(Beta) RPG

{code club}

Introduction:

In this challenge, you'll design and code your own RPG maze game. The aim of the game will be to collect objects and escape from a house, making sure to avoid all the monsters!

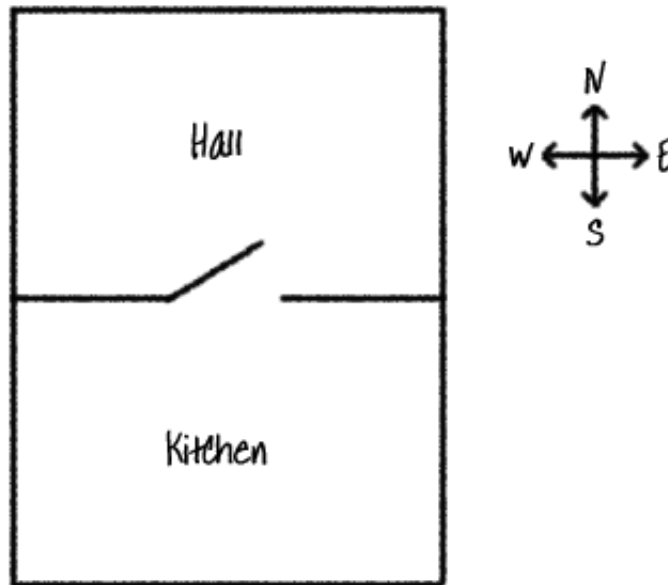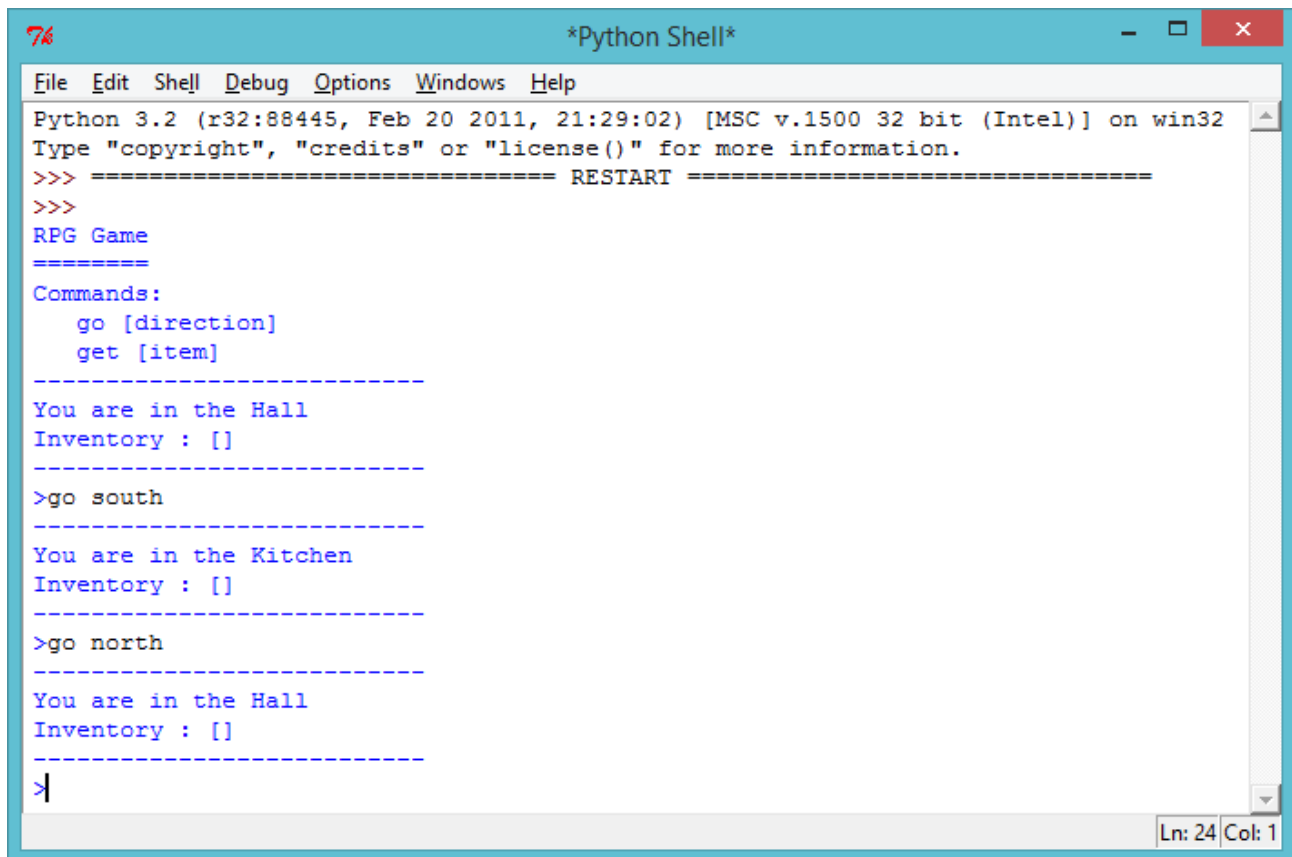|	Activity Checklist	Follow these INSTRUCTIONS one by one
	Test your Project	Click on the green flag to TEST your code
	Save your Project	Make sure to SAVE your work now

