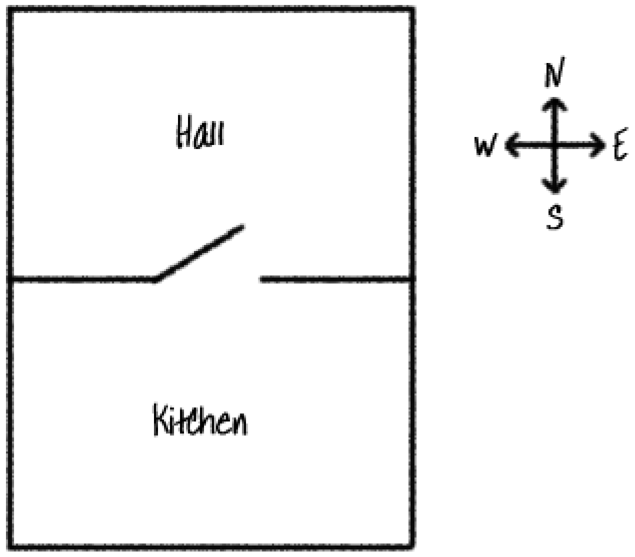
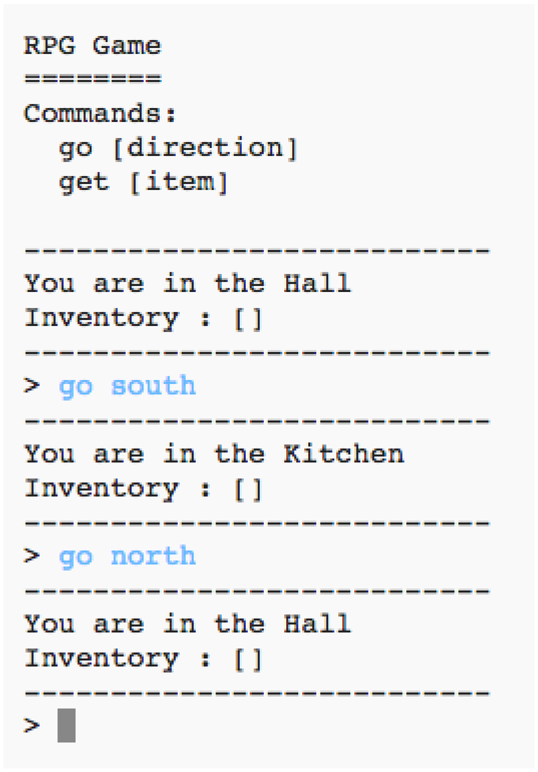
In this project, we will design and code our own RPG maze game. The aim of the game will be to collect objects and escape from a house, making sure to avoid all the monsters!

***

This is a very basic RPG game that only has 2 rooms. Here’s a map of the game:

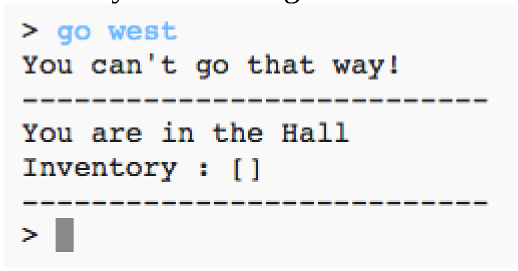


You can type `go south` to move from the hall to the kitchen, and then `go north` to go back to the hall again!



```
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 : []
---------------------------
>
```

What happens when you type in a direction that you cannot go? Type go west in the hall and you'll get a friendly error message.



```
> go west
You can't go that way!
---------------------------
You are in the Hall
Inventory : []
---------------------------
>
```

This is not the best so lets fix that. If you find the rooms variable, you can see that the map is coded as a dictionary of rooms:

--CODE--

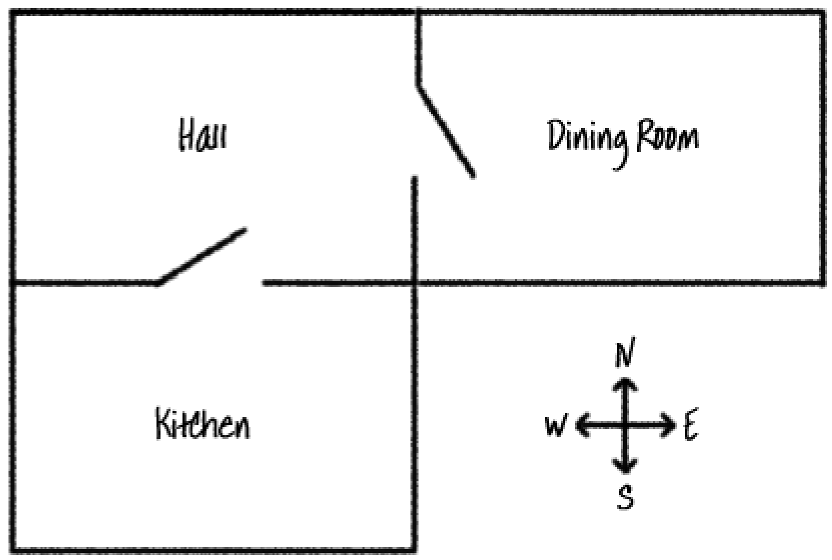
```
rooms = {

        'Hall' : {
          'south' : 'Kitchen'
        },

        'Kitchen' : {
          'north' : 'Hall'
        }

      }
```

--END CODE--

Each room is a dictionary, and rooms are linked together using directions.

