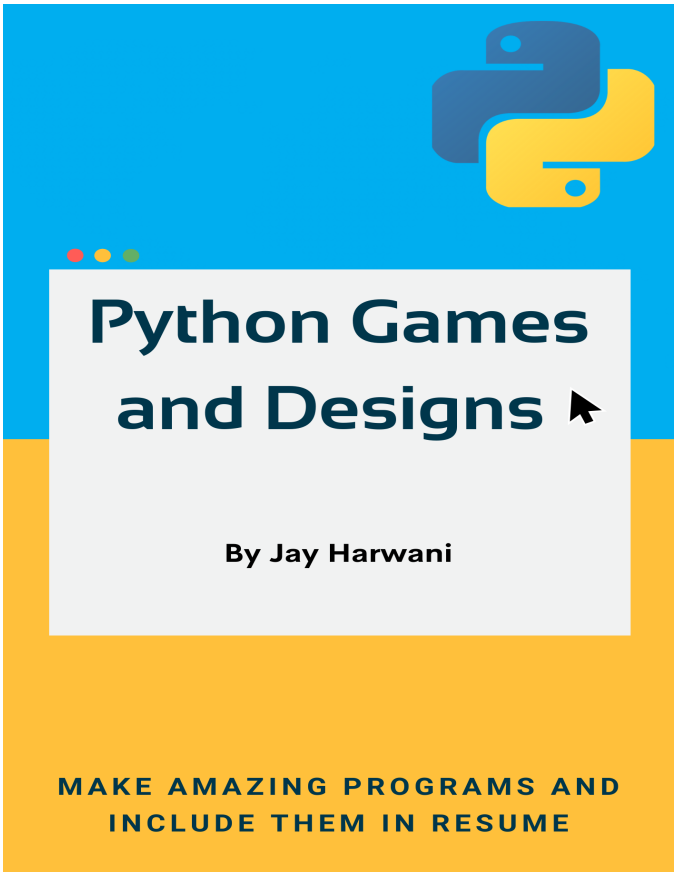

Python Games
and Designs
By Jay Harwani
MAKE AMAZING PROGRAMS AND
INCLUDE THEM IN RESUME

Introduction 1

This small book is going to give you some of the common and also never heard projects of python programming language. I have mentioned in terms and conditions also that this ebook is for those only who have basic knowledge and cleared fundamentals of Python.
You should have downloaded python from its official website and a text editor or IDE, detailed information about IDE's of python is given further in the book.

With a small explanation, SOURCE CODE and screenshots of every program are included. You may find difficulty in this book as programs are long and somewhat tough to understand. The book is devoted to talking about the popular projects of different categories.

By Jay Harwani

Table of Contents

2

Best Text Editors for Python :-

3

1.Visual Studio Code(VS Code) - My favorite editor is vs code which I use most of the time. The best thing about this editor is that it has everything which and programmer expects from any code editor. It is fast, lightweight and it's powerful too! designed by Microsoft. Can be used in Windows, Linux, and macOS.

2.Sublime Text editor - It is a sophisticated text editor that is widely used among developers. It includes wide features such as Syntax Highlight, Macros, Sidebar, and File Type Recognition. It is fully customizable.

3.PyCharm - It is one of the most widely used IDEs for Python programming language. This editor helps programmers to write a variety of software applications in Python quickly and efficiently. Pycharm is the best IDE for me to meet complex project requirements. It further helps programmers to use python more efficiently in BIG DATA and DATA SCIENCE.

4.Atom - Atom is an open-source text editor made by GitHub that lets easily customize every aspect of it. Smart autocompletion, multiple panes, best themes are my chosen features of this editor. This was my first editor when I started learning C and C++.

5.Jupyter Notebook - Also known as IPython is an enhanced Python Interpreter. Possesses the ability to check the properties of an object during runtime. Magic command system for use for controlling the python environment and performing OS tasks.The ability to code from a browser.

Although there are some other IDE's like Vim, Emacs, Bluefish, Thonny and so on which you can use, these above editors were my best recommendations.

HERE WE BEGIN PEEPS!

4

SECTION-1

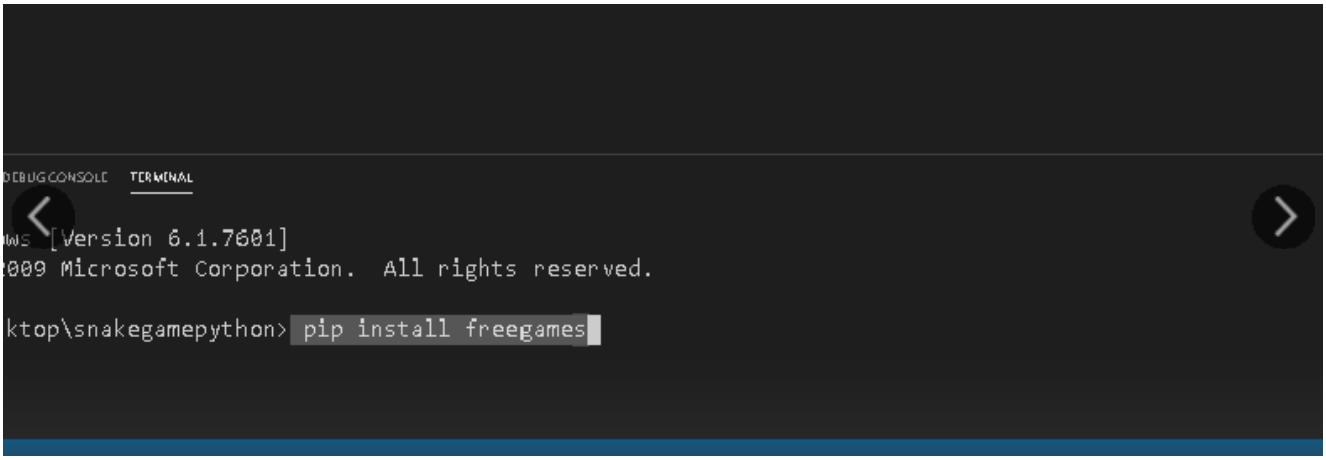
SNAKE GAME

Yes! , you have played this in your first cell phone and were addicted to it, but this time we are playing this time in our python project.
Firstly, you have to install some modules to run this program, here I will not explain what are modules or how to install them. I am using VS code so I will tell you how to install modules in VS code

Modules to be Installed:

1. Install Turtle

2. Install free games



As you can see in this image, if you are using vs code you have to type 'pip install (module name) as shown in highlighted in the terminal. In the game, the player is represented as *a snake,* which grows if it eats food.

The goal of the game is to eat food as much as possible without colliding into yourself or with the walls.

