

# The Gauntlet of Riddles

Use what you have learned about Python so far in class to make your own adventure game!



## Overview

Chatbots are used in a lot of modern applications as a way of providing support and help to users without the need for human intervention. This works well for common enquiries around things like sign-up/registration processes, account management and first level technical support. These chatbots simulate conversation by analysing the user's text input to determine which information should be displayed next.

Some types of puzzle computer games rely on the user looking for clues and/or solving logic challenges to progress to the next stage. You can build your own puzzle game using conversations as a way of providing the user with clues, and analysing their responses to check their answers. This task requires you to build your own *Gauntlet of Riddles* - a series of increasingly difficult conversation-based puzzles that challenge the user to think laterally about the clues provided by the "Guardians" of each gate in the Gauntlet.

## Specification

The game begins with a welcome message, where the user provides their name before starting the tutorial level of the Gauntlet. The tutorial level has been written for you, and the provided code gives you a sample code structure to use for the program.

The game has a minimum requirement of implementing the Base Game, with additional downloadable content (DLC) as an extension, only required to achieve full marks.

## Base game

The following table describes the minimum requirements for each level of the Gauntlet in the base game. There are 7 levels in the base game, and the requirements below specify the correct answer, clue and any necessary hints for each of the levels:

Level	Requirements
Welcome <small>(provided)</small>	<p>The user is greeted, then asked to enter the required information for the program.</p> <p>e.g.</p> <pre data-bbox="391 286 1410 405"> Welcome to the Gauntlet of Riddles. Solve each riddle to pass the Guardians. What is your name? Captain featherbot </pre>
Tutorial <small>(provided)</small>	<p>The user needs to provide an empty response to the Guardian's challenge. The clue is given explicitly by the Guardian (i.e. "Be silent to continue"). An obvious hint is provided when the user is incorrect.</p> <p>e.g.</p> <pre data-bbox="391 656 1410 846"> Be silent to continue. &gt; huh? Shh.... &gt; You may pass. </pre>
1	<p>The user needs to provide a specific string to pass. The string is given in the clue. An additional hint is optional.</p> <p>e.g.</p> <pre data-bbox="391 1097 1410 1288"> I'm the Guardian, say my name to pass &gt; The Guardian Say my name to pass &gt; my name You may pass. </pre>
2	<p>The user needs to provide a specific string to pass, made up of two parts. The clue is given in two parts, with each part revealed based on a different user response. Additional hints are optional.</p> <p>e.g.</p> <pre data-bbox="391 1574 1410 1834"> Two digits are needed - first and last. &gt; first a couple... &gt; last this many is a crowd... &gt; 23 You may pass. </pre>
3	<p>The user needs to answer the question in ALL CAPS to pass. The string is given in the clue. A hint should be provided if the user gets the answer wrong.</p> <p>e.g.</p>

	<pre>Shout the answer to pass. &gt; the answer You have to SHOUT the answer. &gt; THE ANSWER You may pass.</pre>
4	<p>The user must answer consecutive questions correctly.  One question in the sequence must require the name they provided at the beginning of the program to pass.  If an answer in the sequence is incorrect, the user starts the sequence again.  Additional hints are optional.</p> <p>e.g.</p> <pre>Knock knock... &gt; who's there you! &gt; jack sparrow Knock knock... &gt; who's there you! &gt; Captain featherbot You may pass.</pre>
5	<p>The user must repeat a specific string multiple times consecutively to pass.  If an answer in the sequence is incorrect, the user starts the sequence again.  Additional hints are optional.</p> <p>e.g.</p> <pre>Say something 3 times in a row. &gt; what Say something 3 times in a row. &gt; something &gt; hi Say something 3 times in a row. &gt; something &gt; something &gt; something You may pass.</pre>
6	<p>The user must answer a sequence of questions correctly.  There must be multiple possible solutions to the first question.  The answer to the second question must use the answer to the first.  Additional hints are optional.</p> <p>e.g.</p> <pre>What ends everyone? &gt; death ...and that links to? &gt; honey You may pass.</pre> <p>* death is the end of life, which happens to everyone. Death ends in the letter h, which starts the word honey.</p> <p>and:</p>

	<pre> What ends everyone? &gt; e ...and that links to? &gt; elephants You may pass. </pre> <p>* e is the last letter of everyone. E starts the word elephants.</p>
7	<p>The user must interpret a binary representation of a character as an answer to the question. There must be multiple possible solutions based on the answer to the first question. Additional hints are optional.</p> <p>e.g.</p> <pre> Are you happy or sad? &gt; happy 0b00011111011000000000 &gt; 😊 You may pass. </pre> <p>and:</p> <pre> Are you happy or sad? &gt; sad 0b00011111011000010100 &gt; 😞 You may pass. </pre> <p>* Using the chr function with the binary value copy + pasted will give the correct emoji response</p>

Additional levels can be added in a “DLC” extension to the game according to the rules below.

## Testing your code

Your tests will be assessed! Run the program and show the input and output that you used to test each level. To do this, it is sufficient to copy the output from your Python REPL and submit it as a text file along with your assignment.

## DLC

If you complete the base game for the Gauntlet, you can earn additional marks by adding DLC content that extends the game’s features. **In addition to extra levels**, you will need to add the following features:

### DLC 1

The game must include some kind of scoring element. Examples include:

- The user gets +1 for a correct answer, -1 for an incorrect answer. Their score is the total at the end of the game;
- The user gets a percentage score based on the number of guesses they needed to complete the Gauntlet, compared to the smallest number of guesses they needed; or
- The user gets a score based on the speed with which they complete the Gauntlet.

*You can come up with your own version of scoring based on the rest of your game - you don’t need to use one of the examples described above.*

## DLC 2

The game must include some kind of command list that the user can run at any time to display the game's status. Example commands you may want to implement are:

- help - shows the list of possible commands, and what they do;
- quit - immediately quits the game;
- status - shows the level the player is on and their current score.

*You can implement any other commands you like, and may want to think about what other information the player may want to see based on your questions.*

The additional marks that you can achieve by completing the DLC content can be found in the rubric.

## Submission

*<Insert instructions for your students to submit their code here. This will vary depending on your school's submission policy. Student may submit their code / program directly, or you could consider using the ACA Assessment module in Grok Learning for the Gauntlet of Riddles to manage task submission>*