***

Let’s add a dining room to our map, to the east of the hall.



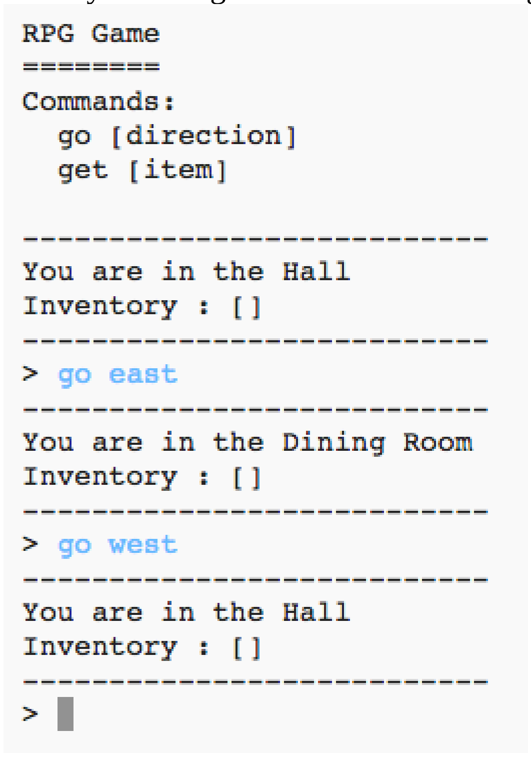
You need to add a 3rd room, called the dining room, and link it to the hall (to the west). You also need to add data to the hall, so that you can move to the dining room to the east. **Don’t forget that you’ll also need to add commas to lines before your new code.**

--CODE--

```
rooms = {

        ‘Hall’ : {
                ‘south’ : ‘Kitchen’,
                ‘east; : ‘Dining Room’
                },

                ‘Kitchen’ : {
                ‘north’ : ‘Hall’
                },

                ‘Dining Room’ : {
                ‘west’ : ‘Hall’
                }
        }
```

--END CODE--

Lets try out our game with the new dining room:



```
RPG Game
========
Commands:
  go [direction]
  get [item]

---------------------------
You are in the Hall
Inventory : []
---------------------------
> go east
---------------------------
You are in the Dining Room
Inventory : []
---------------------------
> go west
---------------------------
You are in the Hall
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--

Can you 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!

--END CHALLENGE--

ADDING ITEMS TO COLLECT

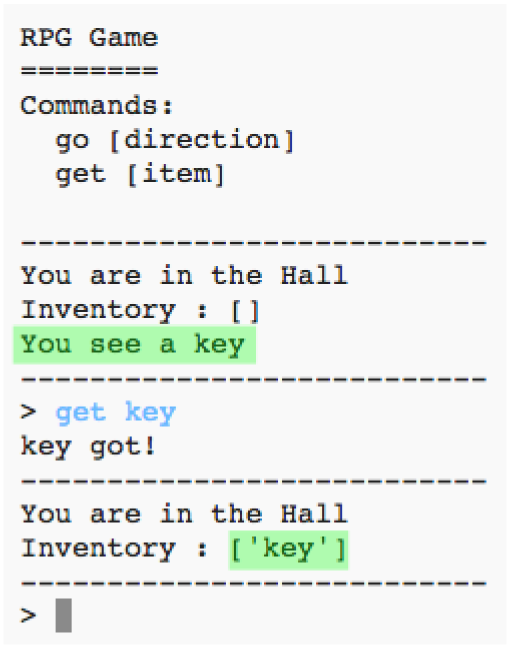
Let's leave items in the rooms for the player to collect as they move through the maze. Adding an item into a room is easy, you can just add it to a room's dictionary. Let's put a key in the hall. Remember to put a comma after the line above the new item, or your program won't run!

--CODE--

```
rooms = {
            'Hall' : {
                        'south' : 'Kitchen',
                        'east' : 'Dining Room',
                        'item' : 'key'
                        },

                        'Kitchen' : {
                                    'north' : 'Hall'
                                    },

                        'Dining Room' : {
                                    'west' : 'Hall'
                                    }
                        }
```

--END CODE--

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) which adds it to your inventory!