Step 1: Adding new rooms

✓ Activity Checklist

1. If you run the RPG.py file, you'll be able to play a very basic
   RPG game that only has 2 rooms. Here's a map of the game:

You can type `go south` to move from the hall to the kitchen, and then `go north` to go back to the hall again!

```
7%                         *Python Shell*                    _  □  ×
File  Edit  Shell  Debug  Options  Windows  Help
Python 3.2 (r32:88445, Feb 20 2011, 21:29:02) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> =============================== RESTART ===============================
>>>
RPG Game
========
Commands:
   go [direction]
   get [item]
--------------------------
You are in the Hall
Inventory : []
--------------------------
>go south
--------------------------
You are in the Kitchen
Inventory : []
--------------------------
>go north
--------------------------
You are in the Hall
Inventory : []
--------------------------
>|
                                                            Ln: 24 Col: 1
```

1. You can also see what happens when you type in a direction that you cannot go. For example, there is no room to the west of the hall, so if you type `go west`, you'll get a friendly error message.

```
7%                         *Python Shell*                    _  □  ×
File  Edit  Shell  Debug  Options  Windows  Help
Python 3.2 (r32:88445, Feb 20 2011, 21:29:02) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> =============================== RESTART ===============================
>>>
RPG Game
========
Commands:
'go [direction]'
'get [item]'
--------------------------
You are in the Hall
Inventory : []
--------------------------
>go west
You can't go that way!
--------------------------
You are in the Hall
Inventory : []
--------------------------
>|
                                                            Ln: 20 Col: 1
```

③

1. If you edit the RPG.py file, you can see that the map is coded as a dictionary of rooms: ☐

```
#a dictionary linking a room to other room positions
rooms = {

        1 : {  "name"  : "Hall" ,
               "south" : 2
             } ,

        2 : {  "name"  : "Kitchen" ,
               "north" : 1
             }

     }
```
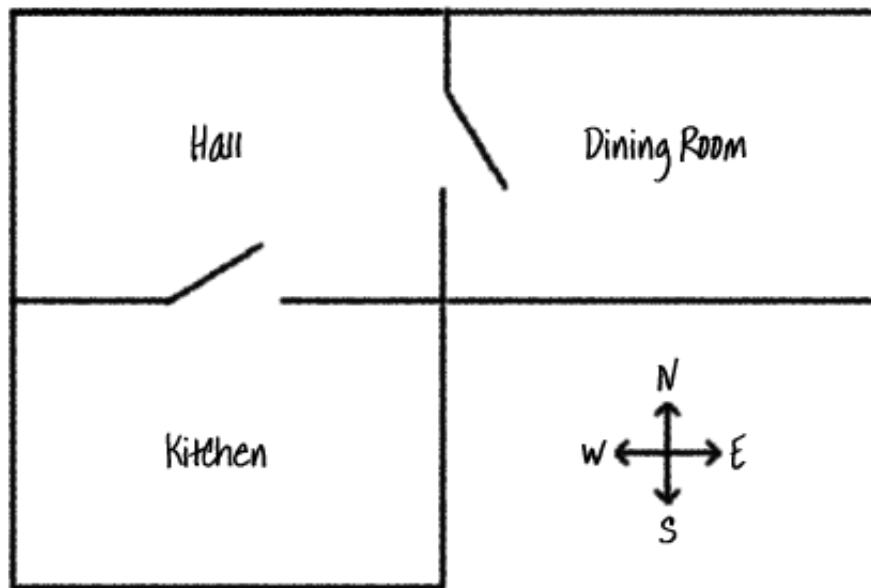
Actually, this is a dictionary that links a room number to another dictionary, containing all of the information about the room. For example, room 1 in the code above is the hall. The hall is linked to room 2 (the kitchen) to the south. Room 2 (the kitchen) also links to room 1 (the hall) to the north.

