Age Calculator using Python

Age Calculator is an amazing application to create as a beginner in any programming language. To create an age calculator, you need two dates:

- today’s date
- date of birth

```
def ageCalculator(y, m, d):
   import datetime
   today = datetime.datetime.now().date()
   dob = datetime.date(y, m, d)
   age = int((today-dob).days / 365.25)
   print(age)
ageCalculator(1998, 9, 3)
```

Send Automatic Emails using Python

It is very important to generate a Google app password for your Gmail account, as you will be sending automatic emails using Python through your Gmail account. Once you’ve generated your Google app password, here’s how you can start the task of sending emails using Python:

```
import os
import random
import smtplib
def automatic_email():
   user = input("Enter Your Name >>: ")
   email = input("Enter Your Email >>: ")
   message = (f"Dear {user}, Welcome to Thecleverprogrammer")
   s = smtplib.SMTP('smtp.gmail.com', 587)
   s.starttls()
   s.login("Your Gmail Account", "Your App Password")
   s.sendmail('&&&&&&&&&&&', email, message)
   print("Email Sent!")
automatic_email()
```

Create Tables using Python

I hope you now have understood some of the important features provided by the tabulate module in Python. Now let’s see how to create tables using Python by using the tabulate module. Below is how you can create a very simple table using Python:

```
from tabulate import tabulate

data = [["Name", "Place", "Gender"], ["Aman", "New Delhi", "Male"],
["Hritika", "New Delhi", "Female"], ["Krishna", "UP", "Male"]]

print(tabulate(data))



#We can also design this table by adding a grid, here is how you can do it:

print(tabulate(data, headers='firstrow'))



#We can also make the grid look better:

print(tabulate(data, headers='firstrow', tablefmt='fancy_grid'))
```

Time Series Graph using Python

To visualize a time series graph using Python, I will be using a stock price dataset. There are many libraries in Python for data visualization; I will be using Plotly as it is easy to visualize interactive visualizations using plotly. So let’s start this task by collecting the latest stock price data of Apple:

```
import pandas as pd

import yfinance as yf

import datetime

from datetime import date, timedelta

today = date.today()
```

```
d1 = today.strftime("%Y-%m-%d")
end_date = d1
d2 = date.today() - timedelta(days=360)
d2 = d2.strftime("%Y-%m-%d")
start_date = d2

data = yf.download('AAPL',
                   start=start_date,
                   end=end_date,
                   progress=False)
print(data.head())
```

The above data is collected by using the yfinance API. Now below is how you can visualize a time series graph using Python:

```
import plotly.express as px
figure = px.line(data, x = data.index, y = "Close")
figure.show()
```

OTP Verification using Python

I hope you now have understood what is an OTP and how we can create an application for the task of OTP verification. Now let's follow the steps mentioned above by using Python to create an application for the task of OTP verification. I will start by importing the necessary Python library that we need for this task:

```
import os
import math
import random
import smtplib

digits="0123456789"
OTP=""
for i in range(6):
   OTP+=digits[math.floor(random.random()*10)]
otp = OTP + " is your OTP"
msg= otp
s = smtplib.SMTP('smtp.gmail.com', 587)
s.starttls()
s.login("Your Gmail Account", "You app password")
emailid = input("Enter your email: ")
s.sendmail('&&&&&&&&&&&',emailid,msg)
a = input("Enter Your OTP >>: ")
if a == OTP:
   print("Verified")
else:
   print("Please Check your OTP again")
```