```
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.

--END CHALLENGE--

ADDING ENEMIES

This game is too easy! Let’s add enemies to some rooms that the player must avoid. Adding an enemy to a room is as easy as adding any other item. Let’s add a hungry monster to the kitchen:

--CODE--

```
rooms = {
            'Hall' : {
                  'south' : 'Kitchen',
                  'east' : 'Dining Room',
                  'item' : 'key'
                  },

                  'Kitchen' : {
                        'north' : 'Hall',
                        'item' : 'monster'
                        },

                  'Dining Room' : {
                        'west' : 'Hall'
                        }
                  }
```

--END CODE--

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:

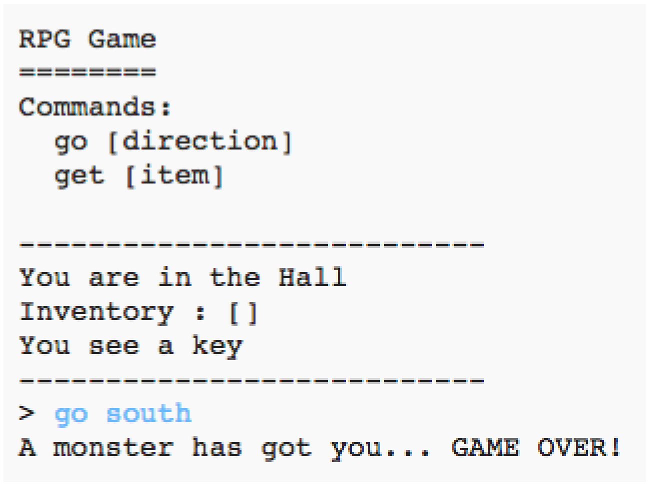
--CODE--

```
else:
      print('Can\'t get' + move[1] + '!')
if 'item' in rooms[currentRoom] and 'monster' in rooms[currentRoom]['item]:
print('A monster has got you... GAME OVER!)
```

```
break
```

--END CODE--

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. Lets test out our code by going into the kitchen, which now contains a monster.



```
RPG Game
========
Commands:
  go [direction]
  get [item]

---------------------------
You are in the Hall
Inventory : []
You see a key
---------------------------
> go south
A monster has got you... GAME OVER!
```

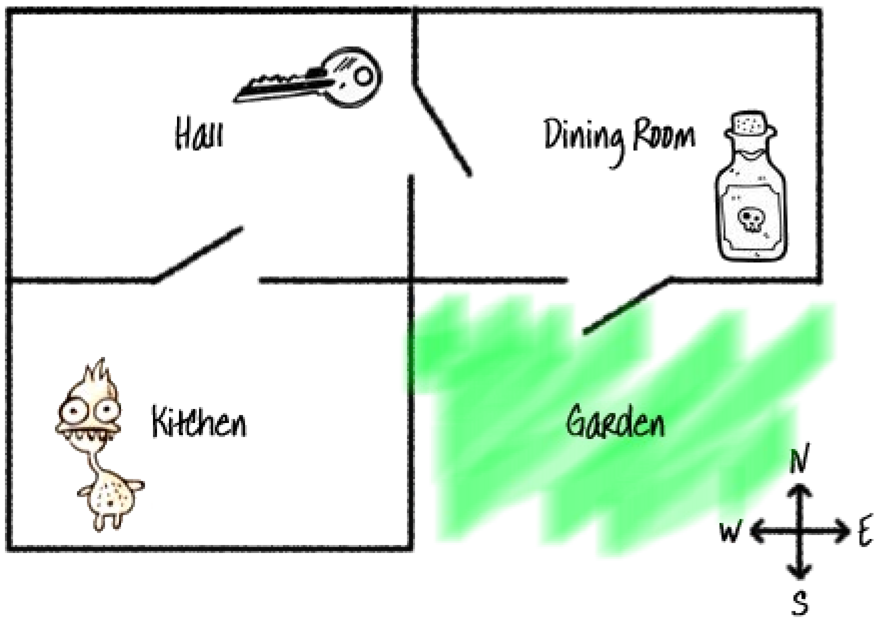
--CHALLENGE--

Challenge: Adding more monsters. Add more monsters to your game, to make it harder to escape the house!

--END CHALLENGE--

WINNING THE GAME

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.



First, you need to add a garden to the south of the dining room. Remember to add doors, to link to other rooms in the house.

--CODE--

```
rooms = {
        ‘Hall’ : {
                ‘south’ : ‘Kitchen’,
                ‘east’ : ‘Dining Room’,
                ‘item’ : ‘key’
                },

        ‘Kitchen’ {
                ‘north’ : ‘Hall’,
                ‘item’ : ‘monster’
                },
	'Dining Room' : {
		'west' : 'Hall',
		'south' : 'Garden'
		},

	'Garden' : {
		'north' : 'Dining Room'
		}
	}
```

--END CODE--

Add a potion to the dining room (or another room in your house).