1. Let's add another room to your map (a dining room), to make it a bit more interesting! ☐

This new dining room is linked to the hall (to the west). Let's add this new room into the code:

```
#a dictionary linking a room to other room positions
rooms = {

        1 : {  "name"  : "Hall" ,
               "south" : 2 ,
               "east"  : 3
            } ,

        2 : {  "name"  : "Kitchen" ,
               "north" : 1
            } ,

        3 : {  "name"  : "Dining Room",
               "west"  : 1
            }

      }
```

To add this room to the game, notice that you need to add the new room (room number 3), and give it a name. You also need to link to room 1 (the hall) to the west of the new room. You need to add information to the dictionary for the hall, to allow you to enter the dining room to the east. Remember to add a comma after the second room in the dictionary, or your code won't run!

1. Try out the game with the new dining room:

```
Python 3.2 (r32:88445, Feb 20 2011, 21:29:02) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> =============================== RESTART ================================
>>>
RPG Game
========
Commands:
'go [direction]'
'get [item]'
--------------------------
You are in the Hall
Inventory : []
--------------------------
>go south
--------------------------
You are in the Kitchen
Inventory : []
--------------------------
>go east
--------------------------
You are in the Dining Room
Inventory : []
--------------------------
>
```

If you can't move in and out of the dining room, just check that you added all of the code above (including the extra commas to the lines above).

Challenge: Add new rooms

Add more rooms to your game. For example, you could create a living room to the south of the dining room. Remember to add a door to/from one of the other rooms!

6

Now that you have lots of rooms, let's leave items in the rooms for the player to collect as they move through the maze.

✓ Activity Checklist

1. Adding an item into a room is easy. You can just add it to the dictionary of a room. For example, let's put a key in the hall.

☐

```
#a dictionary linking a room to other room positions
rooms = {

        1 : {  "name"  : "Hall" ,
               "south" : 2 ,
               "east"  : 3 ,
               "item"  : "key"
             } ,

        2 : {  "name"  : "Kitchen" ,
               "north" : 1
             } ,

        3 : {  "name"  : "Dining Room",
               "west"  : 1
             }

      }
```

Remember to put a comma after the line above the new item, or your program won't run!

1. If you run your game after adding the code above, you can now see a key in the hall, and you can even pick it up (by typing `get key`) and add it to your inventory!

☐

⑦

```
Python 3.2 (r32:88445, Feb 20 2011, 21:29:02) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ============================= RESTART =============================
>>>
RPG Game
========
Commands:
    go [direction]
    get [item]
--------------------------
You are in the Hall
Inventory : []
You see a key
--------------------------
>get key
key got!
--------------------------
You are in the Hall
Inventory : ['key']
--------------------------
>
```

Challenge: Add new items

Add an item to some of the rooms in your game. You can add anything that you think would be helpful in trying to escape the house! For example, a shield or a magic potion.

Step 3: Adding enemies

This game is too easy! Let's add enemies to some rooms that the player must avoid.

