



Python | Display text to PyGame window

Last Updated : 09 Apr, 2022

[Turtle](#) [Tkinter](#) [Matplotlib](#) [Python Imaging Library](#) [Pyglet](#) [Python](#) [Numpy](#) [Pandas](#) [Python Database](#)

games. It includes computer graphics and sound libraries designed to be used with the Python programming language. Now, it's up to the imagination or necessity of the developer, what type of game he/she wants to develop using this toolkit.

Command to install **pygame** on windows based system :

```
pip install pygame
```

There are 7-basic steps to displaying Text on the pygame window :

- Create a display surface object using `display.set_mode()` method of pygame.
- Create a Font object using `font.Font()` method of pygame.
- Create a Text surface object i.e.surface object in which Text is drawn on it, using `render()` method of pygame font object.
- Create a rectangular object for the text surface object using `get_rect()` method of pygame text surface object.
- Set the position of the Rectangular object by setting the value of the center property of pygame rectangular object.
- Copying the Text surface object to the display surface object using `blit()` method of pygame display surface object.
- Show the display surface object on the pygame window using the `display.update()` method of pygame.

Below is the implementation:

Python3

```
# import pygame module in this program
import pygame

# activate the pygame library
# initiate pygame and give permission
# to use pygame's functionality.
pygame.init()

# define the RGB value for white,
# green, blue colour .
white = (255, 255, 255)
green = (0, 255, 0)
blue = (0, 0, 128)

# assigning values to X and Y variable
X = 400
Y = 400

# create the display surface object
# of specific dimension..e(X, Y).
display_surface = pygame.display.set_mode((X, Y))

# set the pygame window name
pygame.display.set_caption('Show Text')

# create a font object.
# 1st parameter is the font file
# which is present in pygame.
# 2nd parameter is size of the font
font = pygame.font.Font('freesansbold.ttf', 32)

# create a text surface object,
# on which text is drawn on it.
text = font.render('GeeksForGeeks', True, green, blue)

# create a rectangular object for the
# text surface object
textRect = text.get_rect()

# set the center of the rectangular object.
textRect.center = (X // 2, Y // 2)

# infinite loop
while True:

    # completely fill the surface object
```

```
# with white color
display_surface.fill(white)

# copying the text surface object
# to the display surface object
# at the center coordinate.
display_surface.blit(text, textRect)

# iterate over the list of Event objects
# that was returned by pygame.event.get() method.
for event in pygame.event.get():

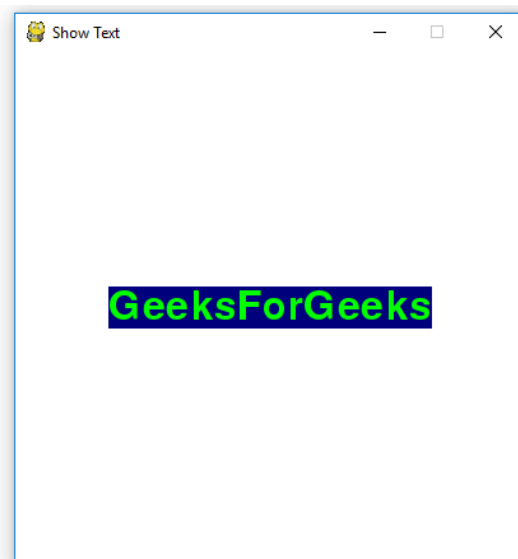
    # if event object type is QUIT
    # then quitting the pygame
    # and program both.
    if event.type == pygame.QUIT:

        # deactivates the pygame library
        pygame.quit()

        # quit the program.
        quit()

# Draws the surface object to the screen.
pygame.display.update()
```

Output :



Now We Will See one of the applications of Displaying the texts but in a different way that is by scrolling the text in 6 different ways on the pygame

window.

1. Scrolling the text on top of the Screen.
2. Scrolling the text at the bottom of the screen.
3. Scrolling the text on the left side of the Screen
4. Scrolling the text on the right side of the Screen
5. Scrolling the text in diagonal from left to right side of the Screen
6. Scrolling the text in diagonal from right side to left side of the Screen.