Game logic

1. If the snake eats the food, the food moves to a new position.

2. If the snake eats food, the length of the snake grows.

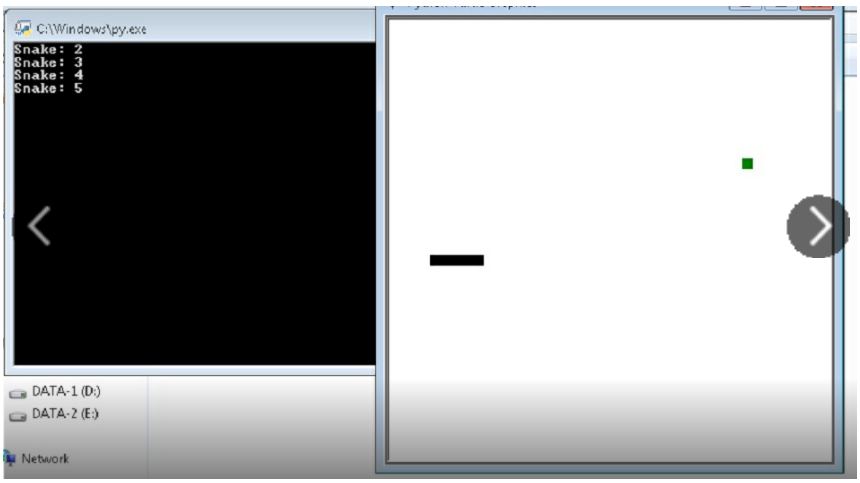
3. If the snake collapse, then the game is over.

Source Code:

```
from turtle import *
from random import randrange
from freegames import square,vector
food = vector(0,0)
snake = [vector(10,0)]
aim = vector(0,-10)
def change(x,y):
    "Change snake direction."
    aim.x = x
    aim.y = y
def inside(head):
    "Return True if head inside boundaries."
    return -200<head.x<190 and -200<head.y<190
def move():
    "Move snake forward one segment."
    head = snake[-1].copy()
    head.move(aim)
    if not inside(head) or head in snake:
        square(head.x,head.y,9,'red')
        update()
        return
    snake.append(head)
    if head == food:
        print('Snake:',len(snake))
        food.x = randrange(-15,15)*10
        food.y = randrange(-15,15)*10
    else:
        snake.pop(0)
    clear()
    for body in snake:
        square(body.x,body.y,9,'black')
    square(food.x,food.y,9,'green')
    update()
```

```
    ontimer(move,100)
setup(420,420,370,0)
hideturtle()
tracer(False)
listen()
onkey(lambda: change(10,0), 'Right')
onkey(lambda: change(-10,0), 'Left')
onkey(lambda: change(0,10), 'Up')
onkey(lambda: change(0,-10), 'Down')
move()
done()
```

This is the source code for the snake game which you can copy-paste and run the program or if you have python knowledge you can just take a look and understand the code and its logic. If you have noticed the entire program there has been the use of maths.



This will be the output of the file

If your file runs successfully without any errors, then CONGRATS! You have created your project. Press control+s to save the file.
Delete the terminal after running the code if you want to run the code again because python is dynamically slow and takes time to run the code, if you run the program again without deleting the terminal then the might program will not run.

Cannon Game

This game is quite different and simple. Most of you have played this game but this is in a very simple format where a ball hits objects which are moving and when balls hit objects they get vanished and you get points, but in this cannon game there is no concept of points.

Game logic:

1.You have to press the left key to hit the objects.

2.Timing is important to hit the objects, if you launch the ball at the proper time more objects get hit and vanished.

3.I have intentionally made this game quite a bit tougher for the learners.

Modules to be installed:

1.Install free games

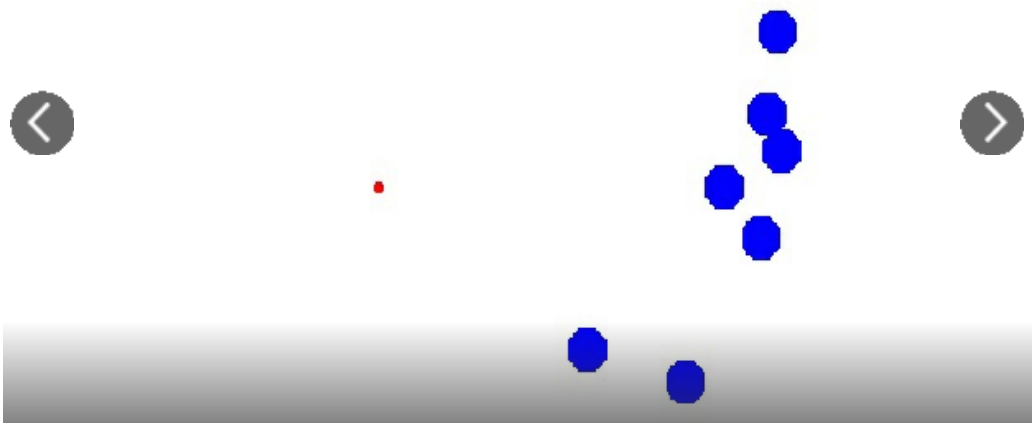
2.Install turtle

The modules to be installed are as same as the first game, so you don't need to download these modules again.

Source Code :-

```
from random import randrange
from turtle import *
from freegames import vector

ball = vector(-200, -200)
speed = vector(0, 0)
targets = []

def tap(x, y):
    "Respond to screen tap."
    if not inside(ball):
        ball.x = -199
        ball.y = -199
        speed.x = (x + 200) / 25
        speed.y = (y + 200) / 25
```

```
def inside(xy):
    "Return True if xy within screen."
    return -200 < xy.x < 200 and -200 < xy.y < 200

def draw():
    "Draw ball and targets."
    clear()

    for target in targets:
        goto(target.x, target.y)
        dot(20, 'blue')

    if inside(ball):
        goto(ball.x, ball.y)
        dot(6, 'red')

    update()

def move():
    "Move ball and targets."
    if randrange(40) == 0:
        y = randrange(-150, 150)
        target = vector(200, y)
        targets.append(target)

    for target in targets:
        target.x -= 0.5

    if inside(ball):
        speed.y -= 0.35
        ball.move(speed)

    dupe = targets.copy()
    targets.clear()

    for target in dupe:
        if abs(target - ball) > 13:
            targets.append(target)

    draw()

    for target in targets:
        if not inside(target):
            return

    ontimer(move, 50)

setup(420, 420, 370, 0)
hideturtle()
up()
tracer(False)
onscreenclick(tap)
move()
done()
```



This will be output!

I have kept the color of objects as blue and of the ball is red, you can keep your own color as you like and you can also change the size of the ball and objects according to your own comfort of playing the game.

Joke Game

Actually, this is not a game though I have kept this project in the games section. This Joke Game is an intermediate project, this program on running generates random jokes.

