Python
1

Gameshow

{code club}

Introduction:

In this project you'll learn how to make a guessing game, in which you have to guess which door will reveal a prize.

✅ **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: What's behind the door?

✅ Activity Checklist

1. Imagine a gameshow, where there is an amazing prize behind one of 3 doors. If you pick the correct door, you win the prize! Pick the wrong door, and you get nothing!

   You can play this game by running the following program:

```python
from random import *

#print the 3 doors and the game instructions
print('''
Gameshow!
=========

There's a prize behind one of the 3 doors!
Guess the correct door to win the prize!

  _____  _____  _____
 |     || |   || |     |
 | [1] | | [2] | | [3] |
 |   o | |   o | |   o |
 |_____| |_____| |_____|

Choose a door (1, 2 or 3):
''')

#get the chosen door and store it as an integer (whole
number)
chosenDoor = input()
chosenDoor = int(chosenDoor)

#randomly choose the winning door number (between 1 and
3)
winningDoor = randint(1,3)

#show the player the winning and chosen door numbers
print("The chosen door is", chosenDoor)
```

```python
    print("The winning door is", winningDoor)

    #player wins if the chosen door and winning door number
    are the same
    if chosenDoor == winningDoor:
        print("Well done!")
    else:
        print("Unlucky!")
```

```
7%                        Python Shell                  –  □  ×
File  Edit  Shell  Debug  Options  Windows  Help
Python 3.2.5 (default, May 15 2013, 23:06:03) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ============================== RESTART ==============================
>>>

Gameshow!
=========

There's a prize behind one of the 3 doors!
Guess the correct door to win the prize!
  _____     _____     _____
 |     |   |     |   |     |
 | [1] |   | [2] |   | [3] |
 |  o  |   |  o  |   |  o  |
 |_____|   |_____|   |_____|

Choose a door (1, 2 or 3):

2
The chosen door is 2
The winning door is 1
Unlucky!
>>>
                                                          Ln: 23 Col: 4
```

Here's how the program works: first, a random number between 1 and 3 is chosen, which is the door containing the prize. `randint(1,3)` means 'choose a random integer between 1 and 3'. The program then asks the player for their choice of door, and says "Well done!" if the two numbers are the same or "Unlucky!" if they are different.

A random number is used so that the winning door is different every time. However, Python can't generate random numbers without importing the code to do this, which is why the `random` library is imported at the top of the program.

The lines starting with `#` are comments. These comments are ignored by Python, but are really useful for reminding you what

③

your program does! Try to use comments in your code from now on, to make difficult bits of your programs easier to understand.

2. Using what you already know about loops, you could easily improve this game to allow the player to have 3 guesses, instead of just 1. Run the following program:

```python
from random import *

#print the 3 doors and the game instructions
print('''
Gameshow!
=========

There's a prize behind one of the 3 doors!
Guess the correct door to win the prize!

  _____   _____   _____
 |     | |     | |     |
 | [1] | | [2] | | [3] |
 |   o | |   o | |   o |
 |_____| |_____| |_____|
''')

#allow the player 3 attempts
for attempt in range(3):

    print("\nChoose a door (1, 2 or 3):")

    #get the chosen door and store it as an integer
(whole number)
    chosenDoor = input()
    chosenDoor = int(chosenDoor)

    #randomly choose the winning door number (between 1
and 3)
    winningDoor = randint(1,3)

    #show the player the winning and chosen door
numbers
```

```python
        print("The chosen door is", chosenDoor)
        print("The winning door is", winningDoor)


        #player wins if the chosen door and winning door
    number are the same
        if chosenDoor == winningDoor:
            print("Well done!")
        else:
            print("Unlucky!")
```

```
7%                              Python Shell                    _  □   ×
File  Edit  Shell  Debug  Options  Windows  Help
Python 3.2.5 (default, May 15 2013, 23:06:03) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ============================== RESTART ==============================
>>>

Gameshow!
=========

There's a prize behind one of the 3 doors!
Guess the correct door to win the prize!

  _____   _____   _____
 |     | |     | |     |
 | [1] | | [2] | | [3] |
 |  o  | |  o  | |  o  |
 |_____| |_____| |_____|


Choose a door (1, 2 or 3):
2
The chosen door is 2
The winning door is 2
You WIN!

Choose a door (1, 2 or 3):
2
The chosen door is 2
The winning door is 3
Unlucky!

Choose a door (1, 2 or 3):
1
The chosen door is 1
The winning door is 2
Unlucky!
>>>
                                                           Ln: 35 Col: 4
```

In this program, notice that the code for printing the 3 doors is outside of the loop, as they only need to be printed once. The code for generating random numbers and asking the player to pick a door is inside the loop, as these will each need to be done 3 times. If the code for choosing a random winning door was outside of the loop, then the winning door would be the same door for each of the 3 guesses.

Also, `\n` in the line `print("\nChoose a door (1, 2 or 3):")` prints a new blank line to the screen. This is done to split up the printed text, to make it easier to read.

Save Your Project

Challenge: Keeping score

Create a variable to keep track of how many times the player guesses the correct door. If you need help, remember that this will work in a similar way to the `score` variable in your quiz program! Remember to show the player their final score, at the end of the program (outside of the loop).

Save Your Project

Step 2: `while` loops

✅ Activity Checklist

1. Instead of making your program loop a set number of times, it might be more interesting to let the player choose how long they want to play for. To do this, you'll need to use a different loop, called a `while` loop. Try out this program:

```python
from random import *

#the user changes this variable to end the game
playing = True

score = 0
```

```
#print the 3 doors and the game instructions
print('''
Gameshow!
=========


There's a prize behind one of the 3 doors!
Guess the correct door to win the prize!

  _____   _____   _____
 |     | |     | |     |
 | [1] | | [2] | | [3] |
 |   o | |   o | |   o |
 |_____| |_____| |_____|
''')


#repeat as long as the 'playing' variable is set to
'True'
while playing == True:

    print("\nChoose a door (1, 2 or 3):")

    #get the chosen door and store it as an integer
(whole number)
    chosenDoor = input()
    chosenDoor = int(chosenDoor)

    #randomly choose the winning door number (between 1
and 3)
    winningDoor = randint(1,3)

    #show the player the winning and chosen door
numbers
    print("The chosen door is", chosenDoor)
    print("The winning door is", winningDoor)

    #player wins if the chosen door and winning door
number are the same
    if chosenDoor == winningDoor:
        print("Well done!")
        score = score + 1
    else:
        print("Unlucky!")
```

```
    print("Your score is now", score)

    #ask the player if they want to keep playing
    print("\nDo you want to play again? (y/n)")
    answer = input()
    #end the game if the player types 'n'
    if answer == 'n':
        playing = False

print("Thanks for playing.")
print("Your final score is", score)
```

```
76                           Python Shell                    –  □  ×
File  Edit  Shell  Debug  Options  Windows  Help
Python 3.2.5 (default, May 15 2013, 23:06:03) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ============================ RESTART ================================
>>>

Gameshow!
=========

There's a prize behind one of the 3 doors!
Guess the correct door to win the prize!

  _____   _____   _____
 |     | |     | |     |
 | [1] | | [2] | | [3] |
 |  o  | |  o  | |  o  |
 |_____| |_____| |_____|


Choose a door (1, 2 or 3):
3
The chosen door is 3
The winning door is 2
Unlucky!
Your score is now 0

Do you want to play again? (y/n)
y

Choose a door (1, 2 or 3):
2
The chosen door is 2
The winning door is 2
Well done!
Your score is now 1

Do you want to play again? (y/n)
n
Thanks for playing.
Your final score is 1
>>>
                                                          Ln: 39 Col: 4
```

A `while` loop allows your program to repeat until something happens to stop it. In this program, you want to keep playing the game as long as the variable `playing` is set to `True`. If the player decides that they don't want to play again and

8

inputs `n`, the variable `playing` is set to `False`, and the loop stops running. The name for data that is either `True` or `False` is Boolean data.

2. Get someone to test out your game, to make sure that it runs until they input `n` to end the game.

Challenge: Fixing the input

☐ What happens if the player tries to quit by typing `N` instead of `n`? Can you use the `lower()` function to fix this problem?

☐ What happens if they type `no` instead of just `n`? Can you fix your program, so that the program ends if the user's `answer == 'n' or answer == 'no'`?

Challenge: Losing the game

Can you modify your game, so that the player's score is set to 0 whenever they choose the wrong door? Does this change make the game more fun? Or does it make the game too hard?

Step 3: How lucky are you?

✅ Activity Checklist

1. Instead of looping the game until the player decides to quit, you could instead loop the game until the player scores 3 points. The aim of the game could then be to score 3 points in as few attempts as possible:

```python
from random import *

#this variable stores the number of times the game is
played
attempts = 0

score = 0

#print the 3 doors and the game instructions
print('''
Gameshow!
=========

There's a prize behind one of the 3 doors!
Guess the correct door 3 times in as few attempts as
possible!

  _____   _____   _____
 |     | |     | |     |
 | [1] | | [2] | | [3] |
 |   o | |   o | |   o |
 |_____| |_____| |_____|
''')

#repeat as long as the score is less than 3
while score < 3:

    #add 1 to the number of attempts
    attempts = attempts + 1

    print("\nAttempt", attempts, ": Choose a door (1, 2
or 3):")

    #get the chosen door and store it as an integer
(whole number)
    chosenDoor = input()
    chosenDoor = int(chosenDoor)

    #randomly choose the winning door number (between 1
and 3)
    winningDoor = randint(1,3)
```

```
    #show the player the winning and chosen door
numbers
    print("The chosen door is", chosenDoor)
    print("The winning door is", winningDoor)

    #player wins if the chosen door and winning door
number are the same
    if chosenDoor == winningDoor:
        print("Well done!")
        score = score + 1
    else:
        print("Unlucky!")

    print("Your score is now", score)

print("\n** You did it! That look you", attempts,
"attempts **")
```