After Seeing the below Code you can implement your own pattern

Below is the Implementation

Python

```
# import pygame module in this program
import pygame

# activate the pygame library
# initiate pygame and give permission
# to use pygame's functionality.
pygame.init()

# create the display surface object
# (x, y) is the height and width of pygame window
win=pygame.display.set_mode((500, 500))

# set the pygame window name
pygame.display.set_caption("Scrolling Text")

# setting the pygame font style(1st parameter)
# and size of font(2nd parameter)
Font=pygame.font.SysFont('timesnewroman', 30)

# define the RGB value for white,
# green, yellow, orange colour
white=(255, 255, 255)
yellow=(255, 255, 0)
green=(0, 255, 255)
orange=(255, 100, 0)
done=False

# Split the text into letters
```

```
# 3rd parameter is font colour and
# 4th parameter is Font background
letter1=Font.render("H", False, orange, yellow)
letter2=Font.render("E", False, orange, green)
letter3=Font.render("M", False, orange, yellow)
letter4=Font.render("A", False, orange, green)
letter5=Font.render("N", False, orange, yellow)
letter6=Font.render("T", False, orange, green)
letter7=Font.render("H", False, orange, yellow)
```

```
# assigning values to
# i and c variable
i=0
c=1
```

```
# infinite loop
while not done:
    if(i>=820):
        i=0
        c+=1
        pygame.time.wait(500)
```

```
# completely fill the surface object
# with white color
win.fill(white)
if(c%6==0):
    # Scrolling the text in diagonal
    # on right side of the Screen.
    # copying the text surface object
    # to the display surface object
    # at the center coordinate.
    win.blit(letter1, (662-i, -162+i))
    win.blit(letter2, (639-i, -139+i))
    win.blit(letter3, (608-i, -108+i))
    win.blit(letter4, (579-i, -79+i))
    win.blit(letter5, (552-i, -52+i))
    win.blit(letter6, (529-i, -29+i))
    win.blit(letter7, (500 -i, 0 + i))
    i+=80
if(c%6==5):
    # Scrolling the text in diagonal on
    # left side of the Screen.
    win.blit(letter1, (-162+i, -162+i))
    win.blit(letter2, (-135+i, -135+i))