✔ Activity Checklist

1. Adding an enemy to a room is as easy as adding any other item. Let's add a hungry monster to the kitchen:

```
#a dictionary linking a room to other room positions
```

8

```
rooms = {

        1 : {  "name"  : "Hall" ,
               "south" : 2 ,
               "east"  : 3 ,
               "item"  : "key"
           } ,

        2 : {  "name"  : "Kitchen" ,
               "north" : 1 ,
               "item"  : "monster"
           } ,

        3 : {  "name"  : "Dining Room",
               "west"  : 1
           }

    }
```
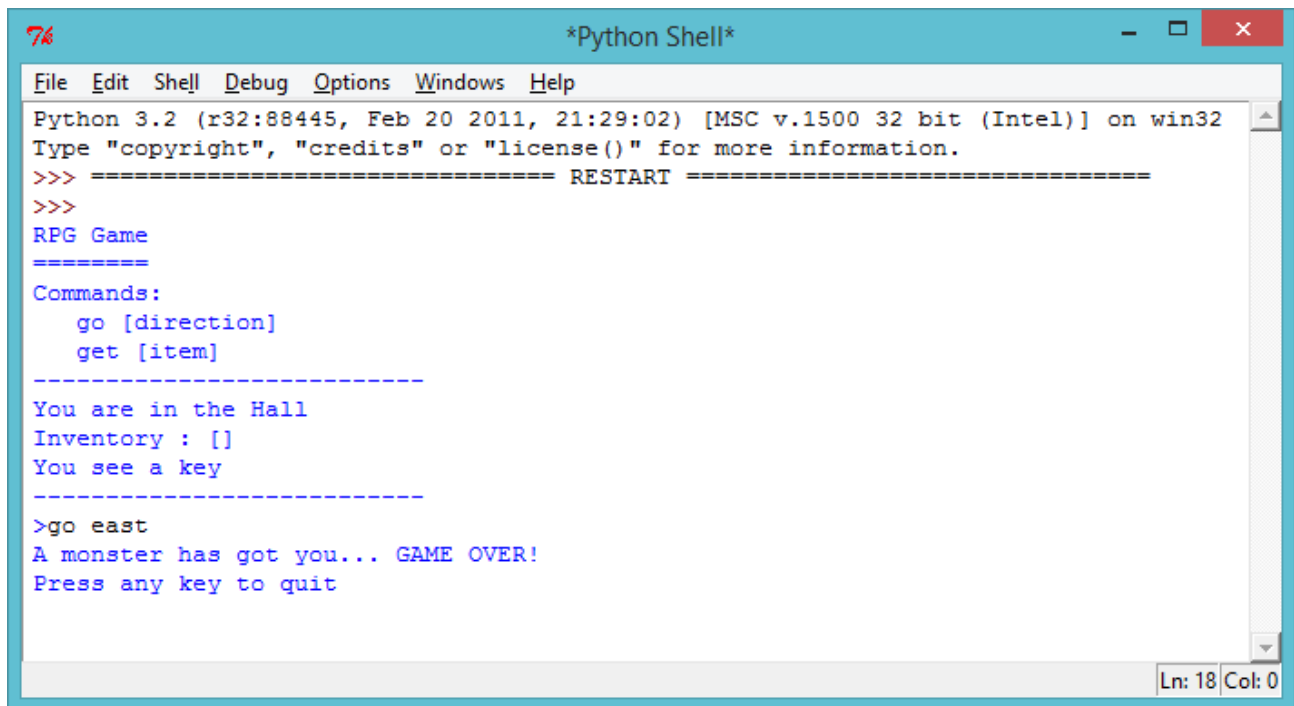
1. You also want to make sure that the game ends if the player
   enters a room with a monster in. You can do this with the
   following code, which you should add to the end of the game.

```
    #player loses if they enter a room with a monster
    if "item" in rooms[currentRoom] and "monster" in
rooms[currentRoom]["item"]:
        print("A monster has got you... GAME OVER!")
        print("Press any key to quit")
        input()
        break
```

This code checks whether there is an item in the room, and if so, whether that
item is a monster. Notice that this code is indented, putting it in line with the
code above it. This means that the game will check for a monster every time
the player moves into a new room.