Game Logic:-

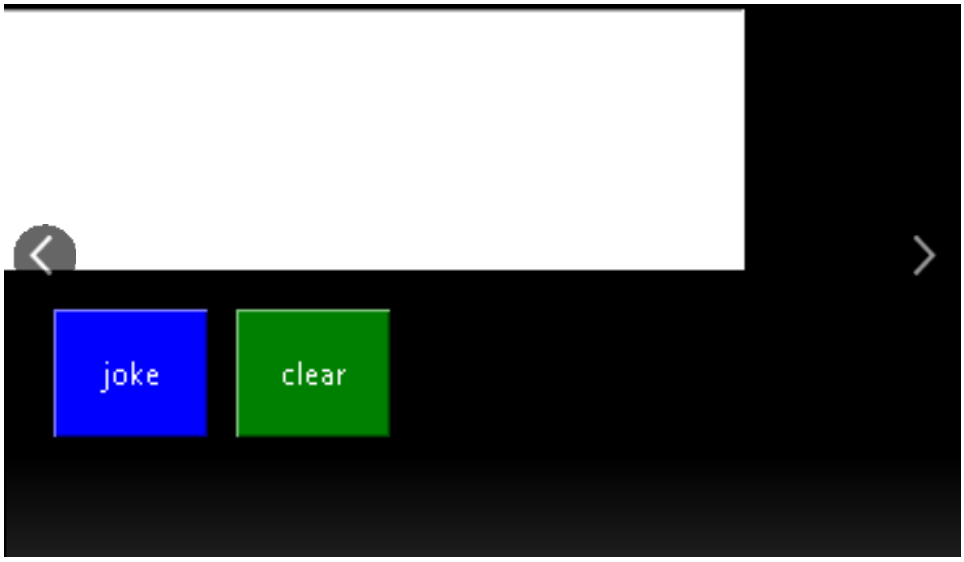
1.There will be 2 buttons namely 'joke' and 'clear' for creating jokes and clearing it.

2.Jokes will be one-liner not the large ones with paragraphs.

3.The output shown after running the code are in module pyjokes these are not mine.

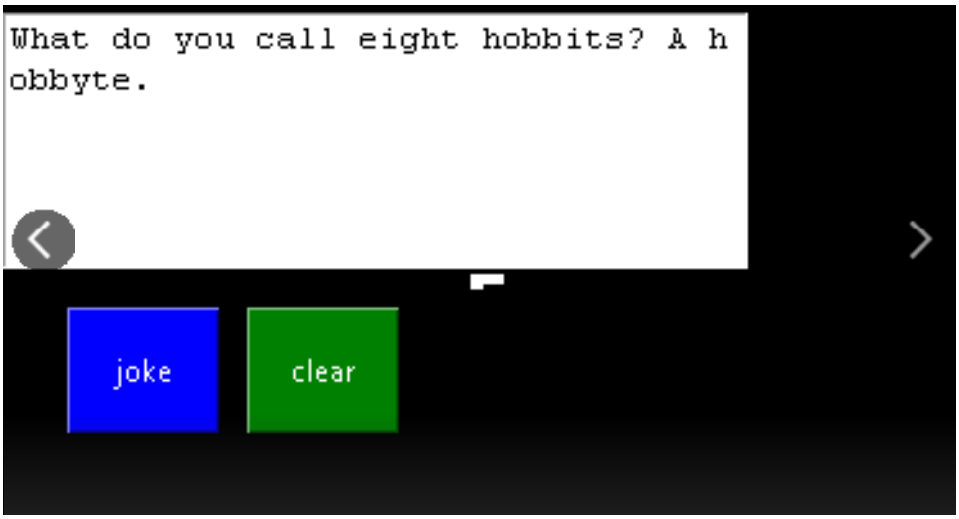
Source Code:

```
import tkinter as tk
import pyjokes
root=tk.Tk()
root.title("jokes game")
root.geometry("300x300")
root.configure(bg="black")

def joke():
    global joke
    joke = pyjokes.get_joke()
    T.insert(tk.END,joke)


def clear():
    T.delete("1.0","end")


T = tk.Text(root)
T.place(x=5,y=5,height=100,width=290)
b1 = tk.Button(root,text = "joke",bg = "blue",fg = "white",command=joke)
b1.place(x=30,y=120,height=50,width=60)
b1 = tk.Button(root,text = "clear",bg = "green",fg = "white",command=clear)
b1.place(x=100,y=120,height=50,width=60)
root.mainloop()
```



This will be the output after running the code!.As i told earlier you can see the two buttons for showing the result and clearing it.

You can set your color of buttons and length of it according to your own comfort.



After clicking the button 'joke' this will be like the result and to if you want another one just hit the clear button and again click on the joke. THAT'S SIMPLE!
I hope you enjoyed this program as it is somewhat different from others.

Rock,Paper,Scissors Game

Experienced programmers will be very quick to point out that there are better ways to write a "Rock, Paper, Scissors" Python game. But for a beginner, it's very important to be able to understand and follow the program. Let's break it out down.

This is a very basic game so here we doesn't need any modules to be installed.

We just need **random and randint** which are already installed in the IDE.

Game logic:

1.We assign a random play to the program namely rock, paper, and scissors.
2.The output will be a question as you know the computer asking us to choose one from three of them and the computer chooses its.

Source Code:

```
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)]
```

Pattern:-

1.User = "Rock" and Comp = "Paper" - Comp wins

2.User = "Rock" and Comp = "Scissor" - User wins

3.User = "Rock" and Comp = "Rock" - Draw

4.User = "Paper" and Comp = "Paper" - Draw

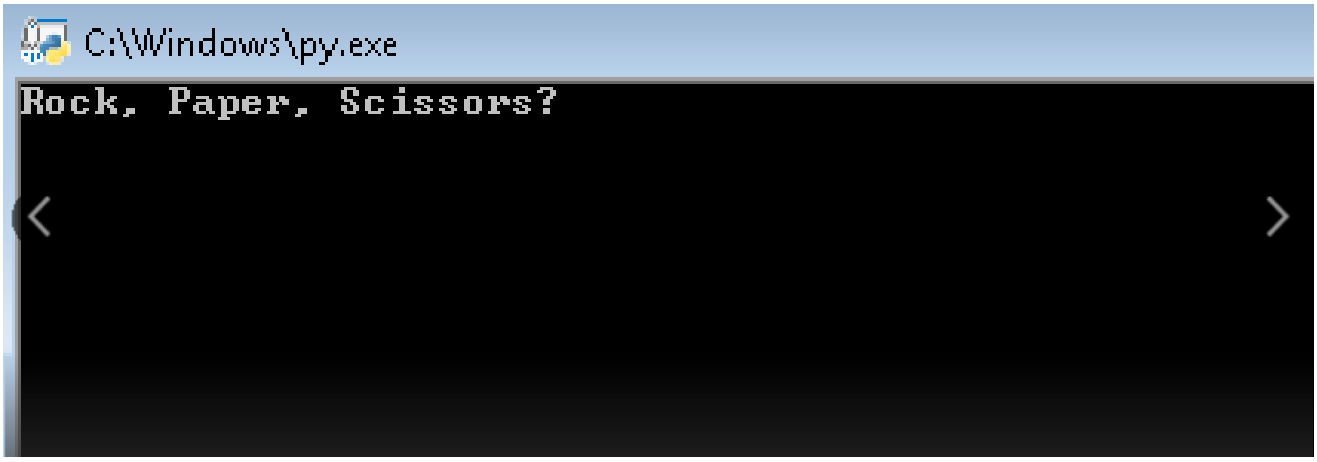
5.User = "Paper" and Comp = "Scissor" - Comp wins