    win.blit(letter3, (-110+i, -110+i))
    win.blit(letter4, (-79+i, -79+i))
    win.blit(letter5, (-52+i, -52+i))
    win.blit(letter6, (-27+i, -27+i))
    win.blit(letter7, (0+i, 0+i))
```

```
# Decides the speed of
# the text on screen
i+=80
if(c%6==4):

    # Scrolling the text in
    # right side of the Screen.
    win.blit(letter1, (480, -180+i))
    win.blit(letter2, (480, -150+i))
    win.blit(letter3, (480, -120+i))
    win.blit(letter4, (480, -90+i))
    win.blit(letter5, (480, -60+i))
    win.blit(letter6, (480, -30+i))
    win.blit(letter7, (480, 0+i))

    # Decides the speed of
    # the text on screen
    i +=80
if(c%6==3):
    # Scrolling the text in left
    # side of the Screen.
    win.blit(letter1, (0, -180+i))
    win.blit(letter2, (0, -150+i))
    win.blit(letter3, (0, -120+i))
    win.blit(letter4, (0, -90+i))
    win.blit(letter5, (0, -60+i))
    win.blit(letter6, (0, -30+i))
    win.blit(letter7, (0, 0+i))

    # Decides the speed of
    # the text on screen
    i+=80
if(c%6==1):

    win.blit(letter1, (-124+i, 0))
    win.blit(letter2, (-102+i, 0))
    win.blit(letter3, (-82+i, 0))
    win.blit(letter4, (-58+i, 0))
    win.blit(letter5, (-40+i, 0))
    win.blit(letter6, (-19+i, 0))
    win.blit(letter7, (0+i, 0))

    # Decides the speed of
    # the text on screen
    i +=80
if(c%6==2):

    # Scrolling the text in bottom of the Screen.
```

```
win.blit(letter1, (-124+i, 470))
win.blit(letter2, (-102+i, 470))
win.blit(letter3, (-82+i, 470))
win.blit(letter4, (-58+i, 470))
win.blit(letter5, (-40+i, 470))
win.blit(letter6, (-19+i, 470))
win.blit(letter7, (0+i, 470))

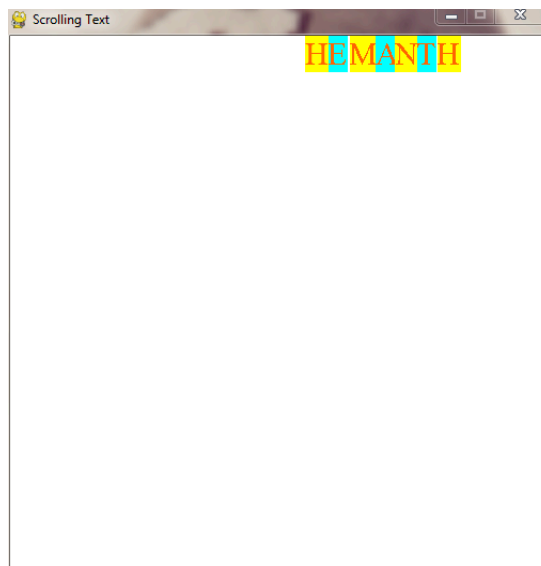
# Decides the speed
# of the text on screen
i+=80

# Draws the surface object to the screen.
pygame.display.update()

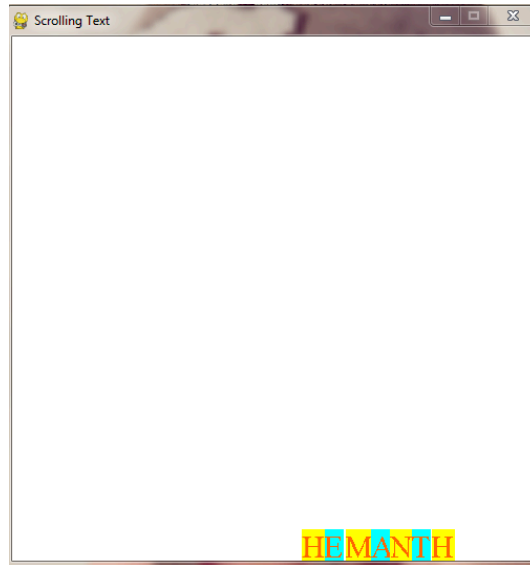
# iterate over the list of Event objects
# that was returned by pygame.event.get() method
for event in pygame.event.get():
    if(event.type==pygame.QUIT):
        done=True
#Delay with 5ms
pygame.time.wait(500)
pygame.quit()
```

