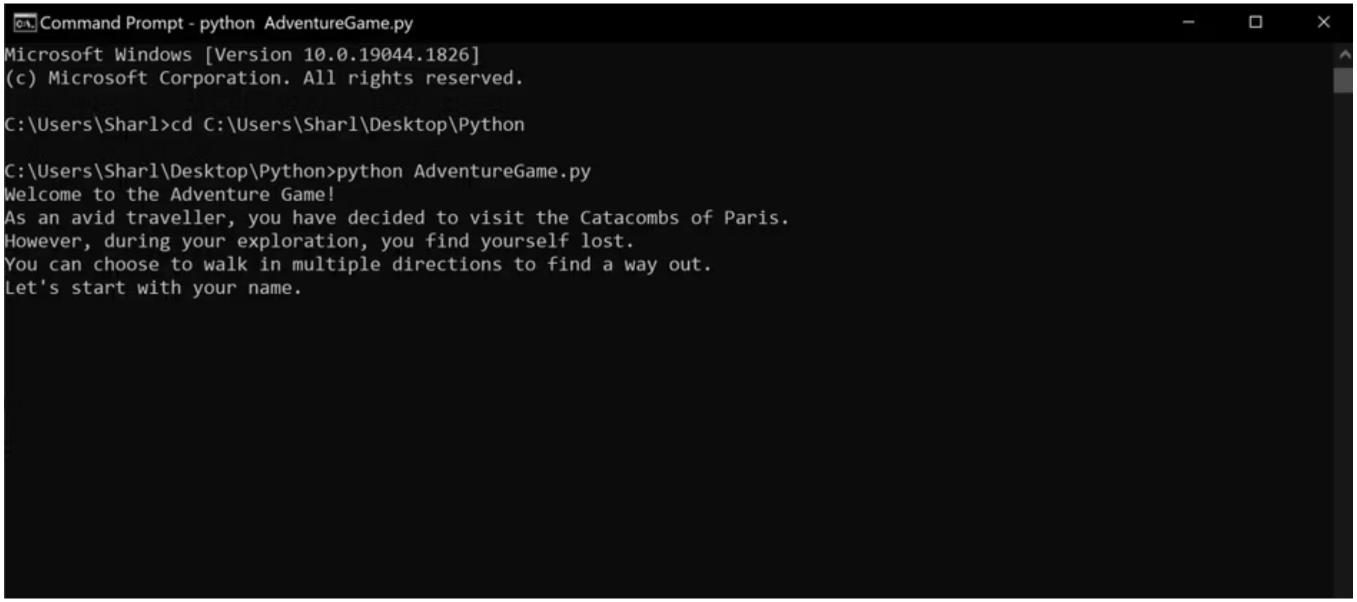
1. Using a terminal or command prompt, navigate to the location where you stored the file for your Python adventure game.

```
cd C:\Users\Sharl\Desktop\Python
```

2. Run the script to start your Python text-based adventure game.

```
python AdventureGame.py
```

3. The opening message will welcome you to start playing the Python adventure game.

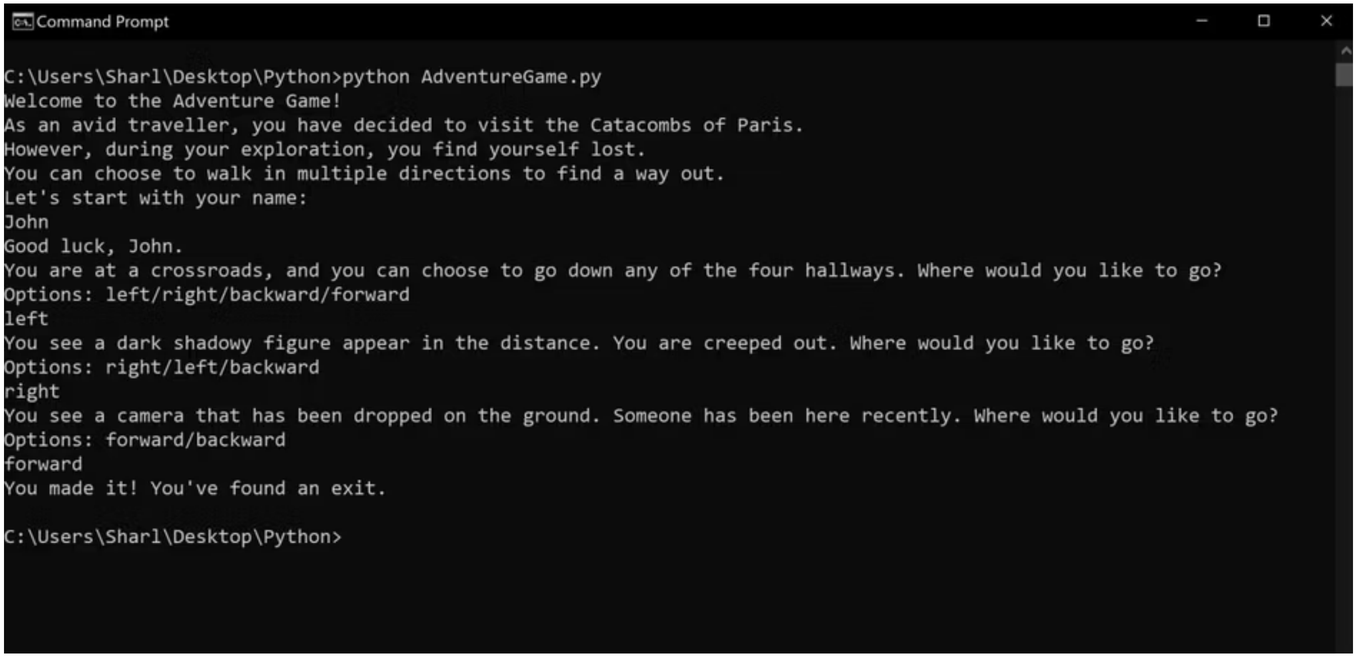


```
Microsoft Windows [Version 10.0.19044.1826]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sharl>cd C:\Users\Sharl\Desktop\Python

C:\Users\Sharl\Desktop\Python>python AdventureGame.py
Welcome to the Adventure Game!
As an avid traveller, you have decided to visit the Catacombs of Paris.
However, during your exploration, you find yourself lost.
You can choose to walk in multiple directions to find a way out.
Let's start with your name.
```

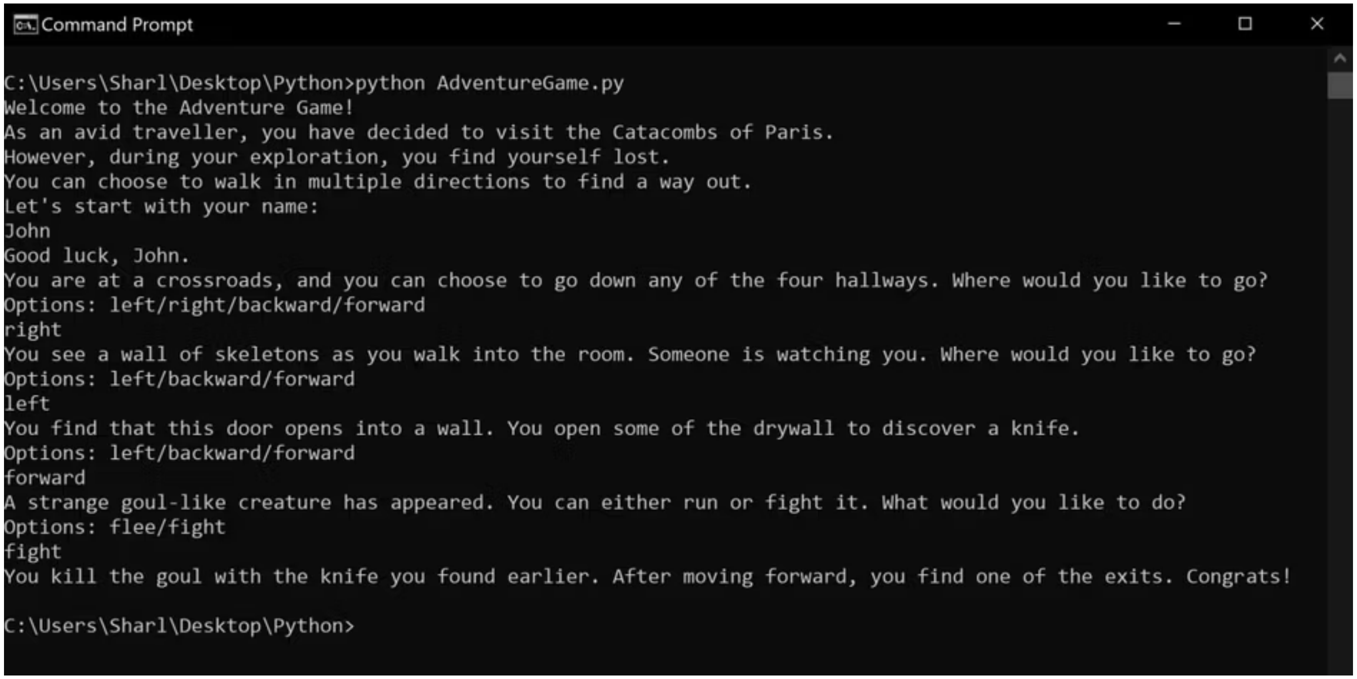
4. Type from the available options listed, such as "left", "right", or "backward". If you enter an invalid input, the Python adventure game will prompt you for a valid one.



```
C:\Users\Sharl\Desktop\Python>python AdventureGame.py
Welcome to the Adventure Game!
As an avid traveller, you have decided to visit the Catacombs of Paris.
However, during your exploration, you find yourself lost.
You can choose to walk in multiple directions to find a way out.
Let's start with your name:
John
Good luck, John.
You are at a crossroads, and you can choose to go down any of the four hallways. Where would you like to go?
Options: left/right/backward/forward
left
You see a dark shadowy figure appear in the distance. You are creeped out. Where would you like to go?
Options: right/left/backward
right
You see a camera that has been dropped on the ground. Someone has been here recently. Where would you like to go?
Options: forward/backward
forward
You made it! You've found an exit.

C:\Users\Sharl\Desktop\Python>
```

5. You can also replay the text-based Python adventure game to choose another path.



```
C:\Users\Sharl\Desktop\Python>python AdventureGame.py
Welcome to the Adventure Game!
As an avid traveller, you have decided to visit the Catacombs of Paris.
However, during your exploration, you find yourself lost.
You can choose to walk in multiple directions to find a way out.
Let's start with your name:
John
Good luck, John.
You are at a crossroads, and you can choose to go down any of the four hallways. Where would you like to go?
Options: left/right/backward/forward
right
You see a wall of skeletons as you walk into the room. Someone is watching you. Where would you like to go?
Options: left/backward/forward
left
You find that this door opens into a wall. You open some of the drywall to discover a knife.
Options: left/backward/forward
forward
A strange goul-like creature has appeared. You can either run or fight it. What would you like to do?
Options: flee/fight
fight
You kill the goul with the knife you found earlier. After moving forward, you find one of the exits. Congrats!

C:\Users\Sharl\Desktop\Python>
```

Create a Simple Text-Based Adventure Game Using Just One Python Script

You can create a Python text-based adventure game using a Python script, and run it on a command line. Inside the Python adventure game, you can present the player with a welcoming message and initial story. The player can then type in their actions based on the options you present within the Python adventure game.

If you want to become a more well-rounded Python developer, there are other projects you can make other than a text-based adventure game. You can have a look at some of the useful tools that you can use or integrate with Python.

Readers like you help support MUO. When you make a purchase using links on our site, we may earn an affiliate commission. ***Read More***.