Typing Speed Test Game with Python

To create a typing speed test user interface game with Python we first need to see the structure and the logic of the code that we need to use. So the code below shows how to write a program to create a Typing test game with Python:

```
from time import time


# calculate the accuracy of input prompt
def typingErrors(prompt):
    global iwords


    words = prompt.split()
    errors = 0


    for i in range(len(iwords)):
        if i in (0, len(iwords)-1):
            if iwords[i] == words[i]:
                continue
            else:
                errors +=1
        else:
            if iwords[i] == words[i]:
                if (iwords[i+1] == words[i+1]) & (iwords[i-1] == words[i-1]):
                    continue
                else:
                    errors += 1
            else:
                errors += 1
    return errors
```

```
# calculate the speed in words per minute
def typingSpeed(iprompt, stime, etime):
    global time
    global iwords

    iwords = iprompt.split()
    twords = len(iwords)
    speed = twords / time

    return speed

# calculate total time elapsed
def timeElapsed(stime, etime):
    time = etime - stime

    return time

if __name__ == '__main__':
    prompt = "Hi, my name is Aman Kharwal, I am a coding instructor."
    print("Type this:- '", prompt, "'")

    # listening to input ENTER
    input("press ENTER when you are ready!")

    # recording time for input
    stime = time()

    iprompt = input()

    etime = time()


    # gather all the information returned from functions

    time = round(timeElapsed(stime, etime), 2)

    speed = typingSpeed(iprompt, stime, etime)

    errors = typingErrors(prompt)


    # printing all the required data

    print("Total Time elapsed : ", time, "s")

    print("Your Average Typing Speed was : ", speed, "words / minute")

    print("With a total of : ", errors, "errors")
```

Contact Book with Python

In this section, I will walk you through how to create a basic contacts book with the Python programming language to store and find contacts. With a few modifications, you can use it as a project where you can implement the same algorithm on a database like MySQL to store the contacts.

```
names = []

phone_numbers = []

num = 3

for i in range(num):

    name = input("Name: ")

    phone_number = input("Phone Number: ") # for convert to int => int(input("Phone Number: "))
    names.append(name)

    phone_numbers.append(phone_number)

print("\nName\t\t\tPhone Number\n")

for i in range(num):

    print("{}\t\t\t{}".format(names[i], phone_numbers[i]))

search_term = input("\nEnter search term: ")

print("Search result:")

if search_term in names:

    index = names.index(search_term)

    phone_number = phone_numbers[index]

    print("Name: {}, Phone Number: {}".format(search_term, phone_number))

else:

    print("Name Not Found")
```

Hangman Game with Python

The logic of creating the Hangman game with python is that we will have users who will guess a letter and all users will have a very limited number of guesses.

To create the hangman game, I will start by making a list of secret words, then start choosing words at random. In the process, I'll just represent each word as "_" and whenever the user guesses the correct word, I'll replace the "_" with the correct word. Now let's follow this logic to create the hangman game with Python:

```
import time

import random

name = input("What is your name? ")

print ("Hello, " + name, "Time to play hangman!")

time.sleep(1)
```

```
print ("Start guessing...\n")
time.sleep(0.5)
## A List Of Secret Words
words = ['python','programming','treasure','creative','medium','horror']
word = random.choice(words)
guesses = ''
turns = 5
while turns > 0:
    failed = 0
    for char in word:
        if char in guesses:
            print (char,end="")
        else:
            print ("_",end=""),
            failed += 1
    if failed == 0:
        print ("\nYou won")
        break
    guess = input("\nguess a character:")
    guesses += guess
    if guess not in word:
        turns -= 1
        print("\nWrong")
        print("\nYou have", + turns, 'more guesses')
        if turns == 0:
            print ("\nYou Lose")
```

Digital Clock with Python

In this section, I will show you how to create a digital clock using python. This is a simple task to get started with the Tkinter library in Python, which is a built-in package that comes with Python. Tkinter has some cool features that can be used to build simple apps.

```
from tkinter import Label, Tk

import time

app_window = Tk()

app_window.title("Digital Clock")

app_window.geometry("420x150")

app_window.resizable(1,1)

text_font= ("Boulder", 68, 'bold')

background = "#f2e750"

foreground= "#363529"

border_width = 25

label = Label(app_window, font=text_font, bg=background, fg=foreground, bd=border_width)

label.grid(row=0, column=1)

def digital_clock():

   time_live = time.strftime("%H:%M:%S")

   label.config(text=time_live)

   label.after(200, digital_clock)

digital_clock()

app_window.mainloop()
```