Output:

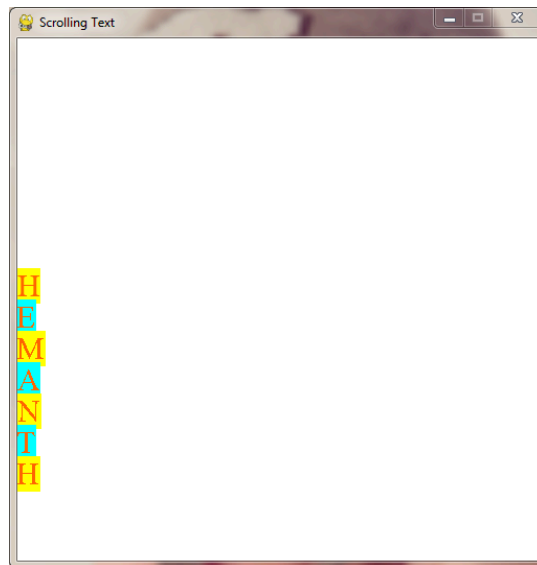
1. When text is Scrolling on top of Screen



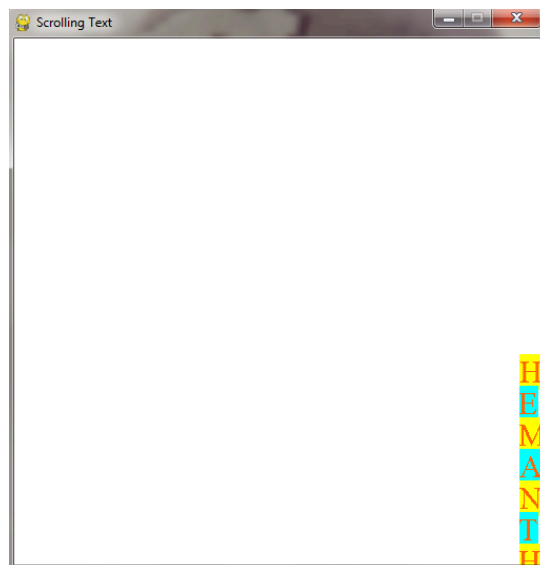
2. When text is Scrolling on bottom of Screen



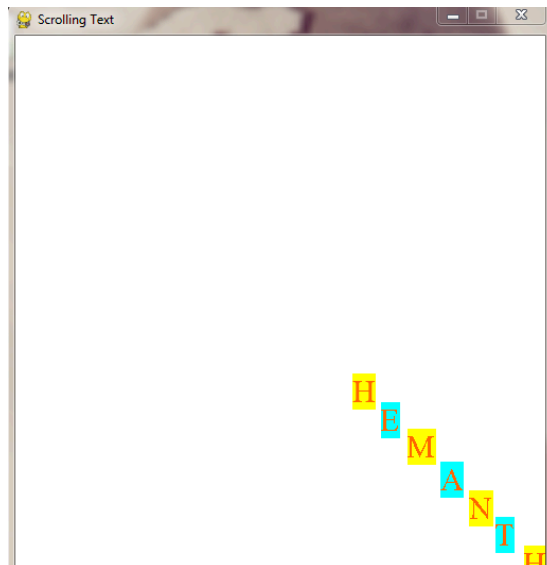
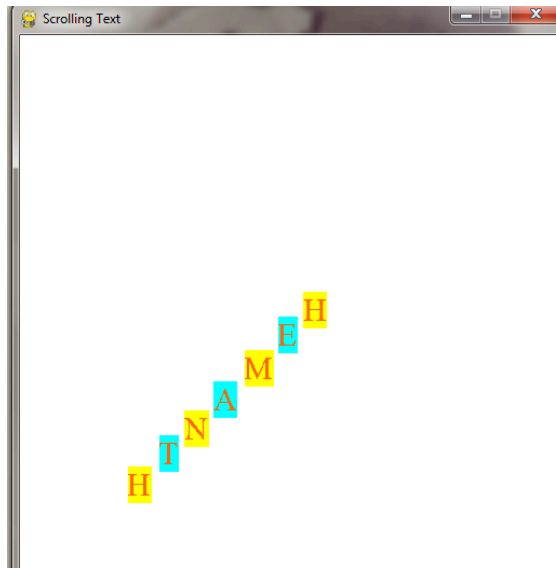
3. When text is Scrolling on left side of Screen



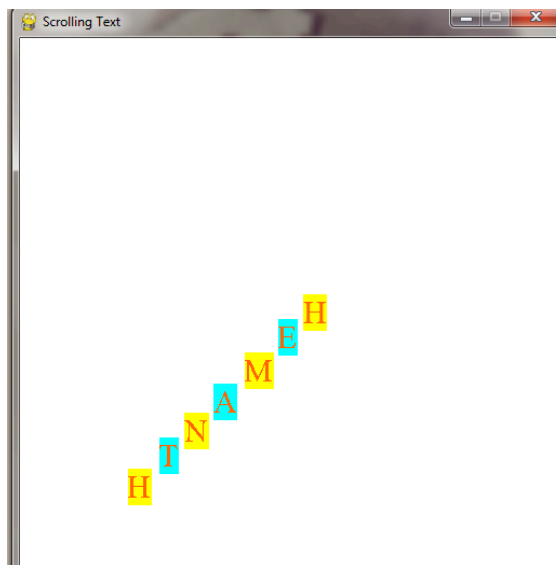
4. When text is Scrolling on right side of Screen



5. When text is Scrolling on diagonal from left side of Screen



6. When text is Scrolling on diagonal from right side of Screen





ankth...



