Day 13-100 of Data Science

by Eternaltek (Vamsidhar Reddy)

Python Projects

1 Number guessing game

In [10]:

```python
import random
import math
# Taking Inputs
lower = int(input("Enter Lower bound:- "))

# Taking Inputs
upper = int(input("Enter Upper bound:- "))

# generating random number between
# the lower and upper
x = random.randint(lower, upper)
print("\n\tYou've only ",
    round(math.log(upper - lower + 1, 2))," chances to guess the integer!\n")

# Initializing the number of guesses.
count = 0

# for calculation of minimum number of
# guesses depends upon range
while count < math.log(upper - lower + 1, 2):
    count += 1

    # taking guessing number as input
    guess = int(input("Guess a number:- "))

    # Condition testing
    if x == guess:
        print("Congratulations you did it in ",count, " try")
# Once guessed, loop will break
        break
    elif x > guess:
        print("You guessed too small!")
    elif x < guess:
        print("You Guessed too high!")

# If Guessing is more than required guesses,
# shows this output.
if count >= math.log(upper - lower + 1, 2):
    print("\nThe number is %d" % x)
    print("\tBetter Luck Next time!")

# Better to use This source Code on pycharm!
```

```
Enter Lower bound:- 4
Enter Upper bound:- 9

        You've only  3  chances to guess the integer!

Guess a number:- 5
You guessed too small!
Guess a number:- 9
You Guessed too high!
Guess a number:- 2
You guessed too small!

The number is 6
        Better Luck Next time!
```

2 Password Generator

In [15]:

```python
import random

#A function do shuffle all the characters of a string
def shuffle(string):
    tempList = list(string)
    random.shuffle(tempList)
    return ''.join(tempList)

#Main program starts here
uppercaseLetter1=chr(random.randint(65,90)) #Generate a random Uppercase letter (based
uppercaseLetter2=chr(random.randint(65,90)) #Generate a random Uppercase letter (based
#Generate more characters here
#....

#Generate password using all the characters, in random order
password = uppercaseLetter1 + uppercaseLetter2 # + ....
password = shuffle(password)

#Ouput
print(password)
```

```
ER
```

3 Rock, Paper, Scissors Game

In [ ]:

```python
from random import randint

#create a list of play options
t = ["Rock", "Paper", "Scissors"]

#assign a random play to the computer
computer = t[randint(0,2)]

#set player to False
player = False

while player == False:
#set player to True
    player = input("Rock, paper, Scissors?")
    if player == computer:
        print("Tie!")
    elif player == "Rock":
        if computer == "Paper":
            print("You lose!", computer, "covers", player)
        else:
            print("You win!", player, "smashes", computer)
    elif player == "Paper":
        if computer == "Scissors":
            print("You lose!", computer, "cut", player)
        else:
            print("You win!", player, "covers", computer)
    elif player == "Scissors":
        if computer == "Rock":
            print("You lose...", computer, "smashes", player)
        else:
            print("You win!", player, "cut", computer)
    else:
        print("That's not a valid play. Check your spelling!")
    #player was set to True, but we want it to be False so the loop continues
    player = False
    computer = t[randint(0,2)]
```

```
Rock, paper, Scissors?dtyy
That's not a valid play. Check your spelling!
Rock, paper, Scissors?ftyui
That's not a valid play. Check your spelling!
Rock, paper, Scissors?78'
That's not a valid play. Check your spelling!
Rock, paper, Scissors?78o'
That's not a valid play. Check your spelling!
```