1. Test out your code by going into the kitchen, which now contains a monster. ☐

```
7⁄4                          *Python Shell*                       _ □ ×
File  Edit  Shell  Debug  Options  Windows  Help
Python 3.2 (r32:88445, Feb 20 2011, 21:29:02) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ================================ RESTART ================================
>>>
RPG Game
========
Commands:
   go [direction]
   get [item]
-------------------------
You are in the Hall
Inventory : []
You see a key
-------------------------
>go east
A monster has got you... GAME OVER!
Press any key to quit
                                                              Ln: 18 Col: 0
```

Challenge: Adding more monsters

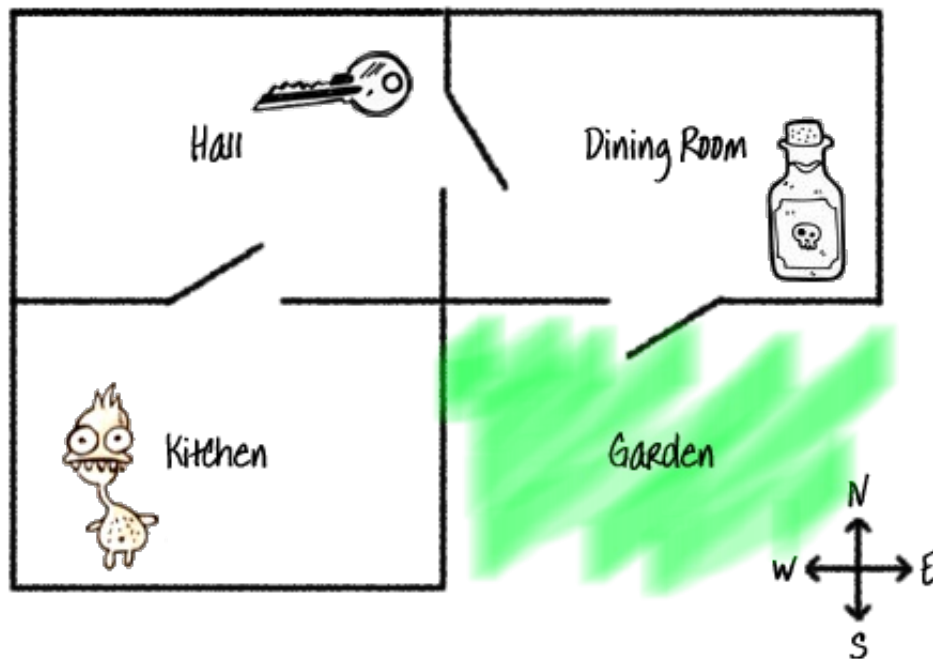Add more monsters to your game, to make it harder to escape the house!

Step 4: Winning the game

Let's give your player a mission, which needs to completed to win the game.

