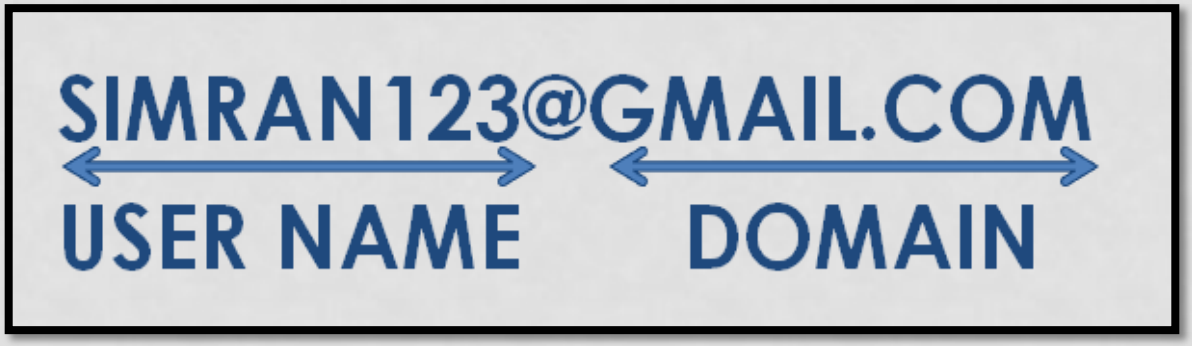
E MAIL SLICER

Email short for **electronic mail** is a method of exchanging messages between people using electronic devices, along with the web. It allows you to send and receive messages to and from anyone with an email address, anywhere in the world.

E-MAIL ADDRESS BREAKDOWN:

- The first portion of all e-mail addresses, the part before the @ symbol, contains the alias, user, group, or department of a company.
- Next, the @ (at sign) is a divider in the e-mail address. It's required for all SMTP e-mail addresses
- Finally, gmail.com is the domain.



SLICE METHOD() [BASIC PYTHON]

- The slice() function returns a slice object.
- A slice object is used to specify how to slice a sequence. You can specify where to start the slicing, and where to end. You can also specify the step, which allows you to e.g. slice only every other item.

SYNTAX :-

```
slice(start, end, step)
```

SPLIT METHOD

- The split() method splits a string into a list.
- You can specify the separator, default separator is any whitespace.

Example.

INPUT:

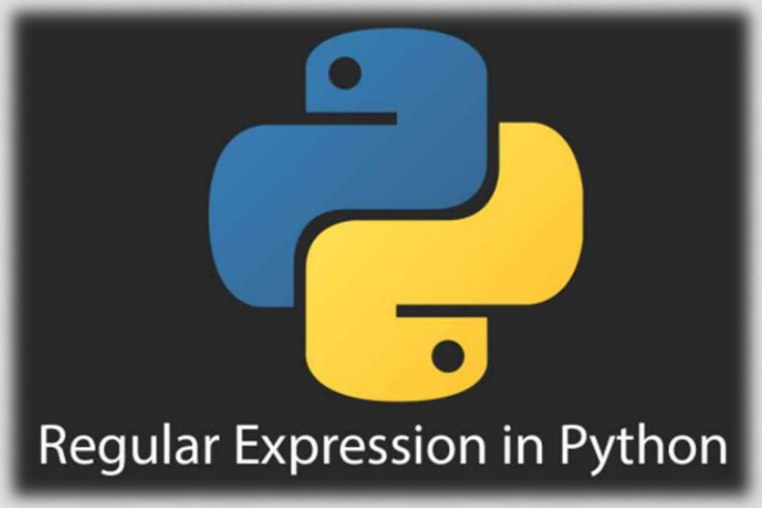
```
txt = "hello, my name is Peter, I am 26 years old"

x = txt.split(", ")

print(x)
```

OUTPUT:

```
['hello', 'my name is Peter', 'I am 26 years old']
```

RE MODULE

A *regular expression* is a special sequence of characters that helps you match or find other strings or sets of strings, using a specialized syntax held in a pattern.



SPECIAL CHARACTERS

SYMBOLS	DESCRIPTION
. (DOT)	In the default mode, this matches any character except a newline.
^ (Caret.)	Matches the start of the string.
$(DOLLAR)	Matches the end of the string or just before the newline at the end of the string.
***(MUL.)**	Causes the resulting RE to match 0 or more repetitions of the preceding RE, as many repetitions as are possible.
+(ADD)	Causes the resulting RE to match 1 or more repetitions of the preceding RE.
?(QUES.)	Causes the resulting RE to match 0 or 1 repetitions of the preceding RE.ab? will match either ‘a’ or ‘ab’.
\(BACK.)	Either escapes special characters (permitting you to match characters like '*', '?', and so forth), or signals a special sequence;

REGULAR EXPRESSION METHODS

The "re" package provides several methods to actually perform queries on an input string. We will see the methods of re in Python:

- re.match()
- re.search()
- re.findall()

RE.MATCH()

re.match() function of re in Python will search the regular expression pattern and return the first occurrence. The Python RegEx Match method checks for a match only at the beginning of the string. So, if a match is found in the first line, it returns the match object. But if a match is found in some other line, the Python RegEx Match function returns null.

RE.SEARCH()

re.search() function will search the regular expression pattern and return the first occurrence. Unlike Python re.match(), it will check all lines of the input string. The Python re.search() function returns a match object when the pattern is found and “null” if the pattern is not found

RE.FINDALL()

findall() module is used to search for “all” occurrences that match a given pattern. In contrast, search() module will only return the first occurrence that matches the specified pattern. findall() will iterate over all the lines of the file and will return all non-overlapping matches of pattern in a single step.

GUI(TKINTER)

- Tkinter is the most common, fast and easy to use Python package to build GUI application.
- To install the library, you can use pip install command to the command prompt.

```
pip install tkinter
```

GEOMETRIC MANAGER

There are mainly three geometry manager classes.

Pack() method

Grid() method

Place() method

WIDGETS USED

- Label

```
w = Label ( master, option, ... )
```

- Entry

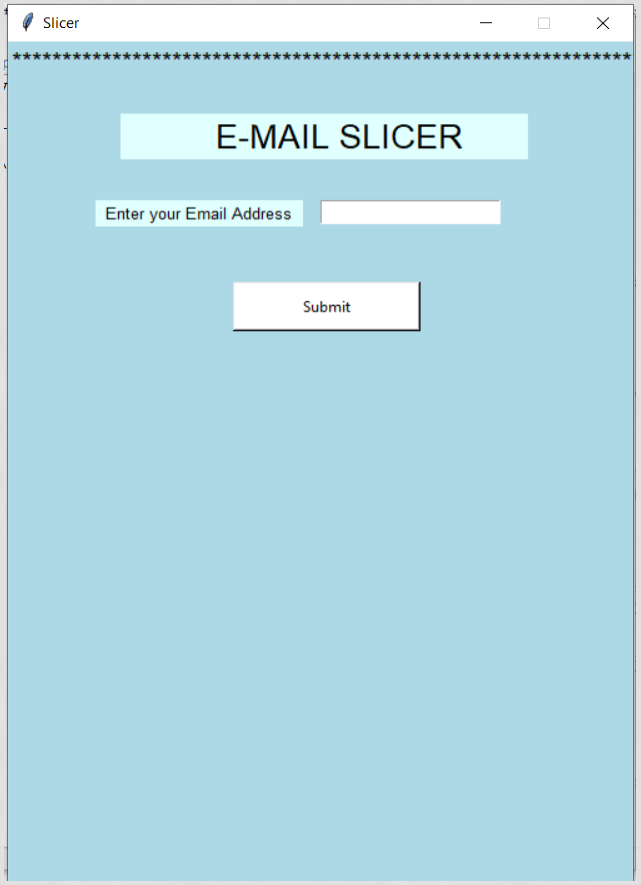
```
entry = tk.Entry(parent, options)
```

- Button

```
w = Button ( master, option=value, ... )
```

PROJECT WORKING

Step 1: Enter the input to check the input is in the valid format or not. Or to slice the address.



OUTPUT

OUTPUT:- If the address is valid then the output shown in the following manner.

INVALID FORMAT

The Output shown "INVALID FORMAT" if the format of the input is not supportable.

LEAP IT

A leap year is a calendar year that contains an additional day added to keep the calendar year synchronized with the astronomical year or seasonal year.

"IF STATEMENTS"

Simple if Statements

if statement" is written by using the **if** keyword.

Syntax:

```
if condition :
    indentedStatementBlock
```

Example:

```
a = 33
b = 200

if b > a:
  print("b is greater than a")
```

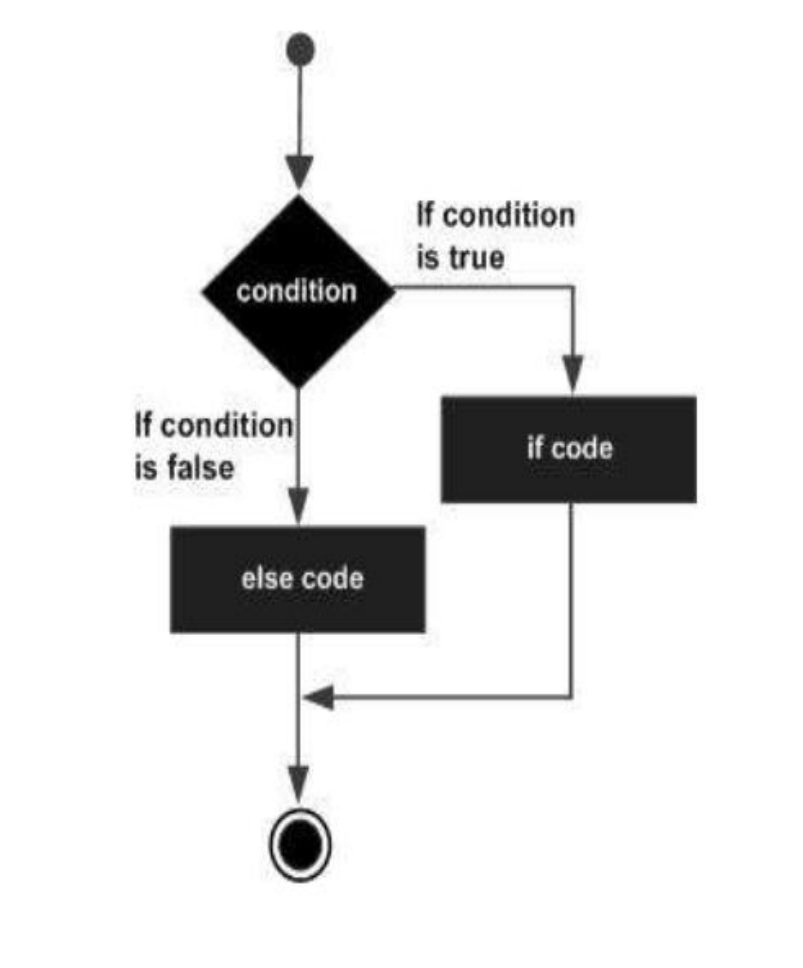
```
b is greater than a
```

IF-ELSE STATEMENTS

Syntax:

```
if expression:
    statement(s)
else:
    statement(s)
```

Flow Diagram:



THE ELIF STATEMENT

The **elif** statement allows you to check multiple expressions for TRUE and execute a block of code as soon as one of the conditions evaluates to TRUE.

Syntax:

```
if expression1:
   statement(s)
elif expression2:
   statement(s)
elif expression3:
   statement(s)
else:
   statement(s)
```

LOGICAL OPERATORS

There are 3 types of operators.

and : True if both the operands are true

or : True if either of the operands is true.

not : True if operand is false(complements the operand)

GUI(TKINTER)

Tkinter is the most common, fast and easy to use Python package to build GUI application.

To install the library, you can use **pip install** command to the **command prompt:**

```
pip install tkinter
```

GEOMETRIC MANAGER

There are mainly three geometry manager classes .

Pack() method
Grid() method
Place() method

WIDGET USED:

- Button

```
w = Button ( master, option=value, ... )
```

- Label

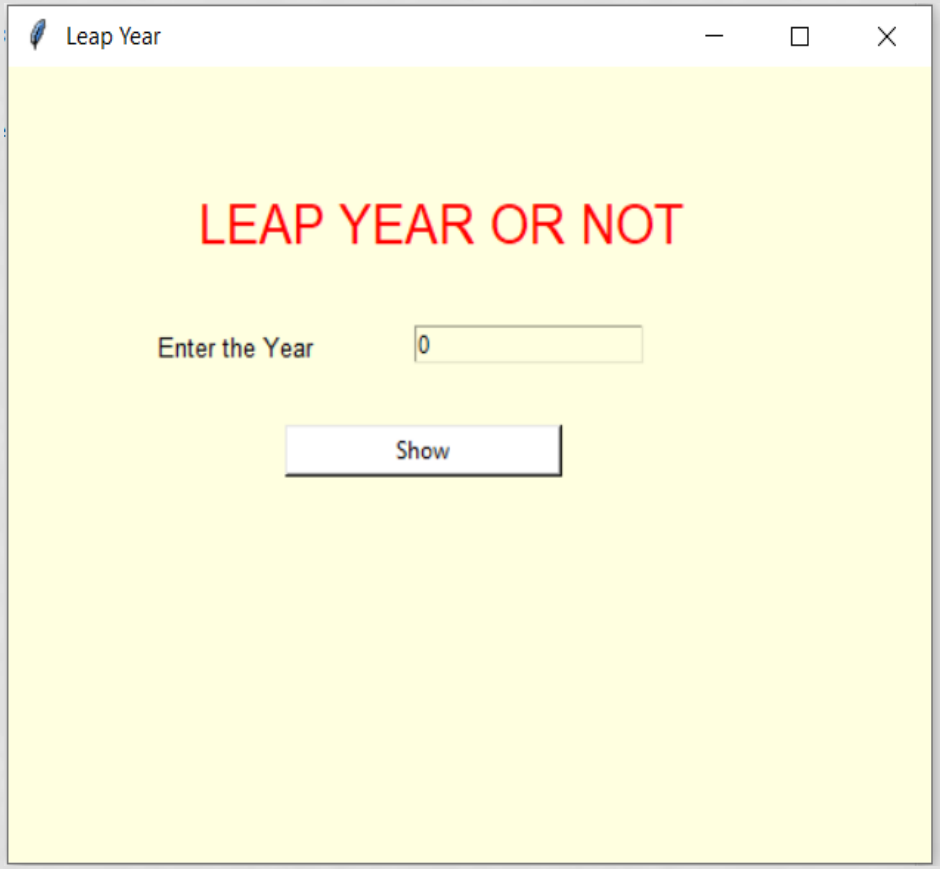
```
w = Label ( master, option, ... )
```

- Entry

```
entry = tk.Entry(parent, options)
```

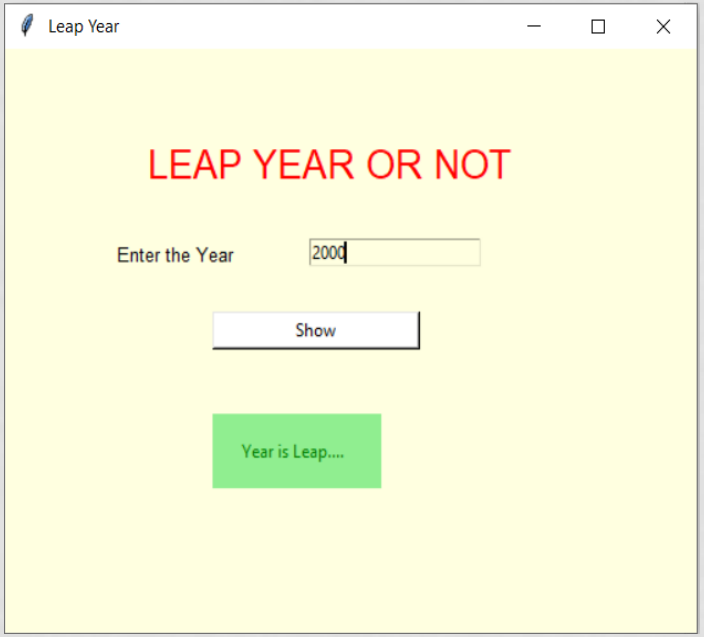
WORKING OF PROJECT

Enter year to check the year is leap or not.

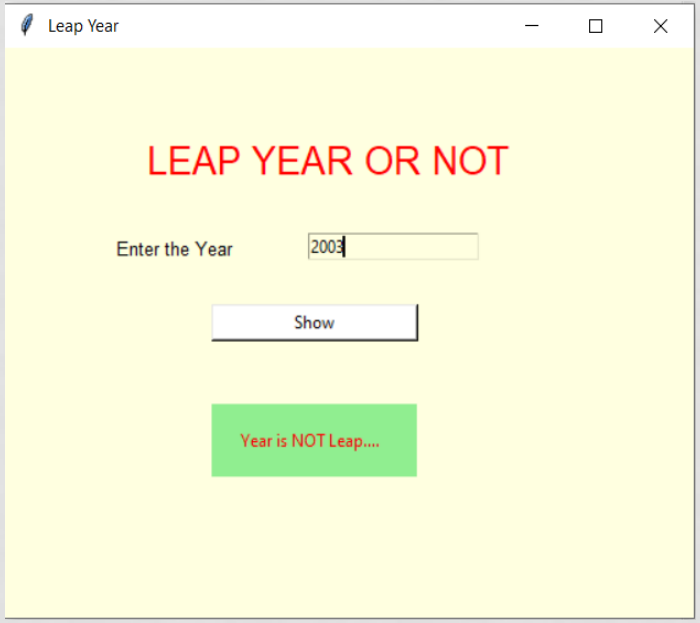


OUTPUT:

Displaying output for yr.2000



Displaying output for yr.2003



NUMBER GUESSING

What is Number Guessing ?

How different to play Number Guessing with your computer ?

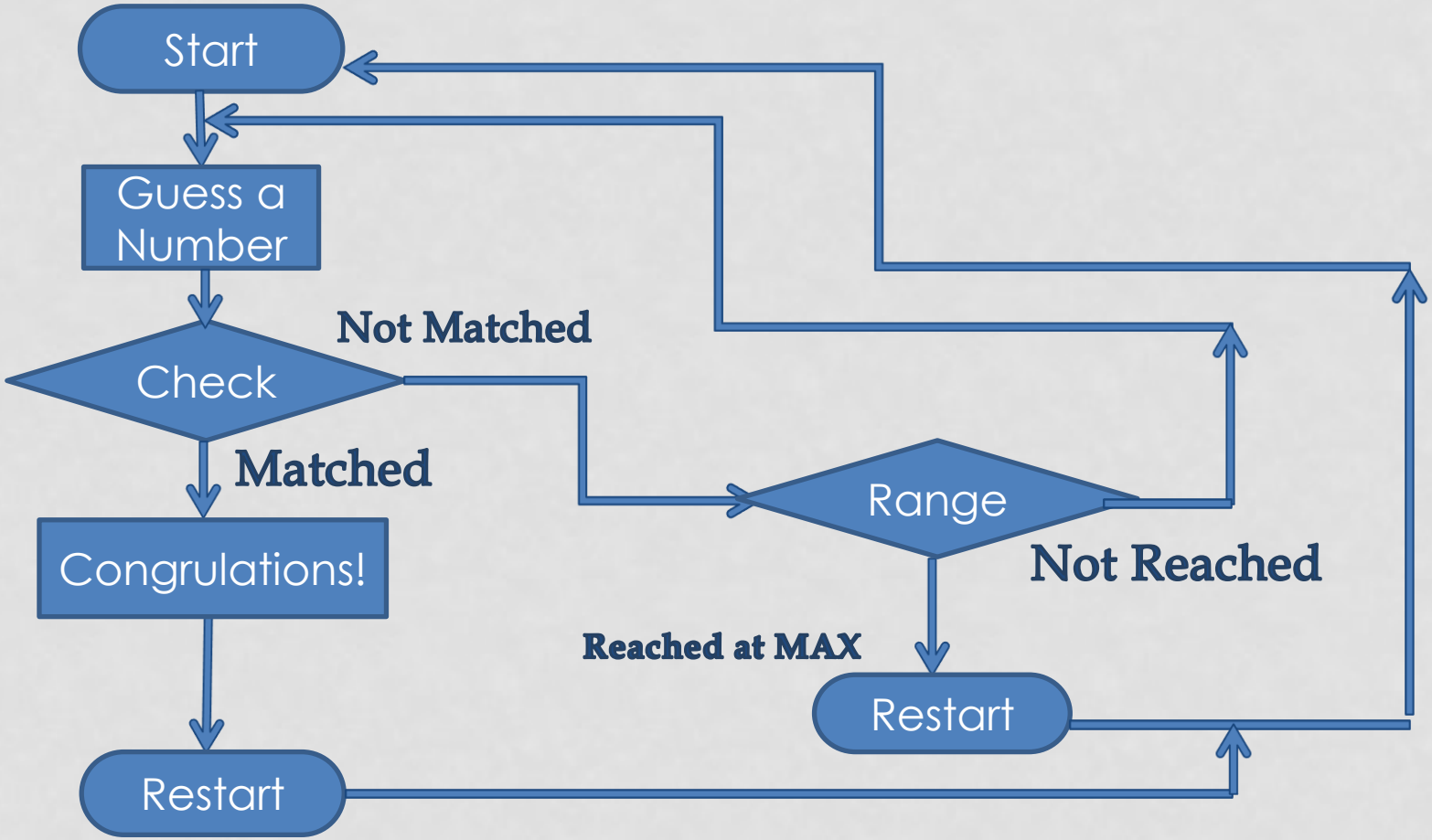
TASK:

Build a Number guessing game, with predefine **range.**

Let's say User select a number, i.e., from **A** to **B**, where **A** and **B** belong to Integer.

Some **random integer will be selected by the system** and the user has to guess that integer in the **minimum number of guesses.**

FLOW CHART:


Start
Guess a Number
Check
Not Matched
Matched
Congrulations!
Restart
Range
Not Reached
Reached at MAX
Restart

GUI(TKINTER)

Tkinter is the most common, fast and easy to use Python package to build GUI application.

To install the library, you can use pip install command to the command prompt:

```
pip install tkinter
```

GEOMETRIC MANAGER

There are mainly three geometry manager classes class.

Pack() method

Grid() method

Place() method

WIDGETS USED:

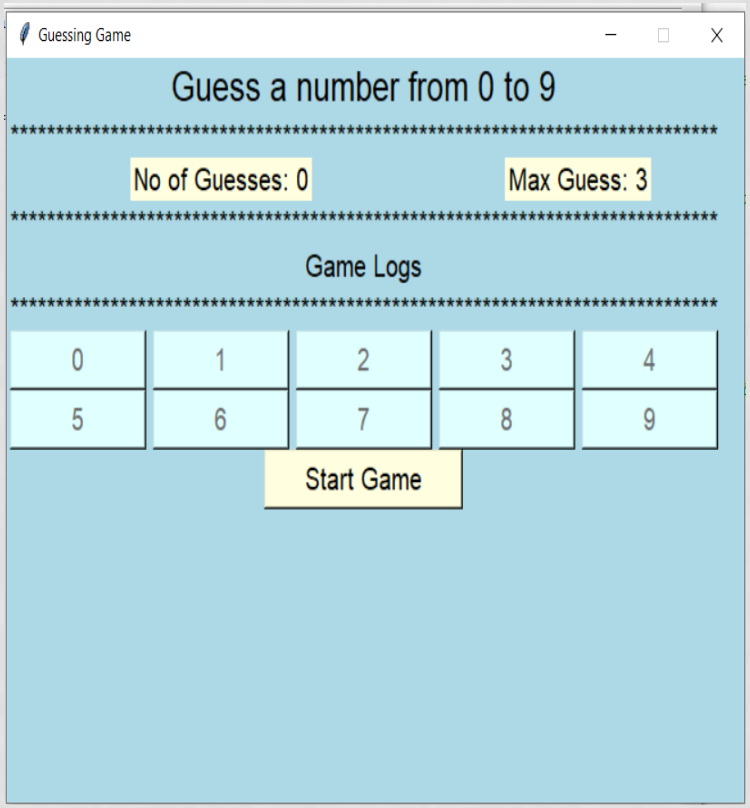
- Button

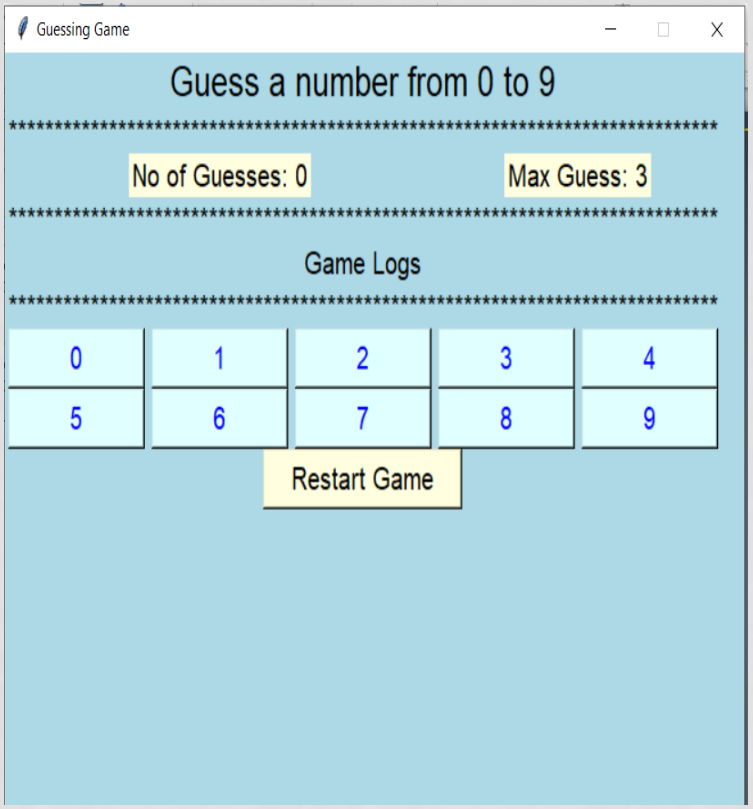
```
w = Button ( master, option=value, ... )
```

- Label

```
w = Label ( master, option, ... )
```

WORKING OF PROJECT

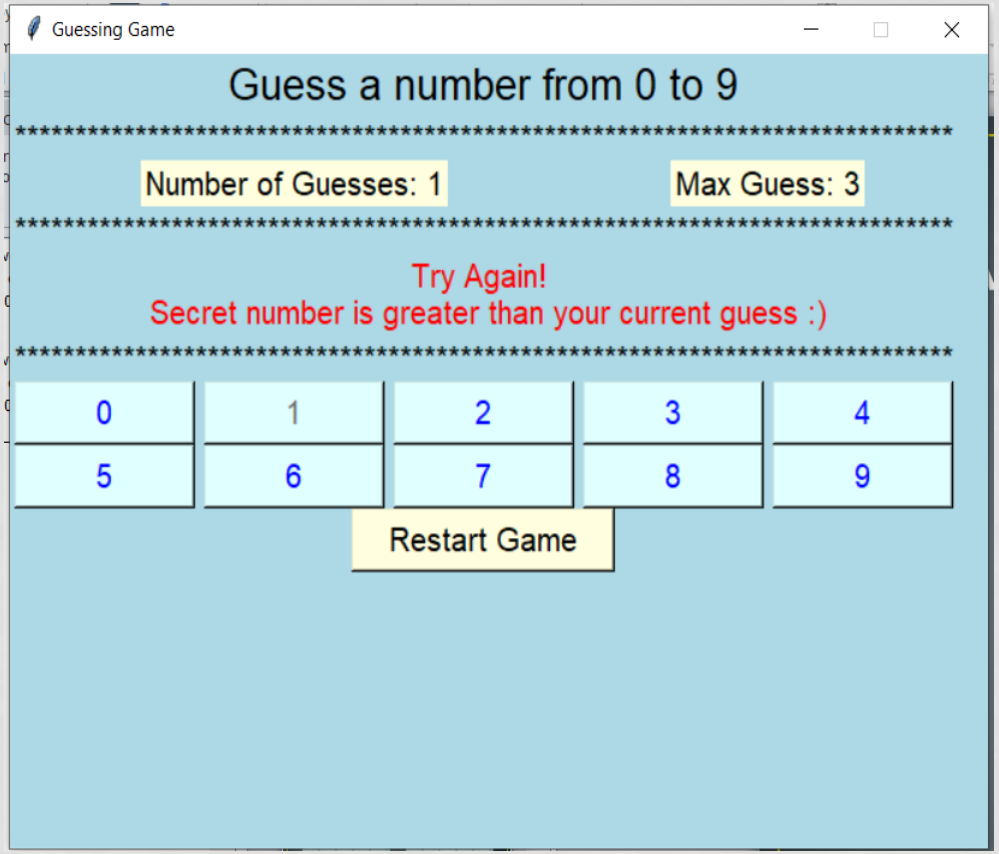




FIRST GUESS:

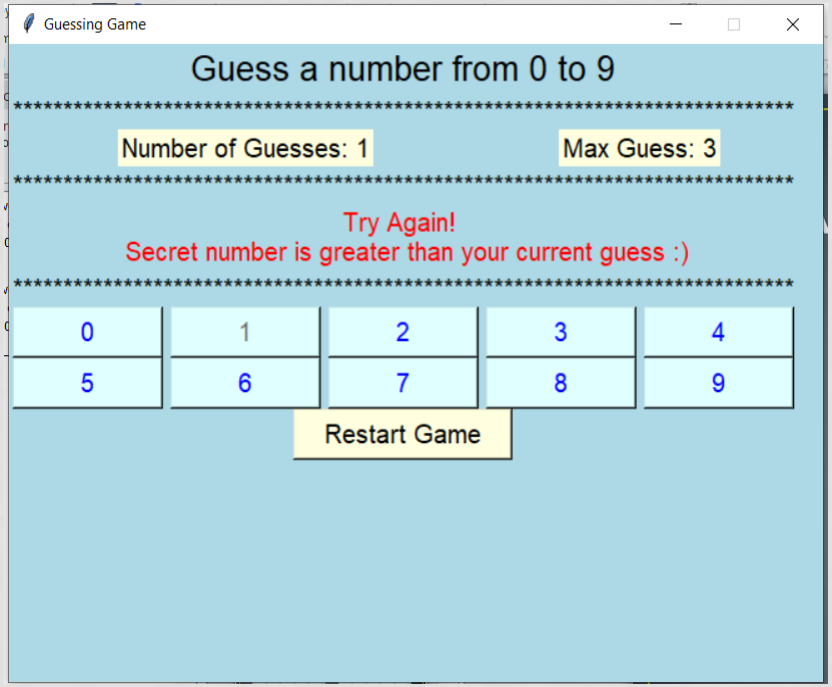
Choose a number for first input.

i.e: user select 1



DISABLE BUTTON:

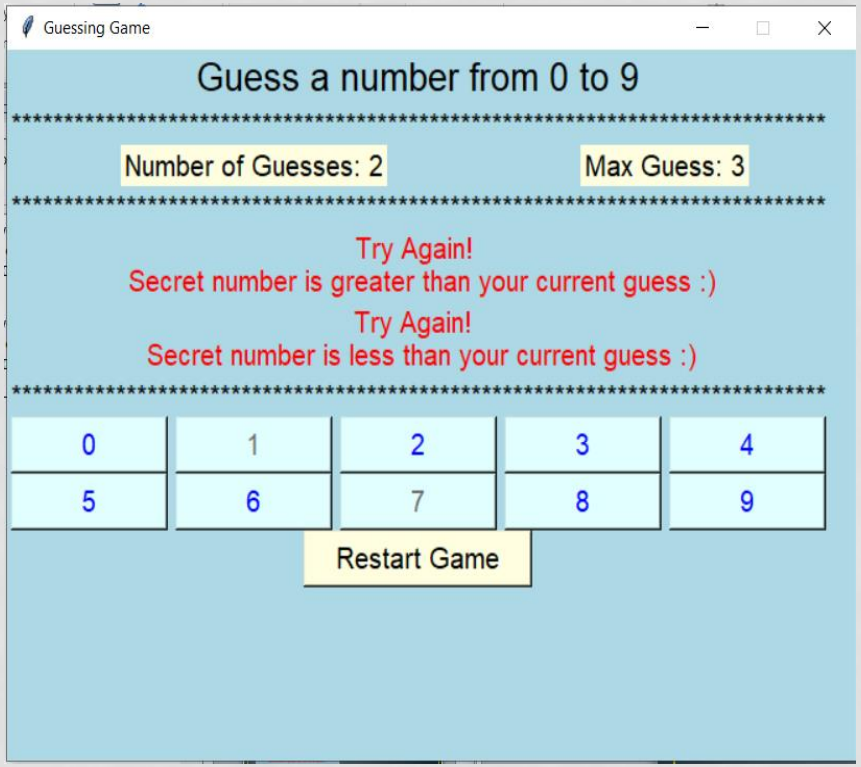
i.e 1 is disable.



SECOND GUESS :

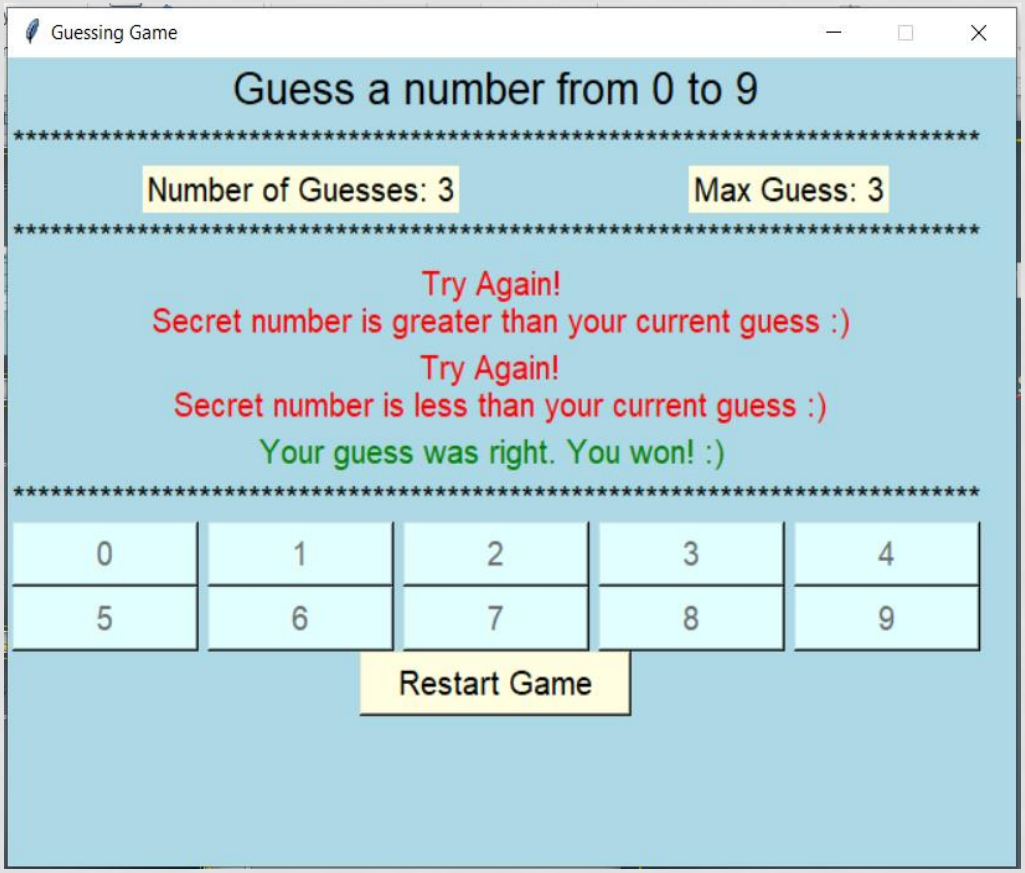
Number of Guesses reached 2.

i.e. user select 7



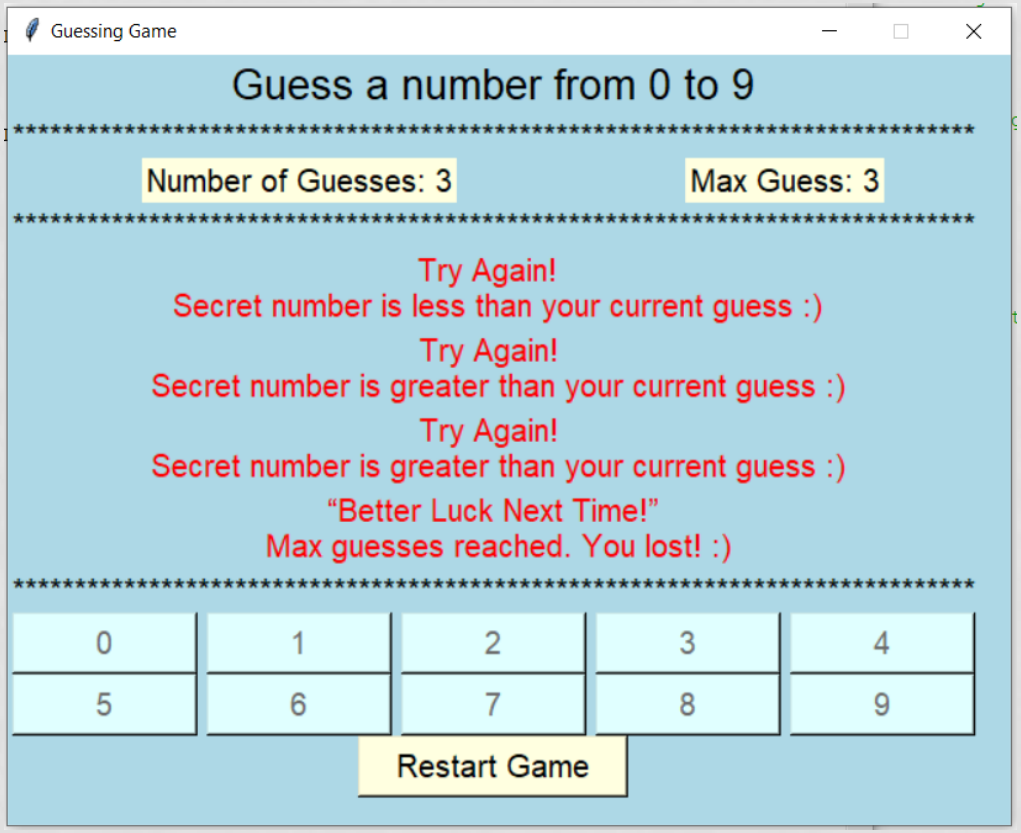
THIRD GUESS :

Number of Guesses reached 3.



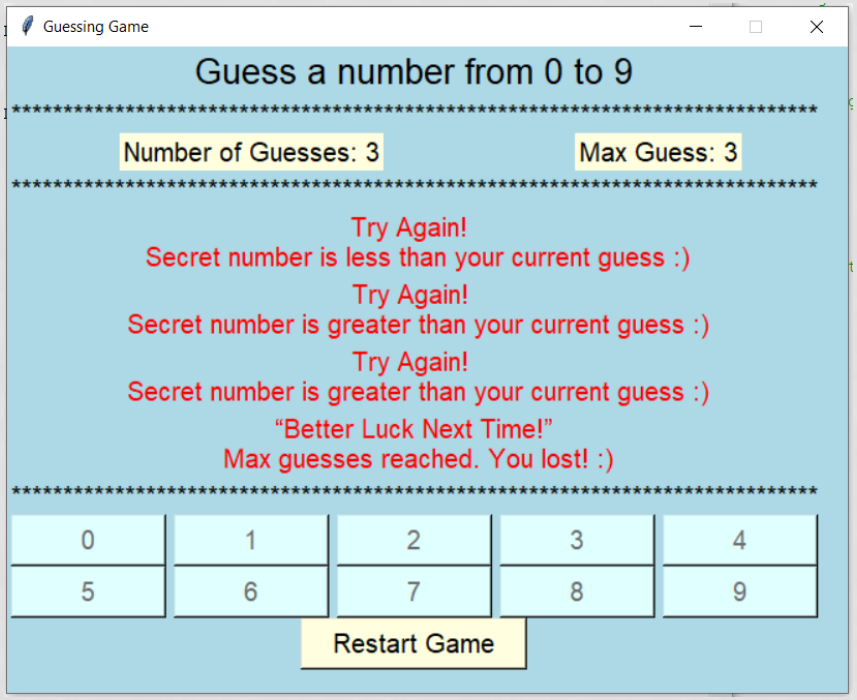
MAX LIMIT REACHED:

At this Stage when user is unable to guess the number generated by the system. and user lost.



RESTART

Max no.of guesses reached



In Between or The user Won

PYTHON WEB BLOCKER

Website Blocker is a tool that denies access to websites permanently or by schedule. To use the internet safely we can block all websites from unwanted categories.

OBJECTIVE

This project will help the user to stay away from their distraction by blocking websites from their device so that they can not open them.

In this Python Website Blocker Project, the user can enter multiple websites to block, and then clicking on the block button will check the condition that if the website already blocked then print ‘already blocked’ else blocked all that websites and print ‘blocked’.

PROJECT PREREQUISITES

To implement website blocker project, we will use the basic concepts of Python and Tkinter library.

Tkinter is a standard GUI Python library. It is one of the fastest and easiest ways to build GUI applications using Tkinter.

To install the library, you can use pip install command to the command prompt:

```
pip install tkinter
```

STEPS TO BUILD WEBSITE BLOCKER PYTHON PROJECT

- Importing the module
- Create the display window
- Create an entry widget
- Define function
- Create a block button

CREATE THE DISPLAY WINDOW

- **Tk()**
- **geometry()**
- **resizable(0,0)**
- **title()**
- **Label()**
- **font**
- **root** – name which we refer to our window
- **text** – which we display on the label
- **pack** – organized widget in block

FILE HANDLING

Data is small then this processing can be done every time you run the script but in case of humongous data repetitive processing cannot be performed, hence the processed data **needs** to be stored. This is where data storage or writing to a **file** comes in.

OPEN A FILE IN PYTHON

Reading and writing file:

We use **open ()** function in Python to open a file in read or write mode.

open ( ) will return a file object.

Syntax:

```
open(filename, mode)
```

ACCESS MODES.

There are Six kinds of mode:

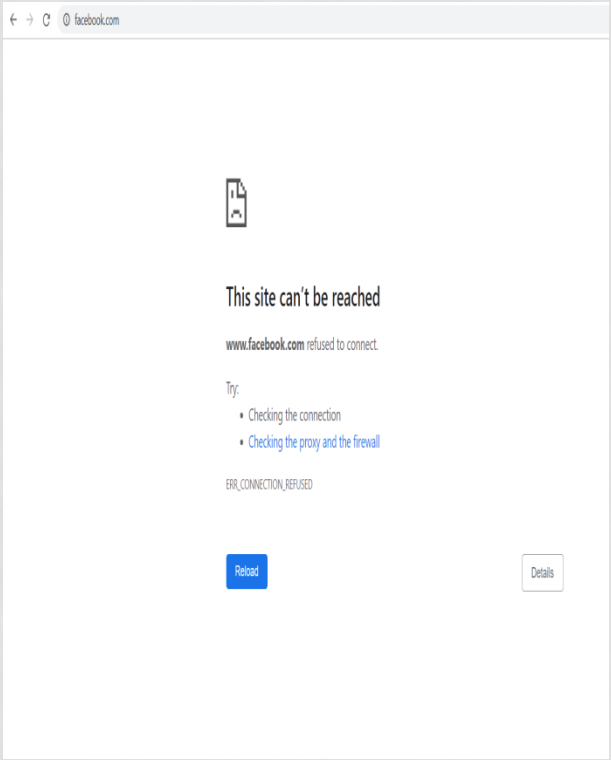
- **Read Only ('r')**
- **Read and Write ('r+')**
- **Write Only ('w')**
- **Write and Read ('w+')**
- **Append Only ('a')**
- **Append and Read ('a+')**

WORKING OF PROJECT

When the user enter the URL.

(i.e www.facebook.com) and on click the block button user blocks the url.

WORKING OF PROJECT

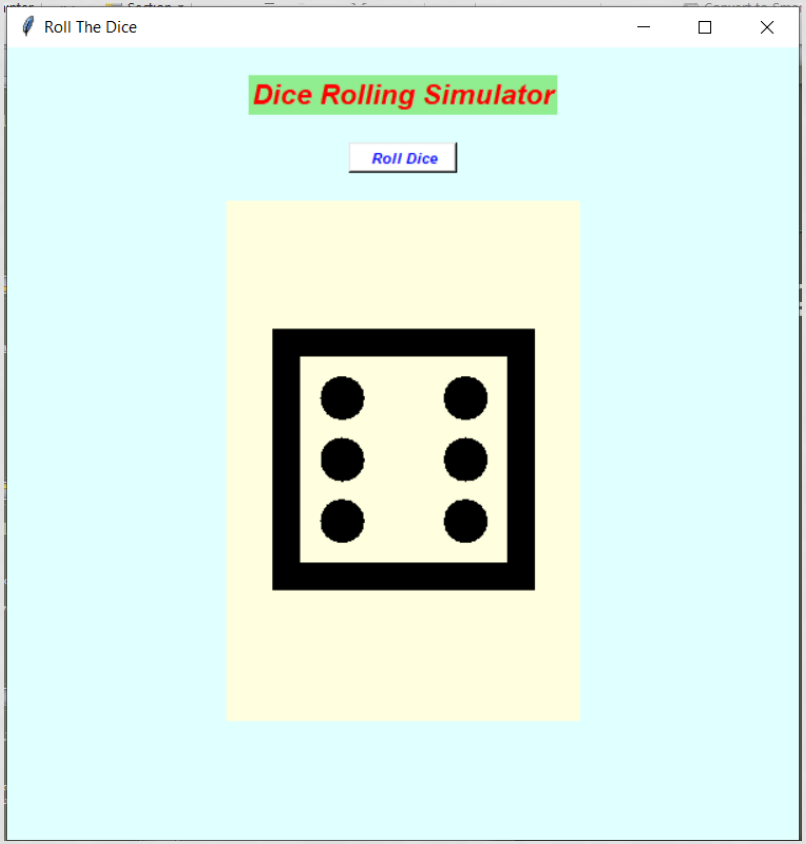


WORKING OF PROJECT

User will not allowed to block already blocked sites.

