

Guess the Number Python

```
while True:
    # Import random module.
    import random

    # Choose an upper limit and a lower limit.
    # Use the int function to ensure the value entered is an integer.
    lower_limit = int(input("Enter the lower limit: "))
    upper_limit = int(input("Enter upper limit: "))

    # We select a number randomly and store it in variable
    # The function takes the upper limit and the lower limit as parameters and picks a number between the two numbers.
    # In python, variables can be declared and assigned at the same time
    random_number = random.randint(lower_limit, upper_limit)
    print("You will have to choose a number between ", upper_limit, " and ", lower_limit)

    # We assign a variable "Chances" that will act as the counter for a loop
    # The user will have to input his guess so we assign his guess into a variable.
    chances = 0
    while chances < 8:
        chances += 1
        guess = int(input("Enter your guess: "))
        if random_number == guess:
            print("Congragulations, you did it. The number was ", random_number)
            break
        elif guess < random_number:
            print("You guessed a small number.")
        elif guess > random_number:
            print("You guessed a large number.")
        if chances == 7:
            print("\n You've run out of chances")
            print("\n The number was ", random_number)
            print("Better luck next time")
            break
    print("\n")
    break
```

GUI Calculator Using Tkinter In Python

```
# importing the tkinter module
from tkinter import *

# initializing the tkinter
root = Tk()

# setting the width and height of the gui
root.geometry("430x500") # x is small case here

# declaring an empty string variable
expression = ""

# defining function which will set expressions and answers to the user
def setexpression(num):
    global expression
    expression = expression + str(num)
    value.set(expression)

# defining a function to calculate the expression entered by the user
def calculator():
    try:
        global expression
        answer = str(eval(expression))
        value.set(answer)
    except:
        value.set("Enter correct expression")
        expression = ""

# function to clear everything in expression
def clear():
    global expression
    expression = ""
    value.set(expression)
```

```

# declaring font variables as ("Language", size)
large_font = ('Verdana', 15)
small_font = ('Verdana', 10)

# declaring variable to take value of expression entered by the user
value = StringVar(value="Enter expression")

# entry widget to take expression from user and to show
# calculations
Entry(root, textvariable=value, font=large_font).grid(row=0,
    column=0, columnspan=4, ipadx=70)

# Now, there are some most basic buttons which should be present
# in a calculator
# here, each button is calling the setexpression function which
# is used to set values in the entry widget entered by the user
# on pressing the buttons 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, .
Button(root, text="+", fg="red", command=lambda:
    setexpression("+"), height=4,width=8).grid(row=1,column=0,pady=10)
Button(root, text="-", fg="red", command=lambda:
    setexpression("-"), height=4, width=8).grid(row=2, column=0, pady=10)
Button(root, text="X", fg="red", command=lambda:
    setexpression("*"), height=4, width=8).grid(row=3, column=0,pady=10)
Button(root, text="/", fg="red", command=lambda:
    setexpression("/"), height=4, width=8).grid(row=4, column=0,pady=10)
Button(root, text="1", fg="red", command=lambda:
    setexpression("1"), height=4, width=8).grid(row=1, column=1,pady=10)
Button(root, text="2", fg="red", command=lambda:
    setexpression("2"), height=4, width=8).grid(row=1, column=2,pady=10)
Button(root, text="3", fg="red", command=lambda:
    setexpression("3"), height=4, width=8).grid(row=1, column=3,pady=10)
Button(root, text="4", fg="red", command=lambda:
    setexpression("4"), height=4, width=8).grid(row=2, column=1,pady=10)
Button(root, text="5", fg="red", command=lambda:
    setexpression("5"), height=4, width=8).grid(row=2, column=2)
Button(root, text="6", fg="red", command=lambda:
    setexpression("6"), height=4, width=8).grid(row=2, column=3,pady=10)
Button(root, text="7", fg="red", command=lambda:
    setexpression("7"), height=4, width=8).grid(row=3, column=1,pady=10)
Button(root, text="8", fg="red", command=lambda:
    setexpression("8"), height=4, width=8).grid(row=3, column=2,pady=10)
Button(root, text="9", fg="red", command=lambda:
    setexpression("9"), height=4, width=8).grid(row=3, column=3,pady=10)
Button(root, text="0", fg="red", command=lambda:
    setexpression("0"), height=4, width=8).grid(row=4, column=2,pady=10)
Button(root, text=".", fg="red", command=lambda:
    setexpression("."), height=4, width=8).grid(row=4, column=1,pady=10)

# "=" button to call the calculator button which will return and
# show the calculated value in the entry widget
Button(root, text="=", fg="red", command=calculator, height=4,
width=8).grid(row=4, column=3, pady=10)

# "Clear" button to call the clear function which will clear the
# entry widget so that the user can start calculating again
Button(root, text="Clear", fg="red", command=clear, height=4,
    width=20).grid(row=5, column=1, pady=10)

# .mainloop() is used when the code is ready to run
root.mainloop()

```

Countdown Timer

```

# import the time module
import time

# define the countdown func.
def countdown(t):

    while t:
        mins, secs = divmod(t, 60)
        timer = '{:02d}:{:02d}'.format(mins, secs)
        print(timer, end="\r")
        time.sleep(1)

```

```

t -= 1

print('Fire in the hole!!')

# input time in seconds
t = input("Enter the time in seconds: ")

# function call
countdown(int(t))

```

Generate QR Code in Python

```

import qrcode
from tkinter import *

cp = Tk()
cp.title('copyassignment.com')
cp.geometry('700x250')
cp.config(bg='#e52165')

def generate():
    img = qrcode.make(msg.get())
    type(img)
    img.save(f'{save_name.get()}.png')
    Label(cp, text='File Saved!', bg='#e52165' , fg='black', font=('Arial Black', 8)).pack()

def show():
    img = qrcode.make(msg.get())
    type(img)
    img.show()

frame = Frame(cp, bg='#e52165')
frame.pack(expand=True)

#-----ENTER THE TEXT OR URL-----

Label(frame, text='Enter the Text or URL : ', font=('Arial Black', 16),
      bg='#e52165').grid(row=0, column=0, sticky='w')

msg = Entry(frame)
msg.grid(row=0, column=1)

#-----ENTER THE FILE NAME-----

Label(frame, text='File Name(Save As) : ', font=('Arial Black', 16),
      bg='#e52165').grid(row=1, column=0, sticky='w')

save_name = Entry(frame)
save_name.grid(row=1, column=1)

#-----BUTTONS TO SHOW OR SAVE QR CODE-----

btn = Button(cp, text='Show QR code', bd='5', command=show, width=15)
btn.pack()
btn = Button(cp, text='Save QR code', command=generate, bd='5', width=15)
btn.pack()

cp.mainloop()

```

GUI Age Calculator

```

# importing the tkinter module
from tkinter import *

# importing date class from datetime module
from datetime import date

# initializing tkinter
root = Tk()

# setting the width and height of gui
root.geometry("700x500")

```

```

# setting the title of gui
root.title("Age Calculator")

# loading image
photo = PhotoImage(file="filename.png")

# creating a Label widget to show the image we load
myimage = Label(image=photo)

# placing the image Label widget using grid method
myimage.grid(row=0, column=1)

# defining function to calculate age according to the data given
# by the user
def calculateAge():
    # storing today's date in "today" variable
    today = date.today()
    # getting birthdate using .get() method
    birthDate = date(int(yearEntry.get()), int(monthEntry.get()), int(dayEntry.get()))
    # calculating age by subtracting birthdate from today's date
    age = today.year - birthDate.year - ((today.month, today.day) < (birthDate.month, birthDate.day))
    # creating a Label widget to show the calculated age using
    # grid method
    Label(text=f"{nameValue.get()} your age is {age}").grid(row=6, column=1)

# creating a label widget for asking user his/her name
Label(text="Name").grid(row=1, column=0, padx=90)

# creating a label widget for asking user his/her year of birth
Label(text="Year").grid(row=2, column=0)

# creating a label widget for asking user his/her month of birth
Label(text="Month").grid(row=3, column=0)

# creating a label widget for asking user his/her day of birth
Label(text="Day").grid(row=4, column=0)

# declaring a variable of string datatype to store the name value
# entered by the user
nameValue = StringVar()

# declaring a variable of string datatype to store the year value
# entered by the user
yearValue = StringVar()

# declaring a variable of string datatype to store the month
# value entered by the user
monthValue = StringVar()

# declaring a variable of string datatype to store the day value
# entered by the user
dayValue = StringVar()

# creating an entry widget to take name value
nameEntry = Entry(root, textvariable=nameValue)

# creating an entry widget to take year value
yearEntry = Entry(root, textvariable=yearValue)

# creating an entry widget to take month value
monthEntry = Entry(root, textvariable=monthValue)

# creating an entry widget to take day value
dayEntry = Entry(root, textvariable=dayValue)

# placing the entry widgets
nameEntry.grid(row=1, column=1, pady=10)
yearEntry.grid(row=2, column=1, pady=10)
monthEntry.grid(row=3, column=1, pady=10)
dayEntry.grid(row=4, column=1, pady=10)

# creating and placing a button to calculate a show age on
# clicking on this button
Button(text="Calculate age", command=calculateAge).grid(row=5, column=1, pady=10)

```

```
# mainloop() is an infinite loop used to run the application when
# it's in ready state
root.mainloop()
```

Random Story Generator

```
import random

# list of books is stored in the list -'books'
books = ['Mother', 'Midnight wlof', 'My experiments with truth']

# An item from the list 'books' is selected
# by random.choice()
print(random.choice(books))
```

Password Generator GUI Application

```
# importing the tkinter module
from tkinter import *

# importing the pyperclip module to use it to copy our generated
# password to clipboard
import pyperclip

# random module will be used in generating the random password
import random

# initializing the tkinter
root = Tk()

# setting the width and height of the gui
root.geometry("400x400") # x is small case here

# declaring a variable of string type and this variable will be
# used to store the password generated
passstr = StringVar()

# declaring a variable of integer type which will be used to
# store the length of the password entered by the user
passlen = IntVar()

# setting the length of the password to zero initially
passlen.set(0)

# function to generate the password
def generate():
    # storing the keys in a list which will be used to generate
    # the password
    pass1 = ['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', '1', '2', '3', '4', '5', '6', '7', '8',
             '9', '0', ' ', '!', '@', '#', '$', '%', '^', '&',
             '*', '(', ')']

    # declaring the empty string
    password = ""

    # loop to generate the random password of the length entered
    # by the user
    for x in range(passlen.get()):
        password = password + random.choice(pass1)

    # setting the password to the entry widget
    passstr.set(password)

# function to copy the password to the clipboard
def copytoclipboard():
    random_password = passstr.get()
    pyperclip.copy(random_password)
```

```
# Creating a text label widget
Label(root, text="Password Generator Application", font="calibri 20 bold").pack()

# Creating a text label widget
Label(root, text="Enter password length").pack(pady=3)

# Creating a entry widget to take password length entered by the
# user
Entry(root, textvariable=passlen).pack(pady=3)

# button to call the generate function
Button(root, text="Generate Password", command=generate).pack(pady=7)

# entry widget to show the generated password
Entry(root, textvariable=passstr).pack(pady=3)

# button to call the copytoclipboard function
Button(root, text="Copy to clipboard", command=copytoclipboard).pack()

# mainloop() is an infinite loop used to run the application when
# it's in ready state
root.mainloop()
```

Thank you by Nivetha Rajam