✓ Activity Checklist

1. In this game, the player wins by getting to the garden and escaping the house. They also need to have the key with them, and the magic potion. Here's a map of the game. ☐
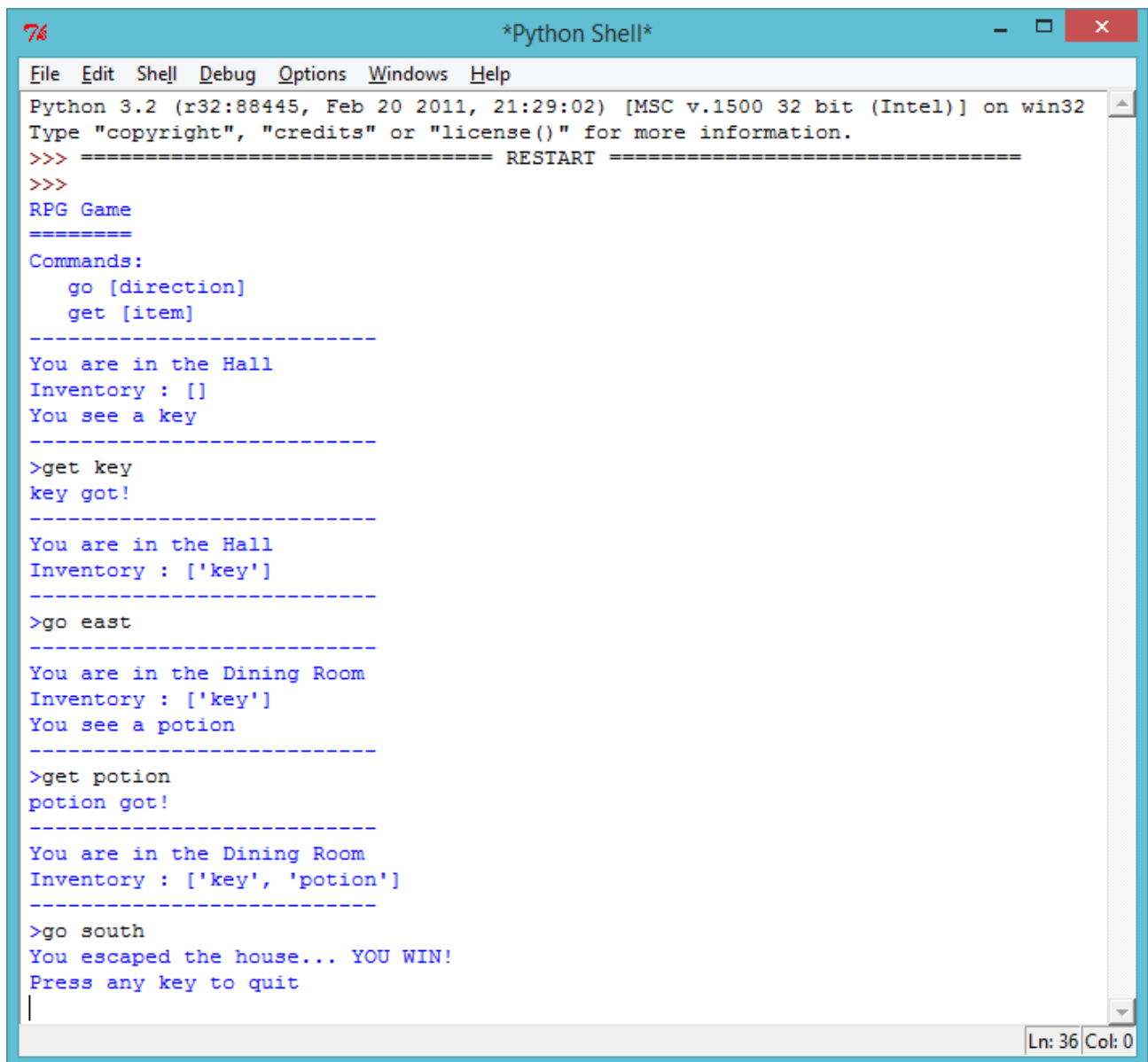
Notice that there's another fourth room (the garden) that links to the dining room (to the north).

1. Use the guidance in step 1 above to add a garden to your game. Remember to add doors, to link to other rooms in the house. ☐

2. Use the guidance in step 2 above to add a potion to the dining room (or another room in your house). ☐

3. To allow the player to win the game when they get to the garden with the key and the potion, add this code to the end of your game: ☐

```python
#player wins if they get to the garden with a key and a shield
if currentRoom == 4 and 'key' in inventory and 'potion' in
inventory:
    print("You escaped the house... YOU WIN!")
    print("Press any key to quit")
    input()
    break
```

Again, make sure this code is indented, and in line with the code above it. This code means that the message `...YOU WIN!` is displayed if the player is in room 4 (the garden) and if the key and the potion are in the inventory. If you have more than 4 rooms, you may have to use a different room number for your garden in the code above.