--CODE--

```
'Dining Room' : {
	'west' : 'Hall',
	'south' : 'Garden'
	'item' : 'potion'
},
```

--END CODE--

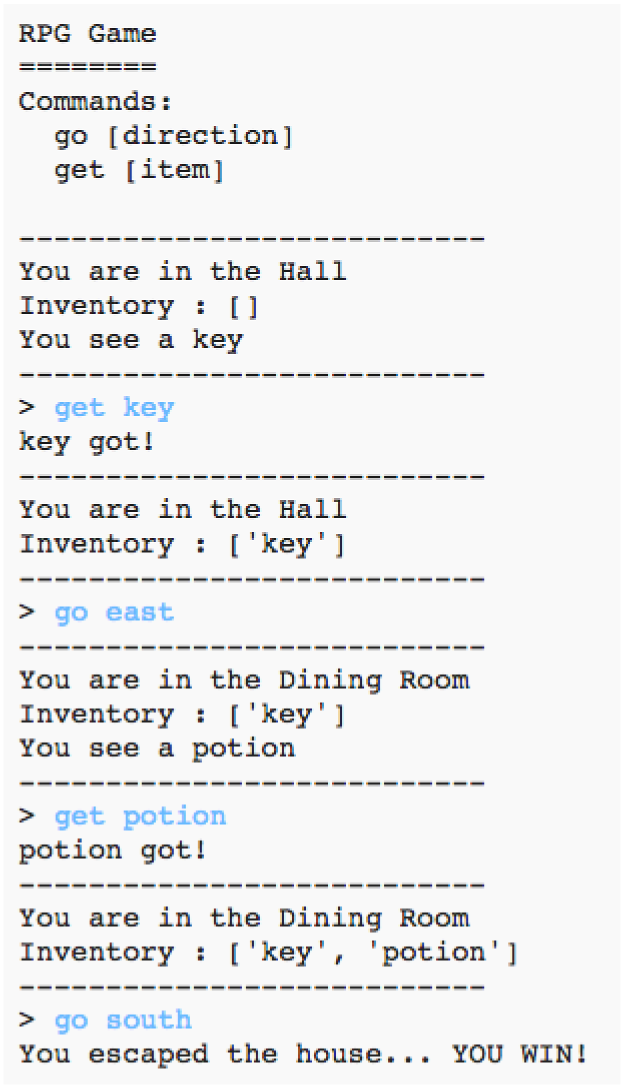
Add this code to allow the player to win the game when they get to the garden with the key and the potion:

--CODE--

```
if 'item' in rooms[currentRoom]  and 'monster' in rooms[currentRoom]['item']:
        print('A monster has got you... GAME OVER!')
        break

    if currentRoom == 'Garden' and 'key' in inventory and 'potion' in inventory:
        print('You have escaped the house... YOU WIN!')
        break
```

--END CODE--

Make sure this code is indented, in line with the code above it. This code means that the message You escaped the house...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. Test your game to make sure the player can win!



```
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!
```

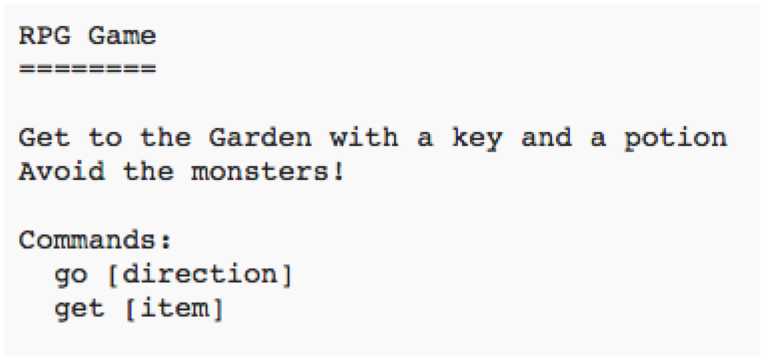
Finally, let's add some instructions to your game, so that the player knows what they have to do. Edit the showInstructions() function to include more information.

--CODE--

```
def showInstructions():
	print('''
RPG Game
========

Get to the Garden with a key and a potion
Avoid the monsters!
Commands:

go [direction]

get [item]

''')
```

--END CODE--

You will need to add instructions to tell the user what items they need to collect, and what they need to avoid! Test your game and you should see your new instructions.



```
RPG Game
========

Get to the Garden with a key and a potion
Avoid the monsters!

Commands:
  go [direction]
  get [item]
```

--CHALLENGE--

Challenge: Make your own game. Use what we have learnt to create your own game. Add lots of rooms, monsters to 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`.

--END CHALLENGE--

I hope you had lots of fun making this project!

Eesa