How to Unlock the Power of Randomization to Create Intriguing Scripts in Python

By Doug Purcell

Randomization is a phenomena that effects all humans regardless of what walk of life they're from. Many things are randomly assigned to us such as such as our date of birth, nationally, family, height, and eye color. With its inherit nonpartisan attributes, randomization is heavily used in statistics, clinical trials, and shuffling cards. Randomization is a fascinating somewhat overlooked portion of life, so let's get more familiar with it by writing some cool python programs.

Project: A Game of Dice. Humans vs. Randomization

Randomization is a phenomena that effects all humans regardless of what walk of life they're from. Many things are randomly assigned to us such as such as our date of birth, nationally, family, height, and eye color. With its inherit nonpartisan attributes, randomization is heavily used in statistics, clinical trials, and shuffling cards. Randomization is a fascinating somewhat overlooked portion of life, so let's get more familiar with it by writing some cool python programs.

How the Gameplay Works?

The user will start with a bank account of $1000 and can keep making wagers on which number the dice will roll on. If the dice falls on the number that the user guesses, then the user gets the money that they wagered, and vice-versa. The game play will continue until the user runs out of money or exits from the game. This is a truly simple game that displays the beauty and the volatile nature of randomness.

Let’s play a couple of rounds!

```
$ python a_game_of_dice.py
```

Script Hints

- Who goes first? The first question is how to determine which player goes first, the human or the computer? One way is to simulate a coin flip of heads or tails. Both players are allowed to pick which side of the coin they want. If both players both pick the correct side then this process should be repeated until there's a single winner.
- How much money will each player start with? What happens when some edge cases arise like a player betting more money than they have, or enters an invalid input like a string?
- What's the minimum and maximum amount of rounds allowed?
- What happens when the player has no more money to bet?

How Many Rounds we Want the Player to Play?

```
from random import randint

def game():

        print('Welcome! You\'re Playing:')

        human_bank = 1000

        while True:

        you = float(input("How much money do you want to wager? "

                        "Your bank has ${} ".format(human_bank)))
```

How to Check if the Player Has Enough Money in the Bank?

```
        if you > 0 <= human_bank:
                print("You're betting ${}".format(you))
                dice_roll = randint(1, 6)
                human_guess = float(input('What number you think the dice will roll
on? '
                                          'Select numbers: 1-6 '))
```

How do we update the player's bank account if they won or loss a wager?

```
if human_guess == dice_roll:

                        human_bank += you

                        print('Dice landed on {}'.format(dice_roll))

                        print("Congrats! You now have ${}".format(human_bank))

                        else:

                        human_bank -= you

                        print("Bummers! Dice landed on {}! Money gone...".format(dice_roll))

                        print("You have ${}".format(human_bank))

                        if human_bank == 0:

                        print("Game over! You're out of cash")

                        break
```

How do we give a player the option to continue or quit gameplay?

```
play_again = input("Enter 'y' to play again or 'n' to stop" ).lower()
                    if play_again == 'y':
                    continue
                    elif play_again == 'n':
                    print("Game Over...")
                    if human_bank > 1000:
                              print("You're lucky! You won ${} ".format(human_bank - 1000))
                       break
                    elif human_bank < 1000:
                              print("Better luck next time! You loss ${} ".format(1000 -
human_bank))
                              break
                    else:
                              print("You didn't win nor you didn't lose!")
                              break
          else:
                    print('Enter a valid amount. You have ${} to bet'.format(human_bank))
```

View the full source code of a Game of Dice

https://github.com/purcellconsult/Code-Cool-Stuff-With-Python/blob/master/sourcecode/ch 03/a game of dice.py

Project: Random Person Generator

Have you ever seen those online name generators? I have, I've used them a couple of times for novels I was writing. Yes, I wrote novels under a pseudonym and occasionally had a difficult time creating names for fictitious characters! It may seem random (no pun) for a software engineer to do that, but there's actually quite a bit of parallels between writing code and writing stories. Anyhoo, to prevent from deviating let's take the functionality of a simple random name generator a step or two further. Let's write a script that allows us to randomly generate first names, last names, full names, emails, ages, telephone numbers, and email passwords. It will be a fun project to code and showcase!

