Python - Day 4 Udemy

Archive	☐
Projects	🎴 100 Days of Code: The Complete Python Pro Bootcamp for 2023
URL	https://www.udemy.com/course/100-days-of-code/

Creating a Random numbers using Random Module

```
import random

random_integer = random.randint(10, 200) #10 and 200 are inclusive
print(random_integer)                    #Randint function will help to choose the numbers
```

we can even create the new module for us , let's create a new file name as my_module.py and save the below code and modify the original code a bit

```
# my_module code , i will define the pi

pi = 3.14
```

```
import random
import my_module

random_integer = random.randint(10, 200)
print(random_integer)
print(my_module.pi)

o/p:
131
3.14
```

you can see we have imported the pi value defined in the another program with help of module

random.random( ) → generate the float numbers between 0.0 to 0.9

if you want between 0 to 10 , just multiply the random value by 10

Sample challenge:

You are going to write a virtual coin toss program. It will randomly tell the user "Heads" or "Tails".

```
import random
HeadorTail = random.randint(0,1)
if (HeadorTail == 1):
    {
        print("Heads")
    }
else:
    {
        print('Tails')
    }
```

```
output:
Tails
```

▼ Lists

set of square brackets with many items stored inside, these items can be any data type or maybe they mixed together. example fruits = [item1 , item 2]

```
fruits = ["apple", "orange", "banana", "pine apple"]
print(fruits[0])
print(fruits) # without modified
fruits[1] = "ooorange" # modifying the orange value
print(fruits)
fruits.append("Guava") #adding value to the last
print(fruits)
print(fruits[-1]) #check from the last
```

```
apple
['apple', 'orange', 'banana', 'pine apple']
['apple', 'ooorange', 'banana', 'pine apple']
['apple', 'ooorange', 'banana', 'pine apple', 'Guava']
Guava
```

Sample Challenge:

You are going to write a program that will select a random name from a list of names. The person selected will have to pay for everybody's food bill.

Important: You are not allowed to use the `choice()` function.

Line 8 splits the string `names_string` into individual names and puts them inside a **List** called `names`. For this to work, you must enter all the names as names followed by comma then space. e.g. name, name, name

```
# Import the random module here
import random
# Split string method
names_string = input("Give me everybody's names, separated by a comma. ")
names = names_string.split(", ") #spliting strings storing as lists
# 🚨 Don't change the code above 👆

#Write your code below this line 👇
length =len(names) #finding how many input we have given
random_number = random.randint(0,(length-1)) #generating random numbers depends on input
random_name = names[random_number] #choosing anyoneof them
print(f"{random_name} is going to buy the meal today!")
```

```
output:
Give me everybody's names, separated by a comma.
siva, jaya, subash #input
subash is going to buy the meal today!
```

Nested Lists:

```
fruits = ["apple", "orange", "banana", "pine apple"]
vegetables = ["tomato", "potato", "carrot"]

fruAndVeg = [fruits,vegetables]
print(fruAndVeg)
#for the nested list length will be 2 , if you want to access any particular
item [1][0] access it like this
```

```
Output:
[['apple', 'orange', 'banana', 'pine apple'], ['tomato', 'potato', 'carrot']]
#it contains two different lists inside a list
```

Exercise - Treasure Map

```
Instructions
You are going to write a program that will mark a spot with an X.
In the starting code, you will find a variable called map.
This map contains a nested list. When map is printed this is what the nested list looks like:
[['□', '□', '□'],['□', '□', '□'],['□', '□', '□']]
This is a bit hard to work with. So on lines 6 and 23, we've used this line of code print(f"{row1}\n{row2}\n{row3}" to format the 3 lists to be printed as a 3 by 3 square,
each on a new line.
['□', '□', '□']
['□', '□', '□']
['□', '□', '□']

Your job is to write a program that allows you to mark a square on the map using a two-digit system.
The first digit in the input will specify the column (the position on the horizontal axis).
The second digit in the input will specify the row number (the position on the vertical axis).
```

```
Solution:
# 🚨 Don't change the code below 👇
row1 = ["□","□","□"]
row2 = ["□","□","□"]
row3 = ["□","□","□"]
map = [row1, row2, row3]
print(f"{row1}\n{row2}\n{row3}")
position = input("Where do you want to put the treasure? ")
# 🚨 Don't change the code above 👆

#Write your code below this row 👇
row = int(position[0])
column = int(position[1])

map[column-1][row-1] = "X" //updadting the value on nested list, when we give input we don't provide 0 th row , so we need to -1 to get exact column/row to update

#Write your code above this row 👆
# 🚨 Don't change the code below 👇
print(f"{row1}\n{row2}\n{row3}")
```

```
output:
['□', '□', '□']
['□', '□', '□']
['□', '□', '□']
Where do you want to put the treasure? 31
['□', '□', 'X']
['□', '□', '□']
['□', '□', '□']
➜
```

Final Project : ROCK, PAPER, SCISSORS

```
Instructions
Make a rock, paper, scissors game.

Inside the main.py file, you'll find the ASCII art for the hand signals already saved to a corresponding variable: rock, paper, and scissors. This will make it easy to
print them out to the console.

Start the game by asking the player:

"What do you choose? Type 0 for Rock, 1 for Paper or 2 for Scissors."

From there you will need to figure out:

How you will store the user's input.
How you will generate a random choice for the computer.
How you will compare the user's and the computer's choice to determine the winner (or a draw).
And also how you will give feedback to the player.
You can find the "official" rules of the game on the World Rock Paper Scissors Association website.
```

```
import random

rock = '''
    _______
---'   ____)
      (_____)
      (_____)
      (____)
---.__(___)
'''

paper = '''
    _______
---'   ____)____
          ______)
          _______)
         _______)
---.__________)
'''

scissors = '''
    _______
---'   ____)____
          ______)
       __________)
      (____)
---.__(___)
'''

game_images = [rock, paper, scissors]

user_choice = int(input("What do you choose? Type 0 for Rock, 1 for Paper or 2 for Scissors.\n"))
print(game_images[user_choice])

computer_choice = random.randint(0, 2)
print("Computer chose:")
print(game_images[computer_choice])

if user_choice >= 3 or user_choice < 0:
  print("You typed an invalid number, you lose!")
elif user_choice == 0 and computer_choice == 2:
  print("You win!")
elif computer_choice == 0 and user_choice == 2:
  print("You lose")
elif computer_choice > user_choice:
  print("You lose")
```

```
output:
What do you choose? Type 0 for Rock, 1 for Paper or 2 for
Scissors.
0

    _______
---'   ____)
      (_____)
      (_____)
      (____)
---.__(___)

Computer chose:

    _______
---'   ____)
      (_____)
      (_____)
      (____)
---.__(___)

It's a draw
□
```

```
elif user_choice > computer_choice:
  print("You win!")
elif computer_choice == user_choice:
  print("It's a draw")
```