4 Currency Converter

In [ ]:

```
# # Python Project on Currency Converter

import requests
from tkinter import *
import tkinter as tk
from tkinter import ttk


class RealTimeCurrencyConverter():
    def __init__(self,url):
            self.data = requests.get(url).json()
            self.currencies = self.data['rates']

    def convert(self, from_currency, to_currency, amount):
        initial_amount = amount
        if from_currency != 'USD' :
            amount = amount / self.currencies[from_currency]

        # limiting the precision to 4 decimal places
        amount = round(amount * self.currencies[to_currency], 4)
        return amount

class App(tk.Tk):

    def __init__(self, converter):
        tk.Tk.__init__(self)
        self.title = 'Currency Converter'
        self.currency_converter = converter

        #self.configure(background = 'blue')
        self.geometry("530x200")

        # Label
        self.intro_label = Label(self, text = 'IT SOURCECODE Real Time Currency Conver
        self.intro_label.config(font = ('Courier',15,'bold'))

        self.date_label = Label(self, text = f" Date : {self.currency_converter.data['

        self.intro_label.place(x = 10 , y = 5)
        self.date_label.place(x = 200, y= 50)

        # Entry box
        valid = (self.register(self.restrictNumberOnly), '%d', '%P')
        self.amount_field = Entry(self,bd = 3, relief = tk.RIDGE, justify = tk.CENTER,
        self.converted_amount_field_label = Label(self, text = '', fg = 'black', bg =

        # dropdown
        self.from_currency_variable = StringVar(self)
        self.from_currency_variable.set("INR") # default value
        self.to_currency_variable = StringVar(self)
        self.to_currency_variable.set("USD") # default value

        font = ("Courier", 12, "bold")
        self.option_add('*TCombobox*Listbox.font', font)
        self.from_currency_dropdown = ttk.Combobox(self, textvariable=self.from_curren
        self.to_currency_dropdown = ttk.Combobox(self, textvariable=self.to_currency_v

        # placing
        self.from_currency_dropdown.place(x = 30, y= 120)
        self.amount_field.place(x = 36, y = 150)
        self.to_currency_dropdown.place(x = 340, y= 120)
        #self.converted_amount_field.place(x = 346, y = 150)
        self.converted_amount_field_label.place(x = 346, y = 150)

        # Convert button
        self.convert_button = Button(self, text = "Convert", fg = "black", bg = "green
        self.convert_button.config(font=('Courier', 10, 'bold'))
        self.convert_button.place(x = 225, y = 135)

    def perform(self):
        amount = float(self.amount_field.get())
        from_curr = self.from_currency_variable.get()
        to_curr = self.to_currency_variable.get()

        converted_amount = self.currency_converter.convert(from_curr,to_curr,amount)
        converted_amount = round(converted_amount, 2)

        self.converted_amount_field_label.config(text = str(converted_amount))

    def restrictNumberOnly(self, action, string):
        regex = re.compile(r"[0-9,]*?(\.)?[0-9,]*$")
        result = regex.match(string)
        return (string == "" or (string.count('.') <= 1 and result is not None))

if __name__ == '__main__':
    url = 'https://api.exchangerate-api.com/v4/latest/USD'
    converter = RealTimeCurrencyConverter(url)

    App(converter)
    mainloop()
```

5 Number Guessing

In [ ]:

```
import random


def show_score(attempts_list):
    if not attempts_list:
        print('There is currently no best score,'
              ' it\'s yours for the taking!')

    else:
        print(f'The current best score is'
              f' {min(attempts_list)} attempts')


def start_game():
    attempts = 0
    rand_num = random.randint(1, 10)
    attempts_list = []

    print('Hello traveler! Welcome to the game of guesses!')
    player_name = input('What is your name? ')
    wanna_play = input(
        f'Hi, {player_name}, would you like to play '
        f'the guessing game? (Enter Yes/No): ')

    if wanna_play.lower() != 'yes':
        print('That\'s cool, Thanks!')
        exit()
    else:
        show_score(attempts_list)

    while wanna_play.lower() == 'yes':
        try:
            guess = int(input('Pick a number between 1 and 10: '))
            if guess < 1 or guess > 10:
                raise ValueError(
                    'Please guess a number within the given range')

            attempts += 1

            if guess == rand_num:
                attempts_list.append(attempts)
                print('Nice! You got it!')
                print(f'It took you {attempts} attempts')
                wanna_play = input(
                    'Would you like to play again? (Enter Yes/No): ')
                if wanna_play.lower() != 'yes':
                    print('That\'s cool, have a good one!')
                    break
                else:
                    attempts = 0
                    rand_num = random.randint(1, 10)
                    show_score(attempts_list)
                    continue
                else:
                    if guess > rand_num:
                        print('It\'s lower')
                    elif guess < rand_num:
                        print('It\'s higher')

        except ValueError as err:
            print('Oh no!, that is not a valid value. Try again...')
            print(err)


if __name__ == '__main__':
    start_game()
```

```
Hello traveler! Welcome to the game of guesses!
What is your name? Vamsidhar Reddy
Hi, Vamsidhar Reddy, would you like to play the guessing game? (Enter Yes/No): yes
There is currently no best score, it's yours for the taking!
Pick a number between 1 and 10: 8
It's higher
Pick a number between 1 and 10: 3
It's higher
Pick a number between 1 and 10: 1
It's higher
Pick a number between 1 and 10: No
Oh no!, that is not a valid value. Try again...
invalid literal for int() with base 10: 'No'
Pick a number between 1 and 10: 56
Oh no!, that is not a valid value. Try again...
Please guess a number within the given range
```

In [ ]:

In [ ]:

Follow us on Social Media

Linkedin: https://www.linkedin.com/company/eternaltek/about/?viewAsMember=true

Medium: https://medium.com/@eternaltek.info

WhatsApp Channel: https://whatsapp.com/channel/0029Va5onCbDjiOTi6D1vU36

Github: https://github.com/Vamsi-2203

In [ ]: