Game show

INTRODUCTION

Imagine a game show where there are 3 doors. You have to guess which door hides the fabulous prize!

We can also make the game so you have 3 guesses, or you keep playing until you get 3 right guesses.

```
digilocaladmin@digilocal-HP-255-G7-Notebook-PC:~$
3 Door Game Show!
=================

Guess which door
the prize is behind
 _______   _______   _______
I       I I       I I       I
I  [1]  I I  [2]  I I  [3]  I
I     oI  I I     oI  I     oI
I      I  I I      I  I      I
I______I  I I______I  I______I
Enter your choice
>1
You are a winner!
digilocaladmin@digilocal-HP-255-G7-Notebook-PC:~$
```

Step One: Getting started

- Open VS Code from the icon in your side bar.

- We need a new file to begin editing. Start a new file and save it as ‘game-show.py’.

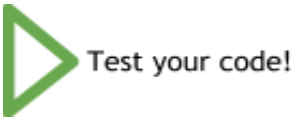
- The core of our game is the randomness of chance. So we begin by importing the python library ‘random’.

```
game-show.py ●
home > digilocaladmin > Barton-Hill > John > game-show.py > {} random
1   import random
```

- Next we’ll add our basic instructions.

```
1   import random
2   instructions = '''3 Door Game Show!
3   =================
4
5   Guess which door
6   the prize is behind
7
8   _________  _________  _________
9   I       I  I       I  I       I
10  I  [1]  I  I  [2]  I  I  [3]  I
11  I      oI  I      oI  I      oI
12  I       I  I       I  I       I
13  I_______I  I_______I  I_______I'''
14
15  print(instructions)
```

- The print statement at the end is so we can test it all looks ok.



Game show

```
digilocaladmin@digilocal-HP-255-G7-Notebook-PC:~$
3 Door Game Show!
=================

Guess which door
the prize is behind

 _______  _______  _______
I       I I       I I       I
I  [1]  I I  [2]  I I  [3]  I
I     oI  I I     oI  I     oI
I      I  I I      I  I      I
I______I  I I______I  I______I
digilocaladmin@digilocal-HP-255-G7-Notebook-PC:~$
```

- Next we need to decide which is the correct door, behind which is the fabulous prize. Add this line below your print statement.

```
14  print(instructions)
15  correct_door = random.randint(1,3)
```

- randint will pick a random integer (whole number) between the numbers given in the brackets, between 1 and 3 for this example.

- Now we need the player's guess.

```
14  print(instructions)
15  correct_door = random.randint(1,3)
16  player_choice = int(input('Enter your choice\n>'))
```

- Our input statement here is a little different. When we store data from input it is always saved as a string, even if it's a number that's been typed. We need to convert it to an integer so we can compare it to the correct_door value, that is what int() does.

- Remember that \n adds a new line to the print statement. The > is there so you know where to type in your choice.

Finally we can compare our input with our random selection and see if the player is a winner.

```
14  print(instructions)
15  correct_door = random.randint(1,3)
16  player_choice = int(input('Enter your choice\n>'))
17  if player_choice == correct_door:
18      print('You are a winner!')
19  else:
20      print('Unlucky, you lose')
```

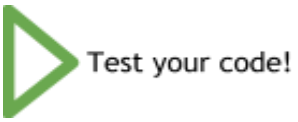
Test your code!

```
digilocaladmin@digilocal-HP-255-G7-Notebook-PC:~$
3 Door Game Show!
=================

Guess which door
the prize is behind

 _______   _______   _______
I       I I       I I       I
I  [1]  I I  [2]  I I  [3]  I
I     oI  I I     oI  I     oI
I      I  I I      I  I      I
I______I  I I______I  I______I
Enter your choice
>1
You are a winner!
digilocaladmin@digilocal-HP-255-G7-Notebook-PC:~$
```

Great, it worked!

But we have to assume it worked. We might be on a losing streak, or something might be wrong with our code. Let's add two lines at line 17 & 18 to confirm our choices.

```
14  print(instructions)
15  correct_door = random.randint(1,3)
16  player_choice = int(input('Enter your choice\n>'))
17  print('The correct door was number',correct_door)
18  print('You chose door number',player_choice)
19  if player_choice == correct_door:
20      print('You are a winner!')
21  else:
22      print('Unlucky, you lose')
```




```
digilocaladmin@digilocal-HP-255-G7-Notebook-PC:~$
3 Door Game Show!
=================

Guess which door
the prize is behind

 _______   _______   _______
I       I I       I I       I
I  [1]  I I  [2]  I I  [3]  I
I     oI  I I     oI  I     oI
I       I I       I I       I
I_____I   I_____I   I_____I
Enter your choice
>1
The correct door was number 2
You chose door number 1
Unlucky, you lose
digilocaladmin@digilocal-HP-255-G7-Notebook-PC:~$
```

Step Two: Best of three!