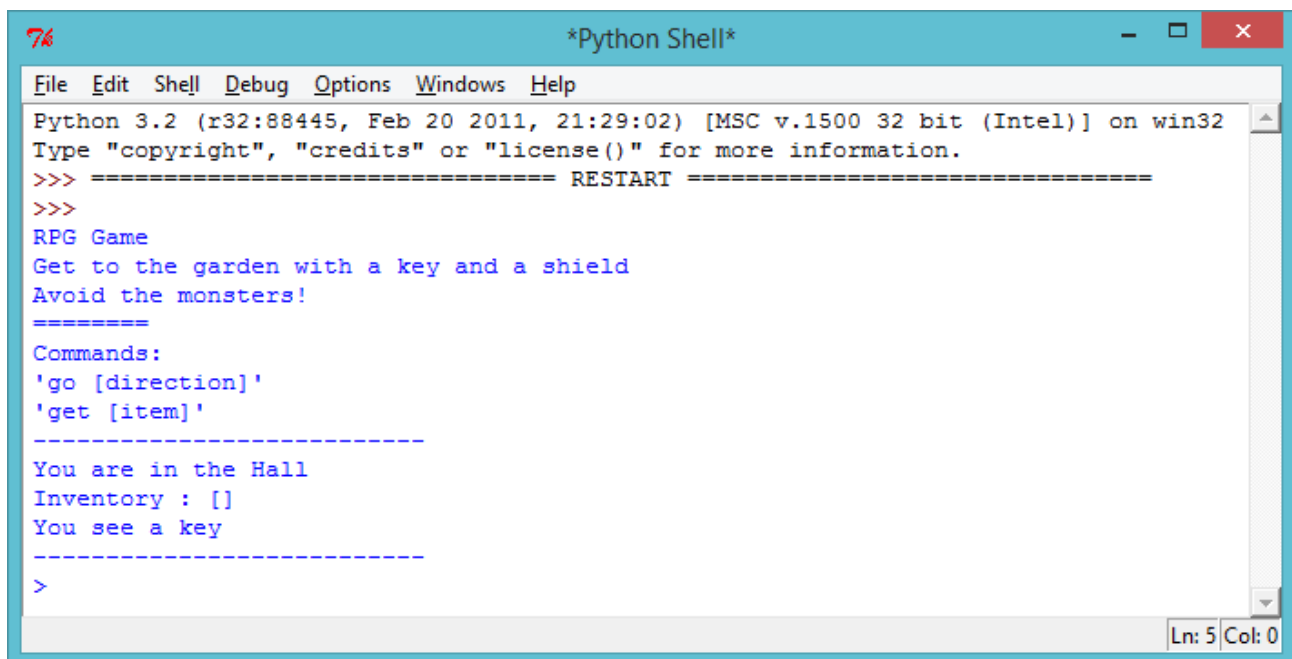1. Test your game to make sure the player can win!

```
7%                          *Python Shell*                    _  □  ×

File  Edit  Shell  Debug  Options  Windows  Help
Python 3.2 (r32:88445, Feb 20 2011, 21:29:02) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ============================= RESTART ===============================
>>>
RPG Game
========
Commands:
   go [direction]
   get [item]
----------------------------
You are in the Hall
Inventory : []
You see a key
----------------------------
>get key
key got!
----------------------------
You are in the Hall
Inventory : ['key']
----------------------------
>go east
----------------------------
You are in the Dining Room
Inventory : ['key']
You see a potion
----------------------------
>get potion
potion got!
----------------------------
You are in the Dining Room
Inventory : ['key', 'potion']
----------------------------
>go south
You escaped the house... YOU WIN!
Press any key to quit
|
                                                              Ln: 36 Col: 0
```

1. Finally, let's add some instructions to your game, so that the player knows what they have to do. Edit the `showInstructions()` function to include more information.

```
def showInstructions():
    #print a main menu, the commands and instructions
    print('''
RPG Game
========
Instructions:
  Get to the garden
  ...
========
Commands:
  go [direction]
  get [item]
''')
```

You will need to add instructions to tell the user what items they need to collect, and what they need to avoid!

```
7% *Python Shell* _ □ ×
File  Edit  Shell  Debug  Options  Windows  Help
Python 3.2 (r32:88445, Feb 20 2011, 21:29:02) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ============================== RESTART ================================
>>>
RPG Game
Get to the garden with a key and a shield
Avoid the monsters!
========
Commands:
'go [direction]'
'get [item]'
--------------------------
You are in the Hall
Inventory : []
You see a key
--------------------------
>
                                                                  Ln: 5 Col: 0
```

Challenge: Develop your own game

Use what you've learnt to create your own game. Add lots of rooms, monsters ot avoid and items to collect. Remember to modify the code so that the player wins when they get to a certain room with some of the objects in their inventory. It may help you to sketch a map before you start coding! You could even add stairs to your map and have more than one level of rooms, by typing `go up` and `go down`.