**Python in 100 days udemy**

**Day1 :** Band name generator

print("Welcome to Sadegh day 1 pyhton in 100 days Band Name Generator")

b1 = input("What's name of the city you grew up?\n")

b2 = input("What's your pet's name?\n")

print(f"Your band name could be {b1} {b2}")

It concats 2 strings name to generate a band name.

**Day2 :** Tip calculator

print("Welcome to Sadegh day 2 pyhton in 100 days Tip Calculator")

bill = float(input("What was the total bill? $"))

percentage = int(input(

    "What percentage tip would you like to give? 10, 12 or 15? %"))

people = int(input("How many people to split the bill? "))

print(f"Each person should pay : ${

      round((round(bill, 2)/people)\*(1+percentage/100), 2)}")

It gets the bill and add the tip percentage to It and then divide the payment that each person should pay.

**Day3:** Treasure island (find treasure)

It’s for learning conditonals in python.

print('''

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

          |                   |                  |                     |

 \_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.=""\_;=.\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_

|                   |  ,-"\_,=""     `"=.|                  |

|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_"=.\_o`"-.\_        `"=.\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

          |                `"=.\_o`"=.\_      \_`"=.\_                     |

 \_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:=.\_o "=.\_."\_.-="'"=.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_

|                   |    \_\_.--" , ; `"=.\_o." ,-"""-.\_ ".   |

|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_.\_"  ,. .` ` `` ,  `"-.\_"-.\_   ". '\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

          |           |o`"=.\_` , "` `; .". ,  "-.\_"-.\_; ;              |

 \_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_| ;`-.o`"=.\_; ." ` '`." ` . "-.\_ /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_

|                   | |o;    `"-.o`"=.\_``  '` " ,\_\_.--o;   |

|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_| ;     (#) `-.o `"=.`\_.--"\_o.-; ;\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_/\_\_\_\_\_\_/\_\_\_\_\_\_/\_\_\_|o;.\_    "      `".o|o\_.--"    ;o;\_\_\_\_/\_\_\_\_\_\_/\_\_\_\_\_\_/\_\_\_\_

/\_\_\_\_\_\_/\_\_\_\_\_\_/\_\_\_\_\_\_/\_"=.\_o--.\_        ; | ;        ; ;/\_\_\_\_\_\_/\_\_\_\_\_\_/\_\_\_\_\_\_/\_

\_\_\_\_/\_\_\_\_\_\_/\_\_\_\_\_\_/\_\_\_\_\_\_/\_\_"=.\_o--.\_   ;o|o;     \_.\_;o;\_\_\_\_/\_\_\_\_\_\_/\_\_\_\_\_\_/\_\_\_\_

/\_\_\_\_\_\_/\_\_\_\_\_\_/\_\_\_\_\_\_/\_\_\_\_\_\_/\_\_\_\_"=.\_o.\_; | ;\_.--"o.--"\_/\_\_\_\_\_\_/\_\_\_\_\_\_/\_\_\_\_\_\_/\_

\_\_\_\_/\_\_\_\_\_\_/\_\_\_\_\_\_/\_\_\_\_\_\_/\_\_\_\_\_\_/\_\_\_\_\_"=.o|o\_.--""\_\_\_/\_\_\_\_\_\_/\_\_\_\_\_\_/\_\_\_\_\_\_/\_\_\_\_

/\_\_\_\_\_\_/\_\_\_\_\_\_/\_\_\_\_\_\_/\_\_\_\_\_\_/\_\_\_\_\_\_/\_\_\_\_\_\_/\_\_\_\_\_\_/\_\_\_\_\_\_/\_\_\_\_\_\_/\_\_\_\_\_\_/\_\_\_\_\_ /

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

''')

print("Welcome to Sadegh day 3 pyhton in 100 days Treasure Island.")

print("Your mission is to find the treasure.")

step1 = input(

    "you're at a cross road. Where do you want to go? Type 'left' or 'right'\n").lower()

if step1 == "right":

    print("You fell into a hole. Game over!")

elif step1 == "left":

    step2 = input("You come to a lake. There is an island in the middle of the lake. Type 'wait' to wait for a boat or type 'swim' to swim across.\n").lower()

    if step2 == "swim":

        print("You got attacked by an angry trout. Game over!")

    elif step2 == "wait":

        step3 = input(

            "You arrive at the island unharmed. There is a house with 3 doors. One red, one yellow and one blue. Which color do you choose?\n").lower()

        if step3 == "yellow":

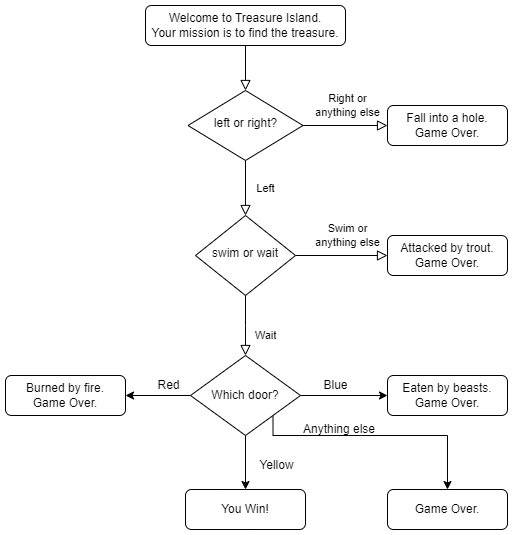
            print("You found the treasure! You win!")

        elif step3 == "red":

            print("It's a room full of fire. Game Over!")

        elif step3 == "blue":

            print("You entered a room of beasts. Game Over!")

It’s based on this flowchart.

**Day4:** Rock, Paper, Scissors

import random

rock = '''

    \_\_\_\_\_\_\_

---'   \_\_\_\_)

      (\_\_\_\_\_)

      (\_\_\_\_\_)

      (\_\_\_\_)

---.\_\_(\_\_\_)

'''

paper = '''

    \_\_\_\_\_\_\_

---'   \_\_\_\_)\_\_\_\_

          \_\_\_\_\_\_)

          \_\_\_\_\_\_\_)

         \_\_\_\_\_\_\_)

---.\_\_\_\_\_\_\_\_\_\_)

'''

scissors = '''

    \_\_\_\_\_\_\_

---'   \_\_\_\_)\_\_\_\_

          \_\_\_\_\_\_)

       \_\_\_\_\_\_\_\_\_\_)

      (\_\_\_\_)

---.\_\_(\_\_\_)

'''

print("Welcome to Sadegh day 4 pyhton in 100 days Rock, Paper and Scissors.")

playerChoiceNum = int(

    input("What do you choose? Type 0 for Rock, 1 for Paper or 2 for Scissors.\n")

)

computerChoiceNum = random.randint(0, 2)

User chooses based on its opinion and computer chooses randomly from 0 and 1 and 2 randomly.

if playerChoiceNum == 0:

    print("You chose:\n", rock)

elif playerChoiceNum == 1:

    print("You chose:\n", paper)

elif playerChoiceNum == 2:

    print("You chose:\n", scissors)

if computerChoiceNum == 0:

    print("Computer chose:\n", rock)

elif computerChoiceNum == 1:

    print("Computer chose:\n", paper)

elif computerChoiceNum == 2:

    print("Computer chose:\n", scissors)

This is the code for output of the chose if user and computer.

if playerChoiceNum == computerChoiceNum:

    print("You Draw!")

elif playerChoiceNum == 0 and computerChoiceNum == 2 or playerChoiceNum == 1 and computerChoiceNum == 0 or playerChoiceNum == 2 and computerChoiceNum == 1:

    print("You Won!")

else:

    print("You Lost!")

And it’s the code for the logic that shows who will win.

**Day5:** Password generator

import random

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']

numbers = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9']

symbols = ['!', '#', '$', '%', '&', '(', ')', '\*', '+']

print("Welcome to Sadegh day 5 python in 100 days PyPassword Generator!")

lettersLength = int(

    input("How many letters would you like in your password?\n"))

symbolsLength = int(

    input("How many symbols would you like in your password?\n"))

numbersLength = int(

    input("How many numbers would you like in your password?\n"))

The password must be made of letters and numbers and symbols to be strong. We make list of them and in next step choose randomly from them.

passwordArr = []

passwordArr.extend(random.choices(letters, k=lettersLength))

passwordArr.extend(random.choices(symbols, k=symbolsLength))

passwordArr.extend(random.choices(numbers, k=numbersLength))

random.shuffle(passwordArr)

print("".join(passwordArr))

We make an empty list and add random letters and numbers and symbols to it but they are next to each other and need to be shuffled. After shuffling the result which is a string made of the list members is shown.

**Day6:** A game for loops

It was online and with a website

**Day7:** Hangman

import random

import os

print(r"""

 \_   \_

| | | |

| |\_| | \_\_ \_ \_ \_\_   \_\_ \_ \_ \_\_ \_\_\_   \_\_ \_ \_ \_\_

|  \_  |/ \_` | '\_ \ / \_` | '\_ ` \_ \ / \_` | '\_ \

| | | | (\_| | | | | (\_| | | | | | | (\_| | | | |

\\_| |\_/\\_\_,\_|\_| |\_|\\_\_, |\_| |\_| |\_|\\_\_,\_|\_| |\_|

                    \_\_/ |

                   |\_\_\_/

Welcome to sadegh day 7 python in 100 days Hangman

""")

wordCategories = {

    "fruits": [

        "apple", "banana", "orange", "grape", "kiwi", "strawberry", "pineapple",

        "watermelon", "blueberry", "peach", "mango", "pear", "cherry", "lemon",

        "lime", "coconut", "apricot", "fig", "plum", "papaya", "raspberry",

        "blackberry", "cranberry", "melon"

    ],

    "animals": [

        "dog", "cat", "elephant", "lion", "tiger", "giraffe", "zebra", "monkey",

        "panda", "kangaroo", "koala", "crocodile", "hippopotamus", "rhinoceros",

        "cheetah", "bear", "wolf", "fox", "rabbit", "deer", "squirrel", "horse",

        "cow", "pig"

    ],

    "colors": [

        "red", "orange", "yellow", "green", "blue", "purple", "pink", "brown",

        "black", "white", "gray", "silver", "gold", "beige", "maroon", "cyan",

        "magenta", "olive", "teal", "navy"

    ],

    "countries": [

        "usa", "canada", "mexico", "brazil", "argentina", "uk", "france", "germany",

        "italy", "spain", "russia", "china", "japan", "india", "australia",

        "southafrica", "egypt", "nigeria", "kenya", "saudiarabia", "unitedarabemirates",

        "turkey", "iran", "pakistan"

    ],

    "vehicles": [

        "car", "bus", "truck", "motorcycle", "bicycle", "train", "airplane",

        "helicopter", "boat", "submarine", "ship", "scooter", "van", "ambulance",

        "firetruck", "policecar", "tractor", "forklift", "jet", "spaceship"

    ],

    "professions": [

        "doctor", "teacher", "engineer", "lawyer", "artist", "chef", "scientist",

        "programmer", "writer", "musician", "actor", "athlete", "pilot", "policeofficer",

        "firefighter", "nurse", "dentist", "architect", "veterinarian", "farmer"

    ],

    "sports": [

        "soccer", "basketball", "football", "tennis", "volleyball", "golf",

        "cricket", "baseball", "swimming", "running", "cycling", "boxing",

        "rugby", "hockey", "skiing", "snowboarding", "surfing", "skateboarding",

        "wrestling", "gymnastics"

    ],

    "foods": [

        "pizza", "hamburger", "spaghetti", "sushi", "sandwich", "salad", "steak",

        "chicken", "soup", "rice", "taco", "burrito", "pasta", "pancake", "waffle",

        "omelette", "burger", "fries", "hotdog", "popcorn"

    ],

    "movies": [

        "avatar", "titanic", "starwars", "thelordoftherings", "harrypotter",

        "jurassicpark", "thematrix", "forrestgump", "thegodfather", "inception",

        "thelionking", "frozen", "findingnemo", "toystory", "backtothefuture",

        "thedarkknight", "pulpfiction", "theshawshankredemption", "fightclub",

        "interstellar"

    ],

    "languages": [

        "english", "spanish", "french", "german", "chinese", "arabic", "russian",

        "japanese", "portuguese", "italian", "korean", "dutch", "turkish",

        "swedish", "polish", "hindi", "greek", "hebrew", "thai", "vietnamese"

    ],

    "emotions": [

        "happy", "sad", "angry", "excited", "nervous", "surprised", "fearful",

        "disgusted", "content", "confused", "bored", "loved", "hated", "jealous",

        "relaxed", "stressed", "hopeful", "proud", "shy", "embarrassed"

    ]

}

hangmanStages = [

    r"""

      +---+

          |

          |

          |

          |

          |

    =========

    """,

    r"""

      +---+

      |   |

          |

          |

          |

          |

    =========

    """,

    r"""

      +---+

      |   |

      O   |

          |

          |

          |

    =========

    """,

    r"""

      +---+

      |   |

      O   |

      |   |

          |

          |

    =========

    """,

    r"""

      +---+

      |   |

      O   |

     /|   |

          |

          |

    =========

    """,

    r"""

      +---+

      |   |

      O   |

     /|\  |

          |

          |

    =========

    """,

    r"""

      +---+

      |   |

      O   |

     /|\  |

     /    |

          |

    =========

    """,

    r"""

      +---+

      |   |

      O   |

     /|\  |

     / \  |

          |

    =========

    """

]

These are the basics and ascii art.

def select\_random\_word(dict):

    category = random.choice(list(dict.keys()))

    word = random.choice(dict[category])

    return category, word

category, word = select\_random\_word(wordCategories)

This function randomy choose a category from the wordCategories dictationary and then select a random word from that category list.

guessedWord = []

for i in word:

    guessedWord.append("\_")

hangmanStage = 0

This code is for the status of the word that user is guessing.

def find\_indexes(string, character):

    indexes = [index for index, char in enumerate(string) if char == character]

    return indexes

This function which has 2 parameters, find indexes of a character that we are searching in a word and return their indexes as a list.

while True:

    counter = 0

    guessedLetter = input(f"Guess a letter, the word is in {category} category : ")

    os.system('cls' if os.name == 'nt' else 'clear')

    if guessedLetter in word:

        for i in find\_indexes(word, guessedLetter):

            guessedWord[i] = guessedLetter

        for i in guessedWord:

            print(i, end=' ')

        print(f"\n{hangmanStages[hangmanStage]}")

This is for the part correct guess by the user

else:

        for i in guessedWord:

            print(i, end=' ')

        print(f"\n\nyou guessed {

              guessedLetter}, that's not in the word, you lose a life")

        if hangmanStage < len(hangmanStages)-2:

            hangmanStage += 1

            print(hangmanStages[hangmanStage])

        else:

            os.system('csls' if os.name == 'nt' else 'clear')

            print(f"The word was {word}")

            print(hangmanStages[len(hangmanStages)-1])

            print(r"""

 \_\_\_\_\_                        \_\_\_\_\_                  \_

|  \_\_ \                      |  \_  |                | |

| |  \/ \_\_ \_ \_ \_\_ \_\_\_   \_\_\_  | | | |\_   \_\_\_\_\_ \_ \_\_  | |

| | \_\_ / \_` | '\_ ` \_ \ / \_ \ | | | \ \ / / \_ \ '\_\_| | |

| |\_\ \ (\_| | | | | | |  \_\_/ \ \\_/ /\ V /  \_\_/ |    |\_|

 \\_\_\_\_/\\_\_,\_|\_| |\_| |\_|\\_\_\_|  \\_\_\_/  \\_/ \\_\_\_|\_|    (\_)

            """)

            break

At this part is for the mistake guesses.Also check that the game is over or not. Each time that user does a mistake, the hangmanStage increases.

    for i in guessedWord:

        if i == "\_":

            counter += 1

    if counter == 0:

        print(r"""

\_\_   \_\_                               \_

\ \ / /                    (\_)       | |

 \ V /\_\_\_  \_   \_  \_\_      \_\_\_ \_ \_\_   | |

  \ // \_ \| | | | \ \ /\ / / | '\_ \  | |

  | | (\_) | |\_| |  \ V  V /| | | | | |\_|

  \\_/\\_\_\_/ \\_\_,\_|   \\_/\\_/ |\_|\_| |\_| (\_)

                  """)

        break

It counts that do we have any empty space or not and if all the spaces were filled with characters it means that user guessed the word and is winner.

**Day8: Encrypter and Decrypter**

import time

art = (

    " ,-----.                                  \n"

    "'  .--./,--,--.,---.  ,---. ,--,--.,--.--. \n"

    "|  |   ' ,-.  | .-. :(  .-'' ,-.  ||  .--' \n"

    "'  '--'\\ '-'  \\   --..-'  `) '-'  ||  |    \n"

    " `-----'`--`--'`----'`----' `--`--'`--'    \n"

    "                                            \n"

    "                                            \n"

    " ,-----.,--.       ,--.                     \n"

    "'  .--./`--' ,---. |  ,---. ,---. ,--.--.   \n"

    "|  |    ,--.| .-. ||  .-.  | .-. :|  .--'   \n"

    "'  '--'\\|  || '-' '|  | |  \\   --.|  |      \n"

    " `-----'`--'|  |-' `--' `--'`----'`--'      \n"

    "            `--'                            \n"

)

alphabets = ['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']

These are the basics.

def encrypt(text, shiftNum):

    newText = ""

    for char in text:

        if char in alphabets:

            newText += alphabets[(alphabets.index(char) + shiftNum) % 26]

        else:

            newText += char

    return newText

def decrypt(text, shiftNum):

    newText = ""

    for char in text:

        if char in alphabets:

            newText += alphabets[(alphabets.index(char) - shiftNum) % 26]

        else:

            newText += char

    return newText

This is the code of encoding and decoding which has 2 parameters which is the text wanna be encoded or decoded and other one which is the shift number. Its shift in alphabets list if its an alphabet otherwise it return the char as it was.

while True:

    os.system('cls' if os.name == 'nt' else 'clear')

    print(art + "\n")

    print("Welcome to sadegh day 8 python in 100 days cryption app")

    try:

        selection = input(

            "Enter 'encode' to encrypt, 'decode' to decrypt:\n").lower()

        if selection == "encode":

            text = input("Enter the text to be encoded: ").lower()

            while True:

                try:

                    shiftNum = int(input("Enter the shift number: "))

                    break

                except ValueError:

                    print("Invalid input. Please enter a valid number for the shift.")

                    time.sleep(3)

            print(f"Encoded text: {encrypt(text, shiftNum)}")

        elif selection == "decode":

            text = input("Enter the text to be decoded: ").lower()

            while True:

                try:

                    shiftNum = int(input("Enter the shift number: "))

                    break

                except ValueError:

                    print("Invalid input. Please enter a valid number for the shift.")

                    time.sleep(3)

            print(f"Decoded text: {decrypt(text, shiftNum)}")

        else:

            print("Invalid input. Please enter 'encode' or 'decode'.")

            time.sleep(3)

            os.system('cls' if os.name == 'nt' else 'clear')

            continue

        while True:

            finished = input("Are you finished? (yes/no): ").lower()

            if finished == 'yes':

                break

            elif finished == 'no':

                break

            else:

                print("Invalid input. Please enter 'yes' or 'no'.")

                time.sleep(3)

        if finished == 'yes':

            break

    except Exception as e:

        print(f"An unexpected error occurred: {e}")

        time.sleep(3)

        os.system('cls' if os.name == 'nt' else 'clear')

At first we choose to encode or decode and then we have the text and shift number. Also the user is asked about wanna carry on or stop. The code has error handling.