6.User = "Paper" and Comp = "Rock" - User wins

7.User = "Scissor" and Comp = "Paper" - User wins

8.User = "Scissor" and Comp = "Scissor" - Draw

9.User = "Scissor" and Comp = "Rock" - Comp wins

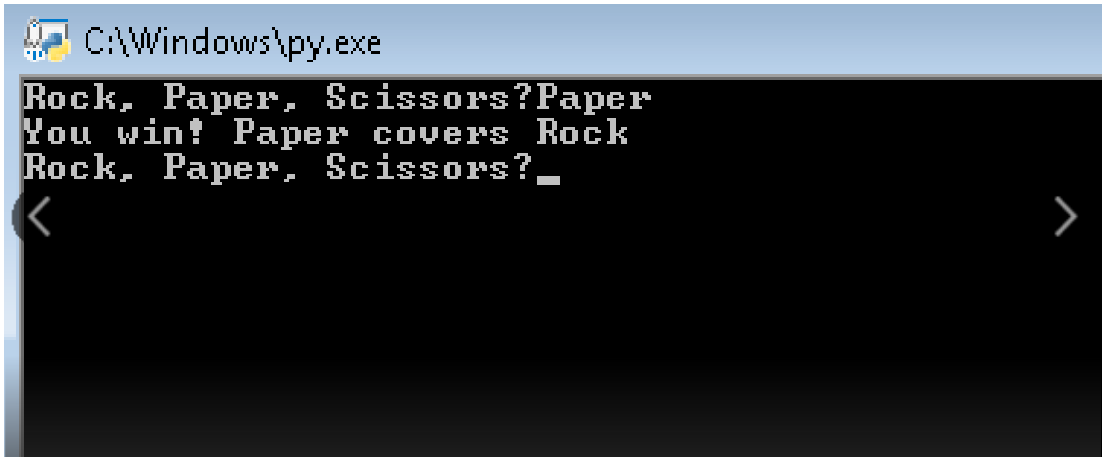


When you run the program the output will be like this means the computer is asking you to choose any one of them.

Suppose I chose 'paper' if a computer chooses 'scissor' then I will lose but if the

computer chooses 'rock'

then I will win the game according to the rules or patterns mentioned above



The Output!

As you can see the example in screenshot,this will be result of the code if you program correctly.

SECTION-2

Amazing Designs

1.Rangoli design

Well, this Rangoli design is a project for the intermediate who have cleared the concepts like length, vectors, and arguments in Python and many more because this is nearly a program of 100 lines of code so this is not for beginners.

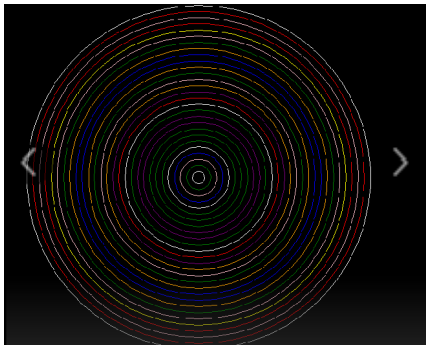
We all have made rangoli in our home during the festival of Diwali, so let's try this thing in Python programming EXCITED to learn?? Me too, to explain this project.

No modules are needed to be installed, we just need a turtle and random module which are installed by default.

Source Code:

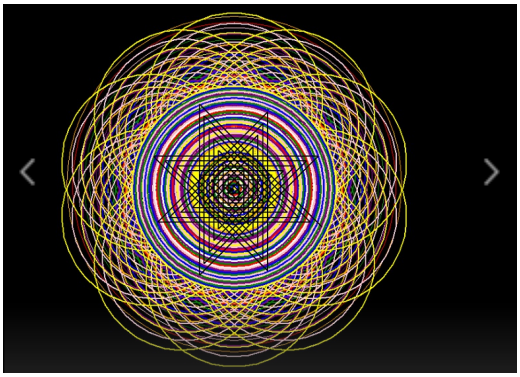
```
import turtle
import random

##################################### Global Variables ###########################################################
#distinct_colors
colors = ["red","green","blue","orange","purple","pink","yellow","dark green","dark red","lime","dark blue","medium violet red",
        "cyan","saddle brown","dark gray","dark orange","medium purple","magenta"]
#dark_colors
dark_colors = ["white","red","green","blue","orange","purple","pink","yellow"]

#line_length
length = 5

##################################################################################################################


##################################### Function Definitions ########################################################
def remote_tangent_circles(circle_turtle,color,dis_range,radius):
    """ Function to draw tangent Circles
    Parameters:
    arg1 (turtle class): Turtle Class Refernce
    arg2 (int) : color
    arg2 (int) : Distance Range
    arg3 (int) : Circle Radius
    Returns:
    None:Returning None
    """
    circle_turtle.color(color)
    for i in range(dis_range):
        circle_turtle.circle(radius*i)

    circle_turtle.up()
    circle_turtle.goto(0,0)
    circle_turtle.down()

def remote_concentric_circles(circle_turtle,dis_range,radius):
    """ Function to draw Concentric Circles
    Parameters:
    arg1 (turtle class): Turtle Class Refernce
    arg2 (int) : Distance Range
    arg3 (int) : Circle Radius
    Returns:
    None:Returning None
    """
    for i in range(dis_range):
        color = random.choice(dark_colors)
        circle_turtle.color(color)
        circle_turtle.circle(radius*i)
        circle_turtle.up()
        circle_turtle.sety((radius*i)*(-1))
```

```
        circle_turtle.down()
    circle_turtle.up()
    circle_turtle.goto(0,0)
    circle_turtle.down()

if __name__ == "__main__":
    remote_circle = turtle.Turtle()
    remote_circle_screen = turtle.Screen()
    remote_circle_screen.bgcolor('black')

    remote_circle.width(1) #set the circle width
    remote_circle.speed(0) #set speed delay to 0

    remote_concentric_circles(remote_circle,30,10)
    remote_circle.width(2) #change the circle width
    for j in range(8):
        for i in range (10):
            remote_tangent_circles(remote_circle,dark_colors[j],10,(10 + j))
            remote_circle.right(360/10)
    remote_circle.width(3)
    remote_concentric_circles(remote_circle,60,3)
    remote_circle.width(2)
    #start drawing the lines
    for count in range(60):
        remote_circle.forward(length)
        remote_circle.right(135)
        remote_circle.color('black') #change the color
        length = length + 5
    #Reset the Original Position
    remote_circle.penup()
    remote_circle.home()
    remote_circle.pendown()
    #exit the turtle class on click()
    turtle.Screen().exitonclick()
```



The output will start like this if you run the code properly!

The Interpreter will nearly take 50 seconds to 1 minute to complete the whole design, if you use VS Code then the turtle will draw faster because VS Code is faster than other IDE's.



This will be the final output!!!

Amazing na?? so is the title of the section.

9

Colorful Stars

So, if you have a clear idea of the fundamentals of Python then you can draw a star. But in this program, you can draw many stars till you make a design with a different color of stars with very little coding.

If you know how to use loops and angles with coordinates then you can easily understand this project

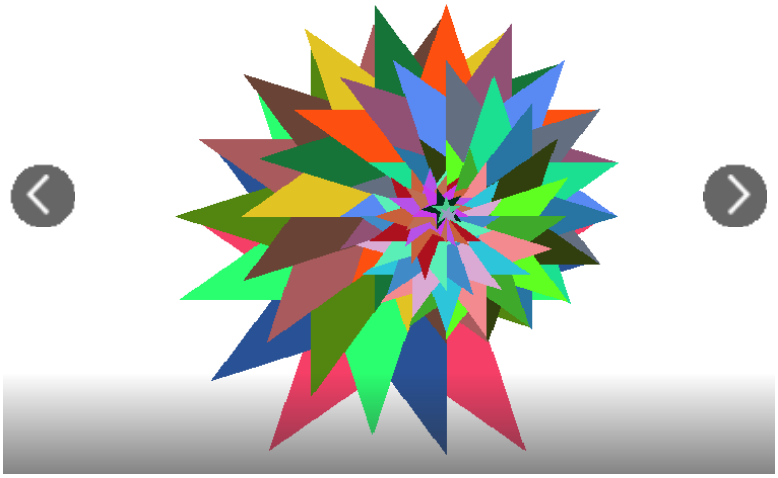
as I mentioned in t&c I will not explain any concept of python programming you should have basic knowledge of it.

Same libraries 'turtle' and 'random' are used in this program.

Source Code:

```
from turtle import *
import random
speed(speed = 'fastest')
def draw(n,x,angle):
    #23 loop for numbers of stars
    for i in range(n):
        colormode(255)
        #23 choosing random integers
        #23 between 0 and 255
        #23 to generate random rgb values
        a = random.randint(0,255)
        b = random.randint(0,255)
        c = random.randint(0,255)
        #23 setting the outline
        #23 and fillcolor
        pencolor(a,b,c)
        fillcolor(a,b,c)
        begin_fill()
        for j in range(5):
            forward(5 * n - 5 * i)
            right(x)
            forward(5 * n - 5 * i)
            right(72 - x)
```

```
            end_fill()
            rt(angle)
n = 30
x = 144
angle = 18
draw(n,x, angle)
turtle.done()
```



This will be the output!.....Give a feedback if you like it :)

This is the easiest program for beginners and intermediates too.

COLORFUL INFINITE HEXAGON

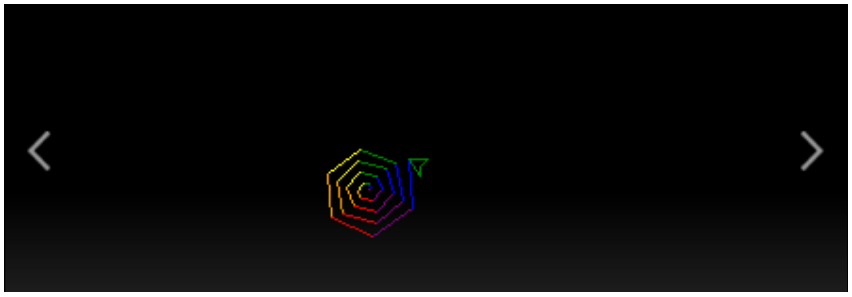
This is the most popular design and program in Python. The compiler takes time to finish the program as it runs very slow because the design is quite large, but for less time to run, I would again recommend using VScode.

It is the smallest and easiest program in this my ebook with just 8 lines of code, so it will be a piece of cake for beginners and intermediates too.

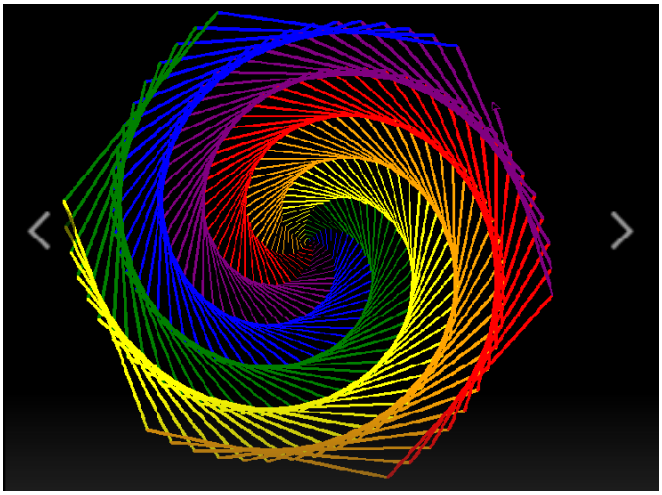
The turtle module is only required as we used also in all previous programs.

Source Code:

```
import turtle
colors = ['red','purple','blue','green','yellow','orange']
turtle.bgcolor('black')
for x in range(360):
    turtle.pencolor(colors[x%6])
    turtle.width(x/100+1)
    turtle.forward(x)
    turtle.left(59)
```



Code will start running like this and it will take some time to complete as python is dynamically slow.
Whenever I run this code it nearly takes 2 min 10 seconds to complete the whole design in VS code. Tell me how much time it took in your IDE which you are using.



The Output!

