# DS201 HW1.5 - Monty Hall problem - python simulation

Write a Python code to simulate the chance of winning in the Monty Hall Problem

(explanation will be in the lecture)

Plan to approach to the problem, here the steps:

1. Create a “random number” to represent the randomness of each game
2. Create a list variable to represent each game scenario. Which are three doors, and which door having a price(car).
3. Create a function to count the “Chance to win”
4. Create a function to count the “Chance to win, if stay”
5. Create a function to count the “Chance to win, if switch”
6. Make a loop to iterate the game
7. increase the iteration up to 10,000 times, to simulate the Probability of chance to win if switch or stay

   note: you can have a different approach than this. The goal is to simulate the Experimental Probability getting closer to the Theoretical Probability

1)

Create a 2D list variable to store the game scenario. Simulate number of game =10

```
import random

n= 10

doors =[]

for i in range(n):

        ran = random.choice("012")

        #print(ran)


        if ran == "0":

           doors.append(["X","O","O"])

       (other conditions)
```

```
del doors[0]
print("len(doors):",len(doors))
print(doors)
```

assume that you always choose the first door, write python code to simulate the Chance to win.

2) create a function chance_to_win(list1):

3) chance_to_win_if_stay(list1):

4) create a function chance_to_win_if_switch(list1):

```
In [6]: chance_to_win(doors)
```

```
Total: 10
Sum0: 2 chance: 0.2
Sum1: 3 chance: 0.3
Sum2: 5 chance: 0.5
```

```
In [7]: chance_to_win_if_stay(doors)
```

```
game_id 0 which_door: 1 Win? 0
game_id 1 which_door: 2 Win? 0
game_id 2 which_door: 2 Win? 0
game_id 3 which_door: 1 Win? 0
game_id 4 which_door: 0 Win? 1
game_id 5 which_door: 0 Win? 1
game_id 6 which_door: 2 Win? 0
game_id 7 which_door: 2 Win? 0
game_id 8 which_door: 2 Win? 0
game_id 9 which_door: 1 Win? 0
Total: 10
Chance to win if stay - Sum0: 2 chance: 0.2
Sum1: 3 chance: 0.3
Sum2: 5 chance: 0.5
```

```
In [8]: chance_to_win_if_switch(doors)
```

```
game_id: 0 which_door: 1 Win? 1
game_id: 1 which_door: 2 Win? 1
game_id: 2 which_door: 2 Win? 1
game_id: 3 which_door: 1 Win? 1
game_id: 4 which_door: 0 Win? 0
game_id: 5 which_door: 0 Win? 0
game_id: 6 which_door: 2 Win? 1
game_id: 7 which_door: 2 Win? 1
game_id: 8 which_door: 2 Win? 1
game_id: 9 which_door: 1 Win? 1
Total: 10
Chance to win if stay - Sum0: 2 chance: 0.2
Chance to win if switch (Sum1+Sum2): 8 chance: 0.8
```

5) Modify the number of game to 1,000 and to 10,000.
   Your probability of the random number should be more evenly distributed.
6) Submit as DS201_HW1-5_MontyHallSimulation_LastFirst.ipynb on Canvas