50

Previous Article

[Pygame - Working with Text](#)

Next Article

[How to create a text input box with Pygame?](#)

Similar Reads

Adding Collisions Using pygame.Rect.colliderect in Pygame

Prerequisite: Drawing shapes in Pygame, Introduction to pygame In this article, we are going to use pygame.Rect.colliderect for adding collision in a shape usin...

3 min read

Python | Display images with PyGame

Pygame is a cross-platform set of Python modules designed for writing video games. It includes computer graphics and sound libraries designed to be used...

2 min read

Snowfall display using Pygame in Python

Not everybody must have witnessed Snowfall personally but wait a minute, What if you can see the snowfall right on your screen by just a few lines of...

3 min read

Python | Drawing different shapes on PyGame window

Pygame is a cross-platform set of Python modules designed for writing video games. It includes computer graphics and sound libraries designed to be used...

3 min read

How to create an empty PyGame window?

Pygame window is a simple window like any other window, in which we display our game screen. It is the first task we do so that we can display our output onto...

2 min read

Allowing resizing window in PyGame

In this article, we will learn How to allow resizing a PyGame Window. Game programming is very rewarding nowadays and it can also be used in advertising...

2 min read

How to get the size of PyGame Window?

In this article, we will learn How to get the size of a PyGame Window. Game programming is very rewarding nowadays and it can also be used in advertising...

2 min read

How to make a PyGame Window?

PyGame is a free and open-source cross-platform library for game development in Python. It was officially written by Pete Shinnars to replace PySDL it is suitable...

2 min read

How to Change the Name of a Pygame window?

PyGame window is a simple window that displays our game on the window screen. By default, pygame uses "Pygame window" as its title and pygame icon...

2 min read

How to make a fully transparent window with PyGame ?

Fully transparent windows with PyGame have their own advantage while building many games and apps via Python. Thus in order to achieve the...

3 min read

Article Tags :

[Python](#)

[Technical Scripter](#)

[Python-PyGame](#)

[python-utility](#)

[+1 More](#)

Practice Tags :

[python](#)



Corporate & Communications Address:-
A-143, 9th Floor, Sovereign Corporate
Tower, Sector- 136, Noida, Uttar Pradesh
(201305) | Registered Address:- K 061,
Tower K, Gulshan Vivante Apartment,
Sector 137, Noida, Gautam Buddh
Nagar, Uttar Pradesh, 201305



Company

About Us
Legal
In Media
Contact Us
Advertise with us
GFG Corporate Solution
Placement Training Program
GeeksforGeeks Community

DSA

Data Structures
Algorithms
DSA for Beginners
Basic DSA Problems
DSA Roadmap
Top 100 DSA Interview Problems
DSA Roadmap by Sandeep Jain
All Cheat Sheets

Web Technologies

HTML
CSS
JavaScript
TypeScript
ReactJS
NextJS
Bootstrap
Web Design

Computer Science

Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design
Engineering Maths
Software Development
Software Testing

System Design

High Level Design
Low Level Design
UML Diagrams
Interview Guide
Design Patterns

Languages

Python
Java
C++
PHP
GoLang
SQL
R Language
Android Tutorial
Tutorials Archive

Data Science & ML

Data Science With Python
Data Science For Beginner
Machine Learning
ML Maths
Data Visualisation
Pandas
NumPy
NLP
Deep Learning

Python Tutorial

Python Programming Examples
Python Projects
Python Tkinter
Web Scraping
OpenCV Tutorial
Python Interview Question
Django

DevOps

Git
Linux
AWS
Docker
Kubernetes
Azure
GCP
DevOps Roadmap

Interview Preparation

Competitive Programming
Top DS or Algo for CP
Company-Wise Recruitment Process
Company-Wise Preparation
Aptitude Preparation

OOAD

Puzzles

System Design Bootcamp

Interview Questions

School Subjects

Mathematics

Physics

Chemistry

Biology

Social Science

English Grammar

Commerce

World GK

GeeksforGeeks Videos

DSA

Python

Java

C++

Web Development

Data Science

CS Subjects

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved