

JAMMING

## 1 Create a Text Based Adventure Game in Python

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

1  
23

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

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

## Create the Python Script for the Text Adventure Game and Add Story to it

To make a text-based adventure game in Python by creating a script using a standard text file with a .py extension. If you are familiar with Python syntax, take a look at some [basic Python examples that may help you learn it faster](#).

You can download the full source code for the Python adventure game from this [GitHub repository](#).

Before looking at other [useful Python one-liners](#) to perform certain tasks, before starting on the Python adventure game.

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

Create a new file called "AdventureGame.py", to store the code for your Python adventure game.

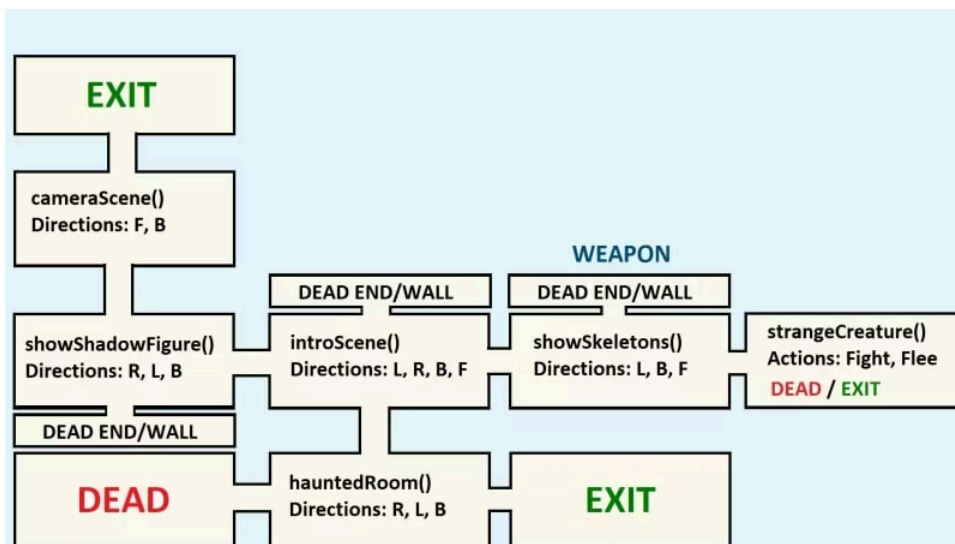
add the main starting function. The function will include a brief opening story to welcome the player to the Python text-based adventure game to clipboard another function called **introScene()**.

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

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

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

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



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

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

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

```
ntroScene():
    actions = ["left", "right", "forward"]
    it("You are at a crossroads, and you can choose to go down any of the four hallways. Where would you like to go")
    Input = ""
    le userInput not in directions:
        rint("Options: left/right/backward/forward")
```

✕

```

userInput = input()
if clipboard_t == "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.")

```

+

+

+

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

+

+

```

showShadowFigure():
    actions = ["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 actions:
        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.")

```

+

amera 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 xt-based adventure game. The player can also still choose to move backward to the previous scene.

+

```

cameraScene():
    actions = ["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 actions:
        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.")

```

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

```

hauntedRoom():
    actions = ["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 actions:
        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()
o clipboard put == "backward":
    introScene()
lse:
    print("Please enter a valid option for the adventure game.")

```



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



```
has_weapon = False
```



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



```

showSkeletons():
    actions = ["backward", "forward"]
    has_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.")
        has_weapon = True
    elif userInput == "backward":
        introScene()
    elif userInput == "forward":
        strangeCreature()
    else:
        print("Please enter a valid option for the adventure game.")

```



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



```

strangeCreature():
    actions = ["fight", "flee"]
    has_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 has_weapon:
            print("You kill the goul with the knife you found earlier. After moving forward, you find one of the exits.")
        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.")

```

## Run the Python Script for the Text-Based Adventure Game

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

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

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

print to start your Python text-based adventure game.

to clipboard

1 AdventureGame.py

ing message will welcome you to start playing the Python adventure game.

and Prompt - python AdventureGame.py

Microsoft Windows [Version 10.0.19044.1826]

Copyright (c) Microsoft Corporation. All rights reserved.

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

C:\Sharl\Desktop\Python>python AdventureGame.py

Welcome to the Adventure Game!

As a traveller, you have decided to visit the Catacombs of Paris.

During your exploration, you find yourself lost.

Choose to walk in multiple directions to find a way out.

Start with your name.

From the available options listed, such as "left", "right", or "backward". If you enter an invalid input, the Python adventure game will prompt you to enter one.

and Prompt

C:\Sharl\Desktop\Python>python AdventureGame.py

Welcome to the Adventure Game!

As a traveller, you have decided to visit the Catacombs of Paris.

During your exploration, you find yourself lost.

Choose to walk in multiple directions to find a way out.

Start with your name:

ok, John.

You are at a crossroads, and you can choose to go down any of the four hallways. Where would you like to go?

left/right/backward/forward

A dark shadowy figure appear in the distance. You are creeped out. Where would you like to go?

right/left/backward

A camera that has been dropped on the ground. Someone has been here recently. Where would you like to go?

forward/backward

Choose it! You've found an exit.

C:\Sharl\Desktop\Python>

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

```
Python Document
Copy to clipboard

C:\Shar1\Desktop\Python>python AdventureGame.py
Welcome to the Adventure Game!
As a traveller, you have decided to visit the Catacombs of Paris.
During your exploration, you find yourself lost.
You choose to walk in multiple directions to find a way out.
Please start with your name:
Name: John.
You are at a crossroads, and you can choose to go down any of the four hallways. Where would you like to go?
Input: left/right/backward/forward
Input: left
You see a wall of skeletons as you walk into the room. Someone is watching you. Where would you like to go?
Input: left/backward/forward
Input: backward
You find that this door opens into a wall. You open some of the drywall to discover a knife.
Input: left/backward/forward
Input: forward
A large goul-like creature has appeared. You can either run or fight it. What would you like to do?
Input: flee/fight
Input: fight
You kill the goul with the knife you found earlier. After moving forward, you find one of the exits. Congrats!
C:\Shar1\Desktop\Python>
```

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

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

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

Help support MUO. When you make a purchase using links on our site, we may earn an affiliate commission. [Read More](#).

### Your Tech IQ With Our Free Newsletters

Email Address

SUBSCRIBE

By subscribing, you agree to our [Privacy Policy](#) and may receive occasional deal communications; you can unsubscribe anytime.

Comments



### RELATED TOPICS

[GAMING](#) [PROGRAMMING](#) [GAME DEVELOPMENT](#) [PYTHON](#)

### THE AUTHOR

Khan • Author

Learning new things through personal projects. Outside coding, Shay also loves gaming and playing the piano.

### RELATED ARTICLES



to clipboard



10 Useful Tools for Python Developers



WORD GAMES

Today's Connections Hints and Answer (Monday, June 10, 2024)



APPLE PENCIL

I Shouldn't Have Bothered Buying an Apple Pencil for My iPad



IPHONE TRICKS

How to Overlay and Combine Two Images on iPhone



ONLINE PRIVACY

This Hidden Network Practice Can Compromise Your Privacy: How to Deal With It



Join Our Team

Our Audience

About Us

Press & Events

Contact Us

Follow Us



Advertising

Careers

Terms

Privacy

Policies

MUO is part of the Valnet Publishing Group