



# Pygame – Drawing Objects and Shapes

Last Updated : 31 Oct, 2021

---

In this article, we are going to see how to draw an object using Pygame. There can be two versions for drawing any shape, it can be a solid one or just an outline of it.

## Drawing Objects and Shapes in PyGame

You can easily draw basic shapes in pygame using the draw method of pygame.

### Drawing Rectangle shape:

To draw a rectangle in your pygame project you can use `draw.rect()` function.

**Syntax:** `pygame.draw.rect(surface, color, rect, width)`

#### **Parameters:**

- *surface :- Here we can pass the surface on which we want to draw our rectangle. In the above example, we created a surface object named 'window'.*
- *color :- Here we can pass the color for our rectangle. We are using blue color in our example.*
- *rect :- Here we can pass the rectangle, position, and dimensions.*
- *width :- Here we can pass the line thickness. we can also create a solid rectangle by changing the value of this width parameter. So let's look at that.*

First, import the required module and initialize pygame. Now, Create the surface object of a specific dimension using the `display.set_mode()` method of pygame. Fill the background of the surface object with white color using the `fill()` function of pygame. Create a rectangle using the `draw.rect()` method of pygame. Update the Surface object.

---

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

---

## Python3

```
# Importing pygame module
import pygame
from pygame.locals import *

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

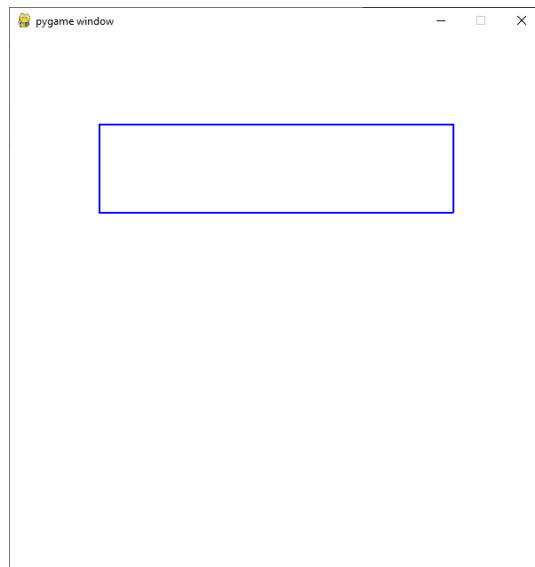
# create the display surface object
# of specific dimension.
window = pygame.display.set_mode((600, 600))

# Fill the scree with white color
window.fill((255, 255, 255))

# Using draw.rect module of
# pygame to draw the outlined rectangle
pygame.draw.rect(window, (0, 0, 255),
                  [100, 100, 400, 100], 2)

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

**Output :**



We can create a solid rectangle by setting the width parameter equal to 0 and the rest of the approach remains the same.

**Example 2:** Drawing a solid rectangle.

---

## Python3

```
# Importing pygame module
import pygame
from pygame.locals import *

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

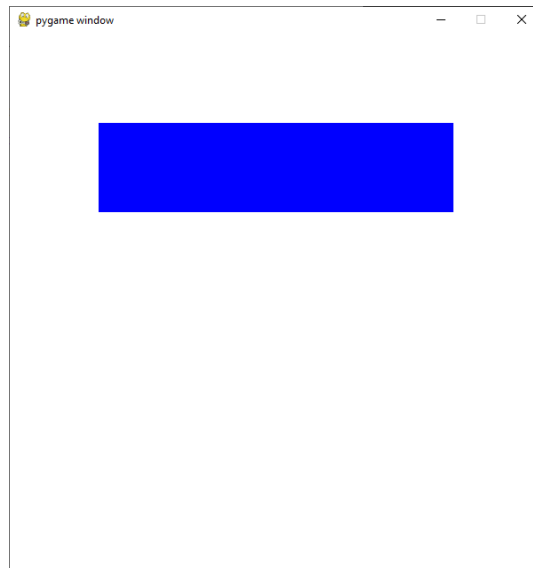
# create the display surface object
# of specific dimension.
window = pygame.display.set_mode((600, 600))

# Fill the screen with white color
window.fill((255, 255, 255))

# Using draw.rect module of
# pygame to draw the solid rectangle
pygame.draw.rect(window, (0, 0, 255),
                 [100, 100, 400, 100], 0)

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

## Output :



## Drawing Circle Shape:

To draw a circle in your pygame project you can use `draw.circle()` function. The entire approach is the same as above only the function and parameters are changed accordingly.

**Syntax :** `pygame.draw.circle(surface, color, center, radius, width)`

### Parameters :

- *surface :- Here we can pass the surface on which we want to draw our circle. In the above example, we created a surface object named 'window'.*
- *color :- Here we can pass the color for our circle. We are using green color in our example.*
- *center :- Here we can pass the ( x, y ) coordinates for the center of the circle.*
- *radius :- Here we can pass the radius of our circle.*
- *width :- Here we can pass the line thickness. we can also create a solid circle by changing the value of this width parameter. So let's look at that.*

## Example 1: Drawing a hollow circle.

---

### Python3

```
# Importing pygame module
import pygame
from pygame.locals import *

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

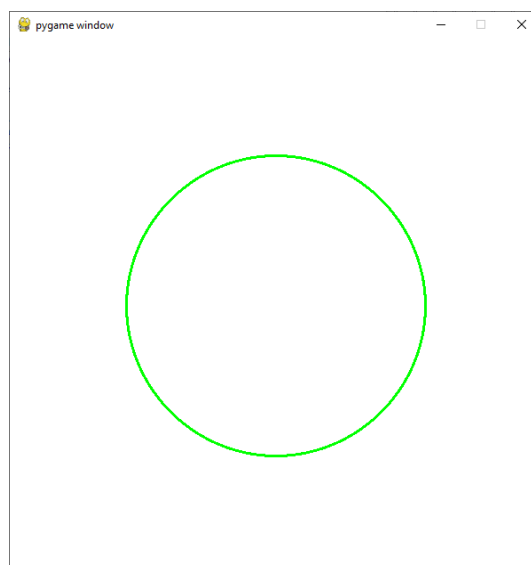
# create the display surface object
# of specific dimension.
window = pygame.display.set_mode((600, 600))

# Fill the scree with white color
window.fill((255, 255, 255))

# Using draw.rect module of
# pygame to draw the solid circle
pygame.draw.circle(window, (0, 255, 0),
                   [300, 300], 170, 3)

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

### Output :



We can create a solid circle by setting the width parameter equal to 0.

## Example 2: drawing a solid circle

---

### Python3

```
# Importing pygame module
import pygame
from pygame.locals import *

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

# create the display surface object
# of specific dimension.
window = pygame.display.set_mode((600, 600))

# Fill the scree with white color
window.fill((255, 255, 255))

# Using draw.rect module of
# pygame to draw the solid circle
pygame.draw.circle(window, (0, 255, 0),
                   [300, 300], 170, 0)

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

### Output :



Similarly, you can draw other basic shapes using the draw module of pygame.

## Drawing Polygon Shape:

The desired polygon can be drawn using `polygon()` function.

**Syntax:** `polygon(surface, color, points, width)`

Again the approach remains the same only the function and the parameters change.

**Example 1:** drawing a solid polygon

---

### Python3

```
# Importing pygame module
import pygame
from pygame.locals import *

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

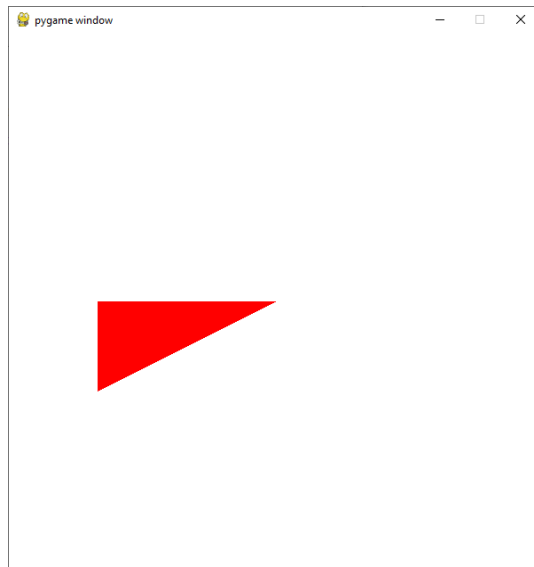
# create the display surface object
# of specific dimension.
window = pygame.display.set_mode((600, 600))

# Fill the scree with white color
window.fill((255, 255, 255))

# Using draw.rect module of
# pygame to draw the outlined polygon
pygame.draw.polygon(window, (255, 0, 0),
                    [[300, 300], [100, 400],
                     [100, 300]])

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

**Output :**



### Example 2: Drawing a hollow polygon

---

## Python3

```
# Importing pygame module
import pygame
from pygame.locals import *

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

# create the display surface object
# of specific dimension.
window = pygame.display.set_mode((600, 600))

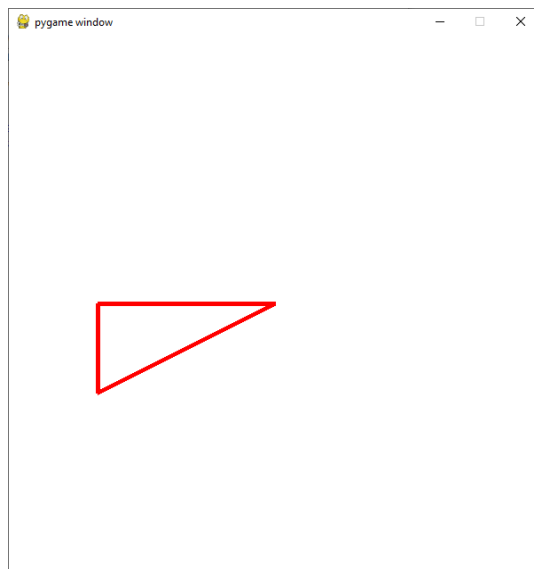
# Fill the scree with white color
window.fill((255, 255, 255))

# Using draw.rect module of
# pygame to draw the outlined polygon
pygame.draw.polygon(window, (255, 0, 0),
                    [[300, 300], [100, 400],
                     [100, 300]], 5)

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

### Output:





## Drawing Line Shape:

A line is the most basic drawing entity and can be drawn in pygame using `line()` function.

*Syntax : `pygame.draw.line(surface, color, start_pos, end_pos, width)`*

### Example 1: drawing a line

---

## Python3

```
# Importing pygame module
import pygame
from pygame.locals import *

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

# create the display surface object
# of specific dimension.
window = pygame.display.set_mode((600, 600))

# Fill the scree with white color
window.fill((255, 255, 255))

# Using draw.rect module of
# pygame to draw the line
```

```
pygame.draw.line(window, (0, 0, 0),  
                  [100, 300],  
                  [500, 300], 5)  
  
# Draws the surface object to the screen.  
pygame.display.update()
```

## Output:



## Draw Multiple Shapes:

You can draw multiple shapes on the same surface object. For that, the first required modules are imported and pygame is initialized. Now, create the surface object of a specific dimension using the `display.set_mode()` method of pygame. Fill the background of the surface object with white color using the `fill()` function of pygame. Create required shapes are described above. Update the Surface object

**Example 1:** Drawing a circle inside a rectangle.

---

## Python3

```
# Importing pygame module  
import pygame  
from pygame.locals import *  
  
# initiate pygame and give permission  
# to use pygame's functionality.  
pygame.init()
```

```
# create the display surface object
# of specific dimension.
window = pygame.display.set_mode((600, 600))

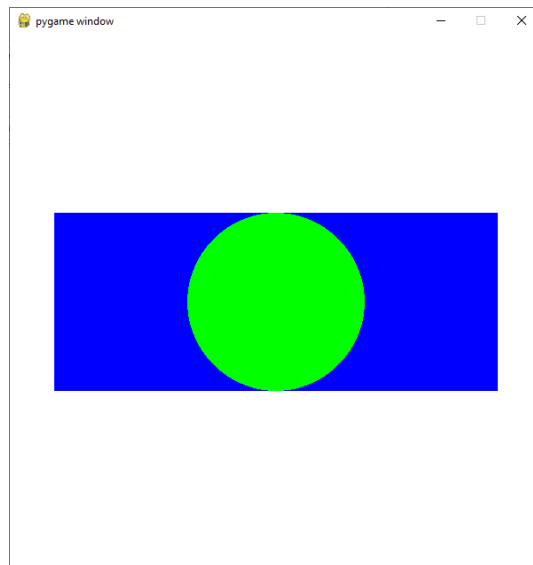
# Fill the scree with white color
window.fill((255, 255, 255))

# Using draw.rect module of
# pygame to draw the rectangle
pygame.draw.rect(window, (0, 0, 255),
                 [50, 200, 500, 200])

# Using draw.rect module of
# pygame to draw the circle inside the rectangle
pygame.draw.circle(window, (0, 255, 0),
                  [300, 300], 100)

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

## Output:



**Example 2:** Drawing rectangles inside one another.

---

## Python3

```
# Importing pygame module
import pygame
from pygame.locals import *

# initiate pygame and give permission
# to use pygame's functionality.
```

```
pygame.init()

# create the display surface object
# of specific dimension.
window = pygame.display.set_mode((600, 600))

# Fill the scree with white color
window.fill((255, 255, 255))

# Creating a list of different rects
rectangle_list = [pygame.Rect(50, 100, 500, 200),
                  pygame.Rect(70, 120, 460, 160),
                  pygame.Rect(90, 140, 420, 120),
                  pygame.Rect(110, 160, 380, 80),
                  pygame.Rect(130, 180, 340, 40)
                  ]