11

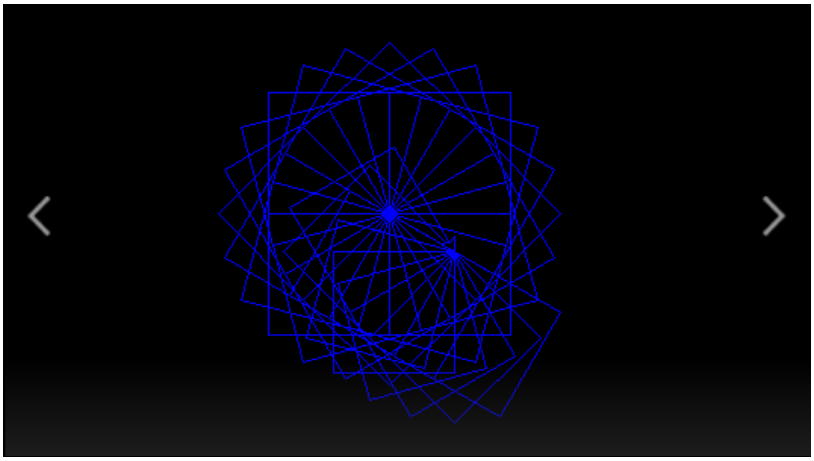
Blue Rangoli with Squares

This is my favorite design! And because I have chosen a blue color in the program so it looks beautiful in output, you can choose the color according to your comfort.
This is also a very small and quite simple program yet with amazing output.

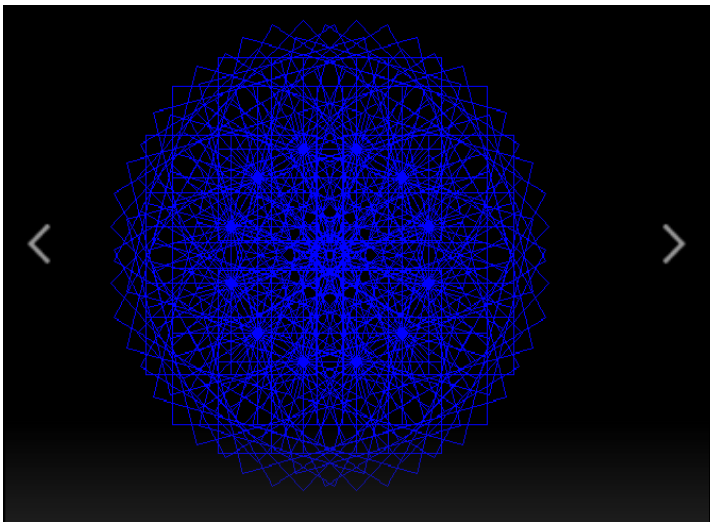
Only the Turtle module will be used which is very important in python if you are drawing shapes, designs, or any other block of code related to colors and shapes.

Source Code:

```
import turtle
t = turtle.Turtle()
s = turtle.Screen()
s.bgcolor('black')
t.pencolor('blue')
t.speed(0)
c = 0
d = 0
while True:
    for i in range(4):
        t.forward(80)
        t.right(90)
    t.right(15)
    c +=1
    if c >= 390/15:
        t.forward(50)
        c = 0
        d += 1
        if d >= 12:
            break
t.hideturtle()
turtle.done()
```



This will be the starting of the result, yeah I know it is not something good. But wait for 10 to 15 sec for the output and it will be amazing.



Output! Give the feedback of this:)

12

SECTION-3

Extra Projects

Well, these extra projects for you guys took a lot of time to make because it creating **AMONG US** character and **INDIAN FLAG** were not easy for me as I am also a fresher in Python. Though these two were tough for me I am not sure, you all may have heard of these or even once have gone through these projects because drawing Indian Flag is the most popular program in python.

I am sure everybody has played Among Us game once in a while. So this time let's draw this which is among us character.

The fact is that no module is not required in this among us project also. The only turtle is needed.

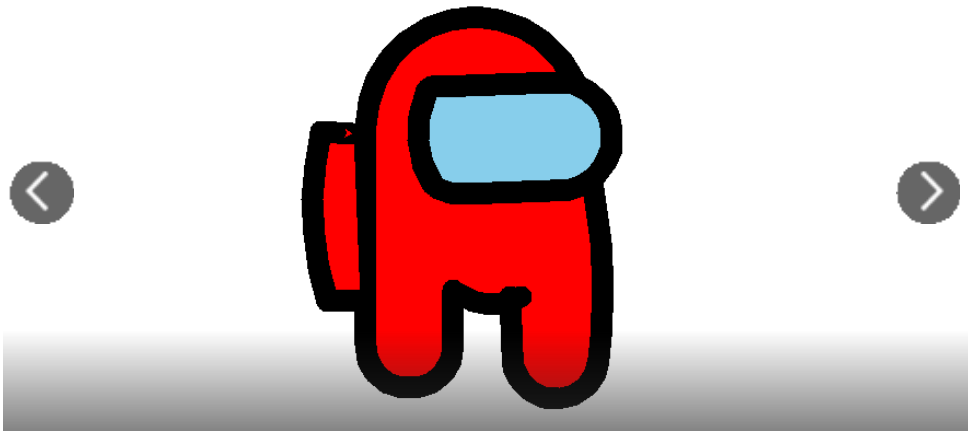
Source Code:

```
import turtle

BODY_COLOR =  'red'
BODY_SHADOW = ''
GLASS_COLOR = 'skyblue'
GLASS_SHADOW = ''

s = turtle.getscreen()
t = turtle.Turtle()

# it can move forward backward left right
def body():
    """ draws the body """
    t.pensize(20)
    #t.speed(15)

    t.fillcolor(BODY_COLOR)
    t.begin_fill()

    # right side
    t.right(90)
    t.forward(50)
    t.right(180)
    t.circle(40, -180)
    t.right(180)
    t.forward(200)
```

```
    # head curve
    t.right(180)
    t.circle(100, -180)

    # left side
    t.backward(20)
    t.left(15)
    t.circle(500, -20)
    t.backward(20)

    #t.backward(200)
    t.circle(40, -180)
    #t.right(90)
    t.left(7)
    t.backward(50)

    # hip
    t.up()
    t.left(90)
    t.forward(10)
    t.right(90)
    t.down()
    #t.right(180)
    #t.circle(25, -180)
    t.right(240)
    t.circle(50, -70)

    t.end_fill()


def glass():
    t.up()
    #t.right(180)
    t.right(230)
    t.forward(100)
    t.left(90)
    t.forward(20)
    t.right(90)

    t.down()
    t.fillcolor(GLASS_COLOR)
    t.begin_fill()

    t.right(150)
    t.circle(90, -55)

    t.right(180)
    t.forward(1)
    t.right(180)
    t.circle(10, -65)
    t.right(180)
    t.forward(110)
```

```
    t.right(180)

    #t.right(180)
    t.circle(50, -190)
    t.right(170)
    t.forward(80)

    t.right(180)
    t.circle(45, -30)

    t.end_fill()
def backpack():
    t.up()
    t.right(60)
    t.forward(100)
    t.right(90)
    t.forward(75)

    t.fillcolor(BODY_COLOR)
    t.begin_fill()

    t.down()
    t.forward(30)
    t.right(255)

    t.circle(300, -30)
    t.right(260)
    t.forward(30)

    t.end_fill()


body()
glass()
backpack()

t.screen.exitonclick()
```

Well, I know this a very lengthy code and yes this is not for beginners so they can just copy and paste, and for intermediates they can try to understand it's not that difficult for them.



This seems perfect?.... Well yes from me!! as because it took 4 days to find the precise coordinates or you can say vectors(maths).
You can choose your color, I chose red and blue and it worked for me.

13

Indian Flag

Well, this is also a popular project in python programming but I thought this would give me a patriotic feeling if I include this one.

We will use many features of turtle graphics like:

1.forward(x)

2.backward(x)

3.right(x) and left(x)

4.penup() and pendown()

5.begin_() and end_fill()

and much more features.

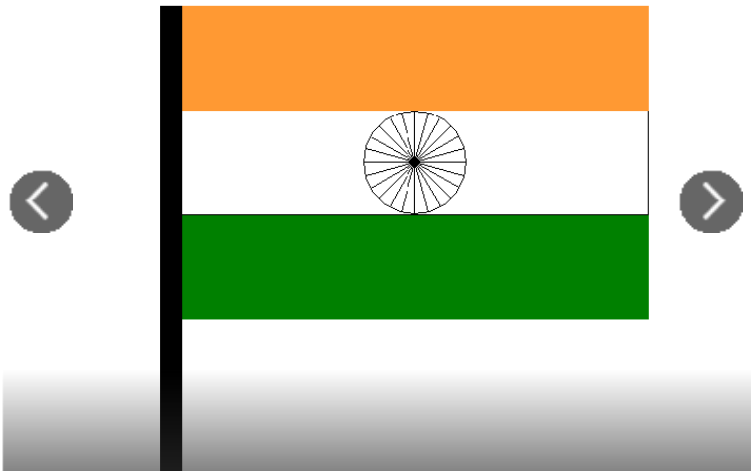
Let's divide this program into 4 steps:

1.First the turtle will draw a green rectangle or the bottom of the flag.
2.2nd step will be a white rectangle which is the middle one.
3.And the next step will be an orange rectangle or the top of the flag.
4.Last step will be the Ashok Chakra and pole.

```
import turtle

def make_outer(player):
    player.setheading(90)
    player.forward(100)
    player.setheading(0)
    player.forward(450)
    player.setheading(270)
    player.forward(100)
    player.setheading(180)
    player.forward(450)
    player.end_fill()


def make_circle(player):
    player.up()
    player.setpos(0,49)
```

```
    player.down()
    player.circle(49)

def make_spokes(player):
    for i in range(1,25):
        player.up()
        player.setpos(0,0)
        player.right(15)
        player.down()
        player.forward(49)

def make_stand(player):
    player.begin_fill()
    player.up()
    player.setpos(-225,150)
    player.setheading(180)
    player.down()
    player.forward(20)
    player.setheading(270)
    player.forward(450)
    player.setheading(0)
    player.forward(20)
    player.setheading(90)
    player.forward(450)
    player.end_fill()


def make_shape():
    window = turtle.Screen()
    window.bgcolor("white")
    window.setup(width=1.0, height=1.0, startx=None, starty=None)
    player = turtle.Turtle()
    player.shape("turtle")
    player.color("black")
    player.speed(1)

    player.up()
    player.setpos(-225,-150)
    player.down()
    player.color("green")
    player.begin_fill()
    make_outer(player)
```

```
    player.up()
    player.setpos(-225,-50)
    player.down()
    player.color("black")
    make_outer(player)

    player.up()
    player.setpos(-225,50)
    player.down()
    player.color("#ff9933")
    player.begin_fill()
    make_outer(player)

    player.color("black")
    make_circle(player)
    make_spokes(player)
    make_stand(player)
    player.up()
    player.setpos(250,200)

    window.exitonclick()
make_shape()
```



Finally,The pride of one's Nation is ready.

14

Hit or Move

That's my favorite program as I have made this game with help of my friend and this is a big project for me.
The concept of the game is similar to the video game Road Fighter if you have played this game in childhood. Car is moving in speed and some other cars come in the way and you have to evade from those cars, but in this game, some blocks will come in way of moving car and you have to evade from these blocks. Well, yeah the graphics will have no match.
So the name of game is 'Hit or Move' and now you know the reason for the title.

Modules required:

1.pygame

2.time

Pygame is the most important library in python if you want to make small and basic games.

Source Code:-

```
import pygame
import time
import random

pygame.init()

##############
crash_sound = pygame.mixer.Sound("crash.wav")
##############

display_width = 800
display_height = 600

black = (0,0,0)
white = (255,255,255)

red = (200,0,0)
green = (0,200,0)

bright_red = (255,0,0)
bright_green = (0,255,0)
```

```
block_color = (53,115,255)

car_width = 73

gameDisplay = pygame.display.set_mode((display_width,display_height))
pygame.display.set_caption('Hit or Move')
clock = pygame.time.Clock()

carImg = pygame.image.load('racecar.png')
gameIcon = pygame.image.load('carIcon.png')

pygame.display.set_icon(gameIcon)

pause = False
#crash = True

def things_dodged(count):
    font = pygame.font.SysFont("comicsansms", 25)
    text = font.render("Dodged: "+str(count), True, black)
    gameDisplay.blit(text,(0,0))

def things(thingx, thingy, thingw, thingh, color):
    pygame.draw.rect(gameDisplay, color, [thingx, thingy, thingw, thingh])

def car(x,y):
    gameDisplay.blit(carImg,(x,y))

def text_objects(text, font):
    textSurface = font.render(text, True, black)
    return textSurface, textSurface.get_rect()



def crash():
    ########################################
    pygame.mixer.Sound.play(crash_sound)
    pygame.mixer.music.stop()
    ########################################
    largeText = pygame.font.SysFont("comicsansms",115)
    TextSurf, TextRect = text_objects("You Crashed", largeText)
    TextRect.center = ((display_width/2),(display_height/2))
    gameDisplay.blit(TextSurf, TextRect)

    while True:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
```