DICE ROLLING SIMULATOR

DICE ROLLING SIMULATOR???



WHAT IS SIMULATOR???

A simulation is a program or machine that simulates a real life situation, meaning that it creates a virtual version of it.

TYPES OF SIMULATOR.

Live Simulation
Virtual Simulation
Constructive Simulation

LIVE SIMULATION

This simulation involving real people operating real systems.

VIRTUAL SIMULATION

This simulation involving real people operating simulated systems.

This inject Human-In-The-Loop in a central role by exercising.

CONSTRUCTIVE SIMULATION

Real people can simulate(make inputs) but are not involved in determining outcomes.

GUI (GRAPHIC USER INTERFACE)

Tkinter is the most common, fast and easy to use Python package to build GUI application.To install the library, you can use pip install command to the command prompt:

```
pip install tkinter
```

GEOMETRIC MANAGER

There are mainly three geometry manager classes class.

Pack() method

Grid() method

Place() method

WIDGETS IN TKINTER

- Button
- Canvas
- Check Button
- Entry
- Frame
- Label
- List Box

RANDOM MODULE

It is used to generate the pseudo-random variables.

To use random module
we have to import first the module into our file.

```
import random
```

METHODS IN RANDOM MODULE

Randrange()
Randint()
Random()
Uniform()
Choice()
Shuffle()

RANDRANGE()

Always returns an integer value which include
the lower limit and exclude the upper limit.

```
import random
print("OUTPUT")
print("\n")
print("Randrange output",random.randrange(1,9))
```

```
OUTPUT


Randrange output 3
>>>
```

RANDINT()

Always returns an integer value which includes
both lower limit and the upper limit.

```
import random
print("OUTPUT")
print("\n")
print("Randrint output",random.randint(1,9))
```

```
OUTPUT


Randrint output 7
>>>
```

RANDOM()

Always returns floating value between 0 and 1. no parameters requires for this method.

```
import random
print("OUTPUT")
print("\n")
print("Random output",random.random())
```

```
==================== RESTART: C:\Users\hp\
OUTPUT


Random output 0.6043086739934407
>>> |
```

UNIFORM()

Always returns floating value which include the lower limit and exclude the upper limit.

```
import random
print("OUTPUT")
print("\n")
print("Uniform output",random.uniform(1,9))
```

```
OUTPUT


Uniform output 2.083229357474318
>>>
```

CHOICE()

This method is used for random selection from list, tuple, string.

```
import random
listdata=[1,2,3,4,5]
tupledata=(1,2,3,4)
stringdata ="SIMRAN"
print("OUTPUT")
print("\n")
print("Choice method (LIST) output",random.choice(listdata))
print("\n")
print("Choice method (TUPLE) output",random.choice(tupledata))
print("\n")
print("Choice method (STRING) output",random.choice(stringdata))
```

```
OUTPUT


Choice method (LIST) output 5


Choice method (TUPLE) output 2


Choice method (STRING) output R
>>>
```

SHUFFLE()

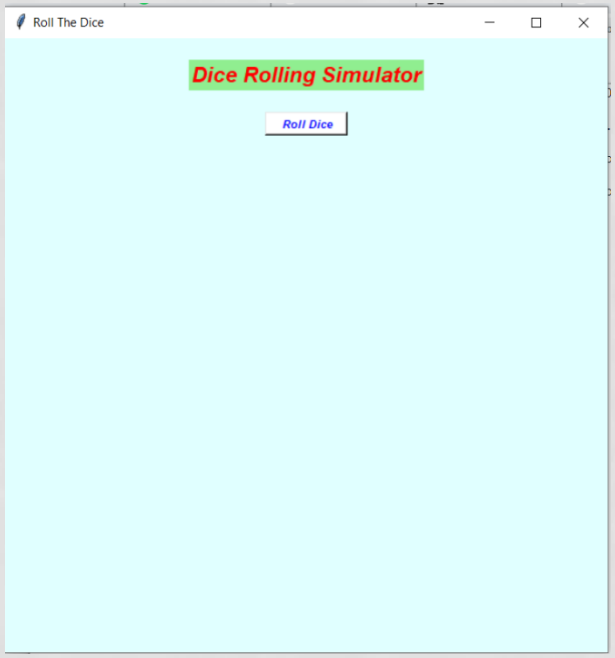
This method shuffle the items of a given list. (change the order of the list item)

```
import random
data = [1,2,3,4,5,8]

print("OUTPUT")
print("\n")
print("Before shuffle method",data)
random.shuffle(data)
print("After shuffle method",data)
```

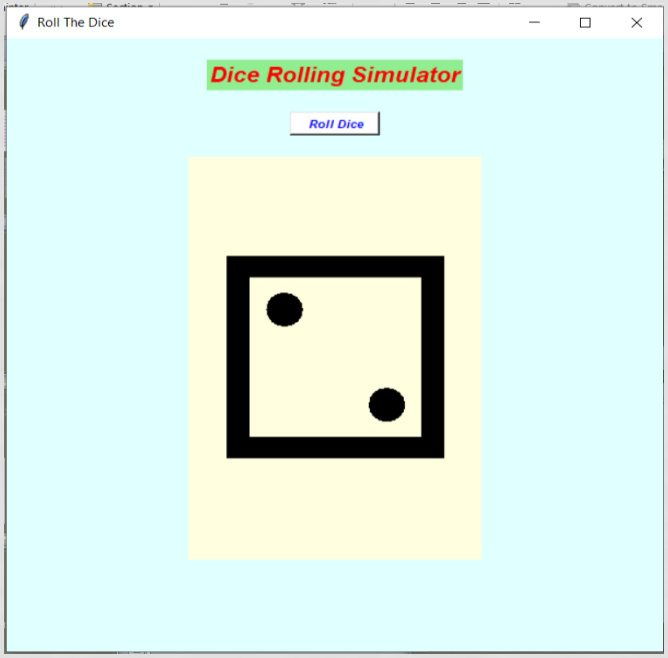
```
OUTPUT


Before shuffle method [1, 2, 3, 4, 5, 8]
After shuffle method [8, 4, 2, 1, 3, 5]
>>>
```

WORKING OF SIMULATOR

Initial stage



Functional stage



THANKYOU…