The `while` loop in this program continues to run as long as the score is less than 3 (`while score < 3:`). Once the score gets to 3, the program ends, and the number of attempts is printed.
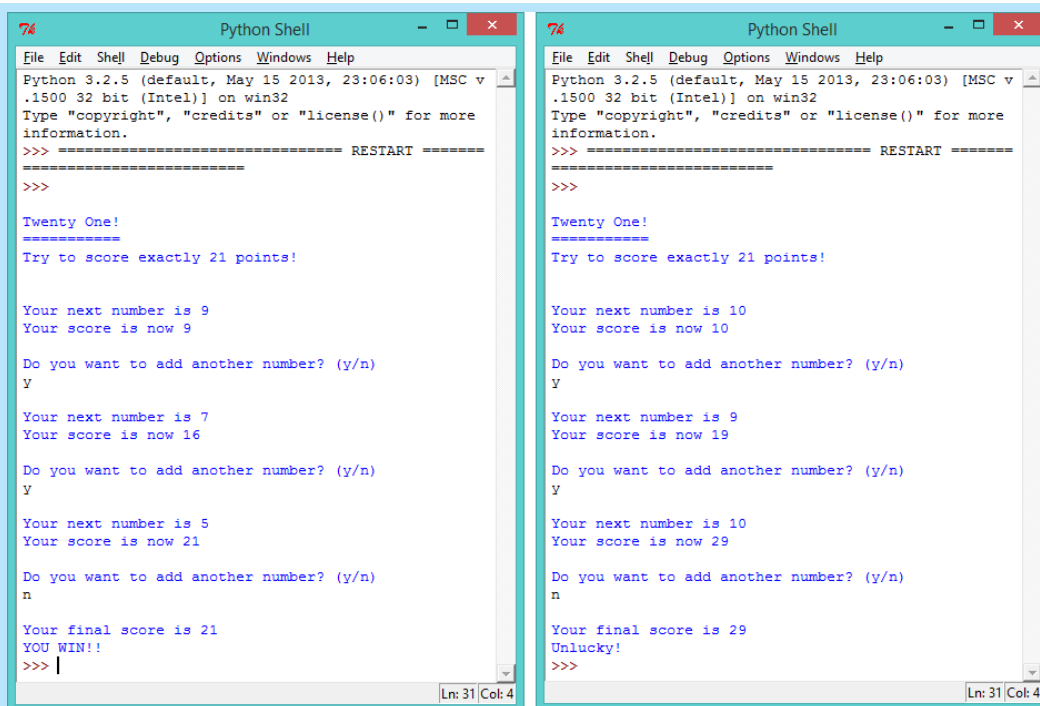
▲ Save Your Project

Challenge: Twenty-one

Can you create a game where the aim is to score exactly 21 points? The program should pick a random number between 1 and 10, which is added to the player's score. The player can then choose to stick with the score they have, or choose to add another random number to their score. The player wins the game if they manage to get to exactly 21 points. Remember to add comments to your program!

Here is an example of the game:

11

```
7⁄₄                Python Shell          _ ▢ ✕
File  Edit  Shell  Debug  Options  Windows  Help
Python 3.2.5 (default, May 15 2013, 23:06:03) [MSC v ▲
.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more
information.
>>> =============================== RESTART =======
=========================
>>>

Twenty One!
===========
Try to score exactly 21 points!


Your next number is 9
Your score is now 9

Do you want to add another number? (y/n)
y

Your next number is 7
Your score is now 16

Do you want to add another number? (y/n)
y

Your next number is 5
Your score is now 21

Do you want to add another number? (y/n)
n

Your final score is 21
YOU WIN!!
>>> |
                                        Ln: 31 Col: 4
```

```
7⁄₄                Python Shell          _ ▢ ✕
File  Edit  Shell  Debug  Options  Windows  Help
Python 3.2.5 (default, May 15 2013, 23:06:03) [MSC v ▲
.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more
information.
>>> =============================== RESTART =======
=========================
>>>

Twenty One!
===========
Try to score exactly 21 points!


Your next number is 10
Your score is now 10

Do you want to add another number? (y/n)
y

Your next number is 9
Your score is now 19

Do you want to add another number? (y/n)
y

Your next number is 10
Your score is now 29

Do you want to add another number? (y/n)
n

Your final score is 29
Unlucky!
>>>
                                        Ln: 31 Col: 4
```

You can use (or change) parts of your gameshow program to
make this new game. Here are some hints that you can use to
help you:

☐ You can use a `while` loop to run the game as long as
the player wants to keep playing:

```
while playing == True:
```

☐ If the user enters `n` because they wish to stick with the
score they have, you can set the `playing` variable to
`False`:

```
if answer == 'n':
    playing = False
```

☐ At the end of the program (outside of the `while` loop),
you can say "Well done!" to the player if their score is
exactly 21:

```
if score == 21:
    print("Well done!")
```