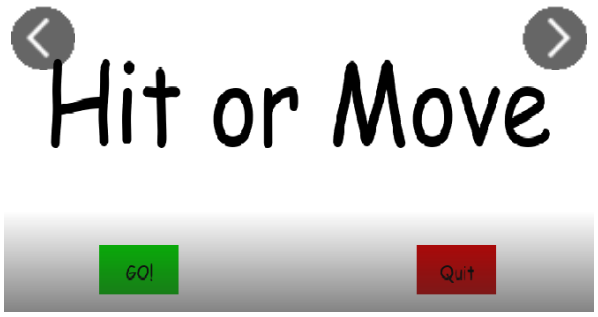
```
            quit()


        button("Play Again",150,450,100,50,green,bright_green,game_loop)
        button("Quit",550,450,100,50,red,bright_red,quitgame)

        pygame.display.update()
        clock.tick(15)

def button(msg,x,y,w,h,ic,ac,action=None):
    mouse = pygame.mouse.get_pos()
    click = pygame.mouse.get_pressed()

    if x+w > mouse[0] > x and y+h > mouse[1] > y:
        pygame.draw.rect(gameDisplay, ac,(x,y,w,h))
        if click[0] == 1 and action != None:
            action()
    else:
        pygame.draw.rect(gameDisplay, ic,(x,y,w,h))
    smallText = pygame.font.SysFont("comicsansms",20)
    textSurf, textRect = text_objects(msg, smallText)
    textRect.center = ( (x+(w/2)), (y+(h/2)) )
    gameDisplay.blit(textSurf, textRect)


def quitgame():
    pygame.quit()
    quit()

def unpause():
    global pause
    pygame.mixer.music.unpause()
    pause = False


def paused():
    #############
    pygame.mixer.music.pause()
    ##############
    largeText = pygame.font.SysFont("comicsansms",115)
    TextSurf, TextRect = text_objects("Paused", largeText)
    TextRect.center = ((display_width/2),(display_height/2))
    gameDisplay.blit(TextSurf, TextRect)


    while pause:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                quit()
```

```
        button("Continue",150,450,100,50,green,bright_green,unpause)
        button("Quit",550,450,100,50,red,bright_red,quitgame)

        pygame.display.update()
        clock.tick(15)


def game_intro():

    intro = True

    while intro:
        for event in pygame.event.get():
            #print(event)
            if event.type == pygame.QUIT:
                pygame.quit()
                quit()


        gameDisplay.fill(white)
        largeText = pygame.font.SysFont("comicsansms",115)
        TextSurf, TextRect = text_objects("Hit or Move", largeText)
        TextRect.center = ((display_width/2),(display_height/2))
        gameDisplay.blit(TextSurf, TextRect)

        button("GO!",150,450,100,50,green,bright_green,game_loop)
        button("Quit",550,450,100,50,red,bright_red,quitgame)

        pygame.display.update()
        clock.tick(15)




def game_loop():
    global pause
    ############
    pygame.mixer.music.load('jazz.wav')
    pygame.mixer.music.play(-1)
    ############
    x = (display_width * 0.45)
    y = (display_height * 0.8)

    x_change = 0

    thing_startx = random.randrange(0, display_width)
    thing_starty = -600
    thing_speed = 4
    thing_width = 100
```

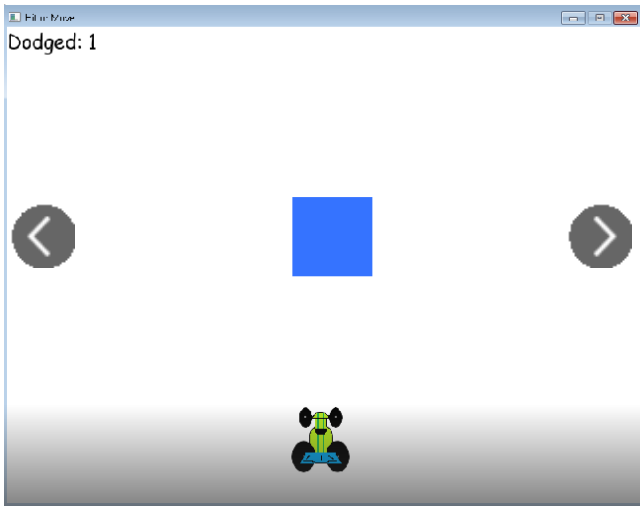
```
thing_height = 100

thingCount = 1

dodged = 0

gameExit = False

while not gameExit:

    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            quit()

        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_LEFT:
                x_change = -5
            if event.key == pygame.K_RIGHT:
                x_change = 5
            if event.key == pygame.K_p:
                pause = True
                paused()


        if event.type == pygame.KEYUP:
            if event.key == pygame.K_LEFT or event.key == pygame.K_RIGHT:
                x_change = 0

    x += x_change
    gameDisplay.fill(white)

    things(thing_startx, thing_starty, thing_width, thing_height, block_color)



    thing_starty += thing_speed
    car(x,y)
    things_dodged(dodged)

    if x > display_width - car_width or x < 0:
        crash()

    if thing_starty > display_height:
        thing_starty = 0 - thing_height
        thing_startx = random.randrange(0,display_width)
```

```
            dodged += 1
            thing_speed += 1
            thing_width += (dodged * 1.2)

        if y < thing_starty+thing_height:
            print('y crossover')

            if x > thing_startx and x < thing_startx + thing_width or x+car_width > thing_startx and x +
car_width < thing_startx+thing_width:
                print('x crossover')
                crash()

        pygame.display.update()
        clock.tick(60)
game_intro()
game_loop()
pygame.quit()
quit()
```

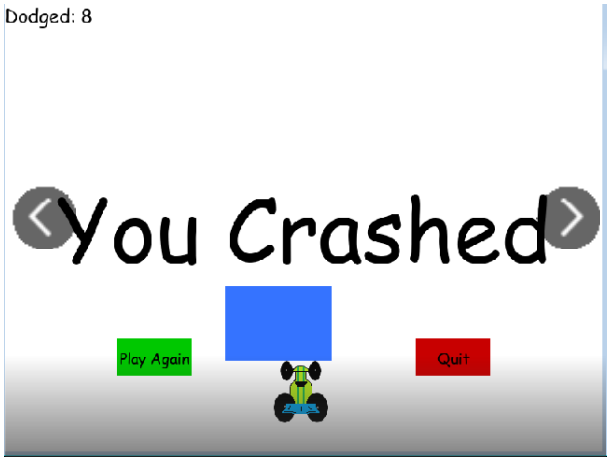
Yeah I know that's very big! you can just copy and paste this is also not my program I just helped my friend with colors, size, shapes, and design everything, and the coordinates which is the most important thing was my friends'.



When you save the code and run it will be shown like this and next you know...



As you can see the screenshot above,game will start like this and you have to evade the blocks and increase dodges or score.



And if you crashed with a block then you know the result.The Above picture shows the scene when you crashed and you will be asked if you want to play again or quit.

ENJOY THE GAME!

A Request

Thats END!.

This is my first ebook of life and I know it is not that good but I have tried my best to create this ebook. When I was learning python, since then only I decided to make an ebook of this programming language. First, it was a decision to teach you python from basics but then I dropped this idea as it would consume a lot of time.

So that's my request whoever is reading this was my first time and if I have made some mistakes please try to share with me your valuable suggestions.
The readers are expected to share their constructive criticism to accept my faults and make this book more effective.

I welcome my readers to reach me if they have any queries related to this book.

I would like to thanks UDEMY and GOOGLE for teaching me python. This work of mine would not be possible without these platforms.

Email id - **harwanijay9498@gmail.com**

Instagram user name : jay_harwani_

Github name : Jayctrl30

Name - Jay Harwani

It would be a pleasure by my side if you share your feedback of my work.

THANK YOU