Script Hints

`first_name`: A function that allows us to generate male or female first names. By default we'll let the program decide on the gendered pronoun.

`last_name`: A function that allows us to generate surnames.

`full_name`: A function that allows us to generate first and last names. Like the first_name function it can be gendered or we can let the program decide this.

`age`: Generate a random age for the person. We can specify something like 1-100.

`phone_number`: Generate a random 10 digit phone number. For the sake of simplicity we can decide on the region which in this example is North American phone numbers. The key here is that the first digit can't be 0 or 1.

`email_password`: We can generate a random email address which uses the random person's first and last names.

How to randomly generate a name? Use the `random` module

```
>>> from random import choice

>>> female_names = ['Molly', 'Sue', 'Angela']

>>> choice(female_names)

...

'Angela'
```

How to randomly generate an age?

```
>>> from random import randint

>>> [randint(1, 100) for x in range(10)]

...

[2, 23, 99, 19, 96, 27, 61, 88, 53, 38]
```

How to generate a random password? Use the `string` module

```
>> from random import choice

>>> from string import ascii_letters

>>> from string import digits

>>> from string import punctuation

>>> ascii_letters

...

'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ'
```

`string` module continued

```
>>> digits

...

'0123456789'

>>> punctuation

'!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'

>>> [ascii for ascii in ascii_letters]

['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v',
'w', 'x', 'y', 'z', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R',
'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z']
```

Random Person Generator Solution

https://github.com/purcellconsult/Code-Cool-Stuff-With-Python/blob/master/sourcecode/ch_03/random_person_generator.py

Project: The California State Lottery

Most states in the US allow denizens the ability to play the lottery. While I'm not advocating for folks to become chronic gamblers, I am advocating simulating the winning numbers through a python program. In this project we're going to simulate the various lotteries that California makes available.

The Type of Lotteries Available

- Daily 3: Pick any 3 numbers within the range of 0-9.
- Daily 4: Pick any 4 numbers within the range of 0-9.
- Fantasy 5: Pick any 5 numbers within the range of 1-39.
- Super Lotto PLUS: Pick any five numbers within the range of 1-47, and then one mega number from 1-27.
- Mega Millions: Select six numbers from two separate pool of numbers, five different numbers from 1-70, and one number from 1-25.
- Powerball: Select 5 numbers between 1-69, and one Powerball number between 1-26.

Script Hints

An easy way to solve this is to create a function that represents each type of lottery, and then fill in the respective logic. Once you know how to code The Daily 3, then you'll also know how to code the rest of them. The reason for this is because the logic is in essence the same, the only thing that changes are the amount of numbers to predict along with their ranges. Below are the functions to be coded:

- daily_3
- daily_4
- fantasy_5
- super_lotto_plus
- mega_millions
- powerball

The `daily_3` function

```
from random import randint
from time import sleep
def daily_3():
        """
        emulates the daily 3
        lottery
        """
        print("Welcome! You're playing Daily 3.")
        lucky_one = randint(0, 9)
        lucky_two = randint(0, 9)
        lucky_three = randint(0, 9)
        guess_one = int(input('Guess your lucky number: 0 - 9 '))
        guess_two = int(input('Guess your lucky number: 0 - 9 '))
        guess_three = int(input('Guess your lucky number: 0 - 9 '))
```

What it does?

Randomly generates three numbers within the range of 0-9, and asks the user for three guesses.

How to check if the user guessed correctly?

```
   if guess_one == lucky_one and guess_two == lucky_two and
guess_three == lucky_three:

        print("Congrats! You won $500")

    else:

        print("Today wasn't your day! Try again.")
```

We've Coded the daily 3!

Now, code the rest of the California lotteries. View the solution on GitHub: https://github.com/purcellconsult/Code-Cool-Stuff-With-Python/blob/master/sourcecode/ch_03/california_lottery.py