Let's give the player 3 goes and see how many correct guesses they make.

First, let's change our instructions a little to add a new line \n before each turn.

```
import random
instructions = '''\n3 Door Game Show!
================

Guess which door
the prize is behind

 _______  _______  _______
I       I  I       I  I       I
I  [1] I  I  [2] I  I  [3] I
I     oI  I     oI  I     oI
I       I  I       I  I       I
I_____I  I_____I  I_____I'''
```

We also want to save their score, the number of correct guesses made.

```
score = 0
```

Now we need a loop to give the player 3 attempts.

```
for attempt in range(3):
print(instructions)
```

- Because we added this new loop in the middle of our existing code. The next line isn't indented and we'll get an error unless we fix it. All the following code is part of the loop, so it all needs to be indented.

```
14  for attempt in range(3):
15      print(instructions)
16      correct_door = random.randint(1,3)
17      player_choice = int(input('Enter your choice\n>'))
18      print('The correct door was number',correct_door)
19      print('You chose door number',player_choice)
20      if player_choice == correct_door:
21          print('You are a winner!')
22      else:
23          print('Unlucky, you lose')
```

- If we select it all, so it's highlighted in blue. We can then press the <TAB> key once and it'll all move across neatly.

- Next we need to change the score when the player guesses correctly.

```
20      if player_choice == correct_door:
21          print('You are a winner!')
22          score += 1
```

- The python command += is a short way of writing 'score = score + 1'

- Finally, let's print out how many correct guesses were made.

```
23      else:
24          print('Unlucky, you lose')
25  print('You made', score, 'correct guesses.')
```

Test your code!
A few times.

Step Three: Are you feeling lucky?

Let's make a couple of small changes so that we can play forever, but we'll keep track of how lucky we are.

First we need another two variables. One to keep track of how many tries we've had, and one to see if we're still playing.

```
14  tries = 0
15  playing = True
```

Now we replace our for loop, with a while loop.

```
15  playing = True
16  while playing:
17      print(instructions)
```

After we work out if the guess was correct, let's add one to the number of tries taken.

```
25      else:
26          print('Unlucky, you lose')
27      tries += 1
```

Now we can ask if they'd like to play again.

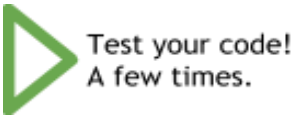
```
28      answer = input('Would you like to play again (y/n)\n>')
29      if answer == 'n':
30          playing = False
```

Test your code!
A few times.

```
Guess which door
the prize is behind

 _______   _______   _______
I       I I       I I       I
I  [1]  I I  [2]  I I  [3]  I
I     oI  I I     oI  I I     oI
I       I I       I I       I
I_____ I  I_____ I  I_____ I
Enter your choice
>1
The correct door was number 3
You chose door number 1
Unlucky, you lose
Would you like to play again (y/n)
>n
You made 0 correct guesses.
digilocaladmin@digilocal-HP-255-G7-Notebook-PC:~$
```

As we know how many tries they took, and how many they got right, we can work out their success record as a percentage.

```
31   lucky = (score / tries) * 100
32   print('You made', score, 'correct guesses.')
33   print('You were correct', lucky,'% of the time')
```




```
Would you like to play again (y/n)
>n
You made 1 correct guesses.
You were correct 25.0 % of the time
digilocaladmin@digilocal-HP-255-G7-Notebook-PC:~$
```

Our decision to play again only works if we type in ‘n’. What happens if we type in ‘N’, or ‘no’, or ‘No’?

- We can modify our input so that we convert everything to .lower case (so that N and n are the same). We can then take just the first letter (so that n, no, never, not are all treated the same).

```
28      answer = input('Would you like to play again (y/n)\n>').lower()[0]
```

- Remember that python starts counting at 0. So the first letter is the one at [0], not the one at [1].

Challenge time!

Can you change your code so the game runs 'while score < 3' so you see how long it takes to get 3 correct answers?

Can you change it so that you lose points if you guess wrong? Does that make it too hard?

Can you add a fourth door?