# Creating list of different colors
color_list = [(0, 0, 0),
              (255, 255, 255),
              (0, 0, 255),
              (0, 255, 0),
              (255, 0, 0)
              ]

# Creating a variable which we will use
# to iterate over the color_list
color_var = 0

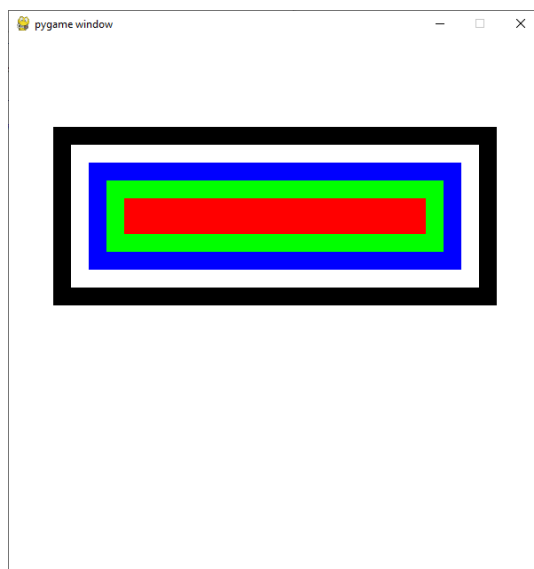
# Iterating over the rectangle_list using
# for loop
for rectangle in rectangle_list:

    # Drawing the rectangle
    # using the draw.rect() method
    pygame.draw.rect(window, color_list[color_var],
                     rectangle)

    # Increasing the value of color_var
    # by 1 after every iteration
    color_var += 1

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

**Output :**



## Writing your own drawing functions:

You can easily create your own specialized drawing functions in pygame.

This can be done by following the given procedure. Create a drawing function that will take starting position, width, and height as parameters. Draw required shapes as described above. Call the `drawfunction()`

## Python3

```
# Importing pygame module
import pygame
from pygame.locals import *

# Creating Drawing function

def drawingfunction(x, y, width, height):

    # Creating rectangle using the draw.rect() method
    pygame.draw.rect(window, (0, 0, 255), [x, y, width, height])

    # Calculation the center of the circle
    circle_x = width/2 + x
    circle_y = height/2 + y

    # Calculating the radius of the circle
    if height < width:
        radius = height/2
    else:
        radius = width/2
```

```
# Drawing the circle
pygame.draw.circle(window, (0, 255, 0), [circle_x, circle_y], radius)

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

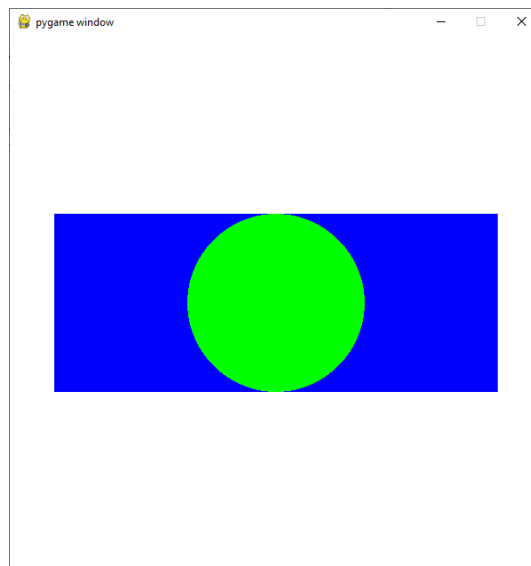
# create the display surface object
# of specific dimension.
window = pygame.display.set_mode((600, 600))

# Fill the scree with white color
window.fill((255, 255, 255))

# Calling the drawing function
drawingfunction(50, 200, 500, 200)

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

## Output:



## Drawing shapes with the mouse:

Now let's see how we can create shapes whenever the user clicks the mouse. We are going to create circles in the next example but you can create any shape you want.

Create a list to store the position of the shape to be drawn. Create a variable to store the color of the shape. Create a variable which we will use to run the

while loop and Create a while loop. Iterate over all the events received from `pygame.event.get()`. If the type of the event is quit then setting the run variable to false. If the type of the event is `MOUSEBUTTONDOWN` ( this event occurs when a user presses the mouse button) then getting the current position in a variable then appending that position in our `circle_positions` list. Iterate over all the positions in of the array created using a for a loop. Keep drawing the shape. Update the surface object.

---

## Python3

```
# Importing pygame module
import pygame
from pygame.locals import *

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

# create the display surface object
# of specific dimension.
window = pygame.display.set_mode((600, 600))

# Fill the scree with white color
window.fill((255, 255, 255))

# creating list in which we will store
# the position of the circle
circle_positions = []

# radius of the circle
circle_radius = 60

# Color of the circle
color = (0, 0, 255)

# Creating a variable which we will use
# to run the while loop
run = True

# Creating a while loop
while run:

    # Iterating over all the events received from
    # pygame.event.get()
    for event in pygame.event.get():
```

```
# If the type of the event is quit
# then setting the run variable to false
if event.type == QUIT:
    run = False

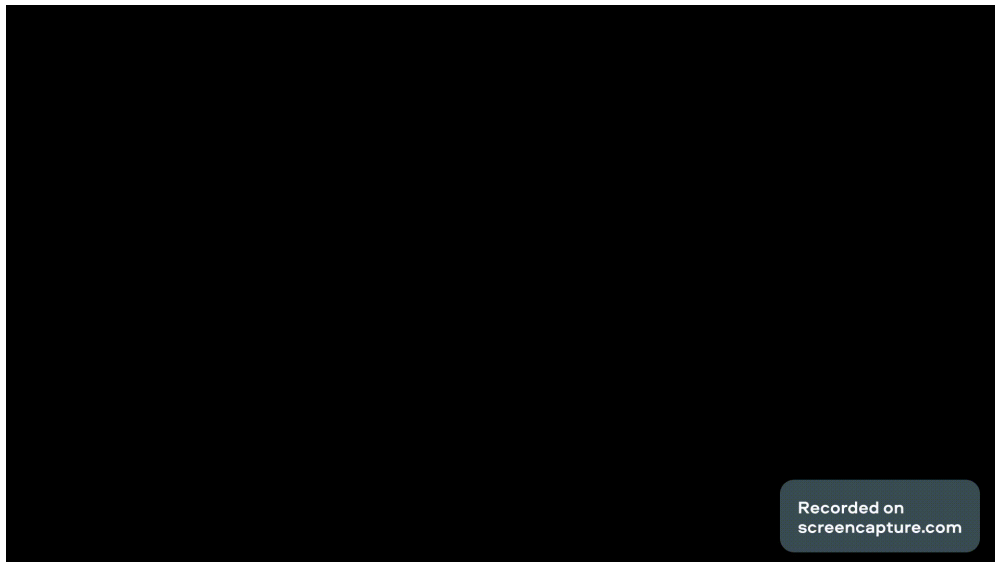
# if the type of the event is MOUSEBUTTONDOWN
# then storing the current position
elif event.type == MOUSEBUTTONDOWN:
    position = event.pos
    circle_positions.append(position)

# Using for loop to iterate
# over the circle_positions
# list
for position in circle_positions:

    # Drawing the circle
    pygame.draw.circle(window, color, position,
                        circle_radius)

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

## Output:



Looking to dive into the world of programming or sharpen your Python skills? Our [Master Python: Complete Beginner to Advanced Course](#) is your ultimate guide to becoming proficient in Python. This course covers everything you need to build a solid foundation from fundamental programming concepts to



advanced techniques. With **hands-on projects**, real-world examples, and expert guidance, you'll gain the confidence to tackle complex **coding challenges**. Whether you're starting from scratch or aiming to enhance your skills, this course is the perfect fit. Enroll now and master Python, the language of the future!

I imra...



8

### Previous Article

[Pygame - Time](#)

### Next Article

[Python | Drawing different shapes on PyGame window](#)

## Similar Reads

### 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](#)

### Python - Drawing design using arrow keys in 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...

[3 min read](#)

### 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 | Creating a Simple Drawing App in kivy

Kivy is a platform-independent GUI tool in Python. As it can be run on Android, IOS, Linux and Windows, etc. It is basically used to develop the Android...

[4 min read](#)

## PyCairo - Drawing different type of line caps

In this article, we will learn how we can draw different line caps types using PyCairo in python. The line caps are endpoints of lines. Steps of Implementation...

3 min read

## Wand Drawing() function in Python

Another module for wand is wand.drawing. This module provides us with some very basic drawing functions. wand.drawing.Drawing object buffers instructions...

1 min read

## Live Webcam Drawing using OpenCV

Let us see how to draw the movement of objects captured by the webcam using OpenCV. Our program takes the video input from the webcam and tracks the...

3 min read

## PYGLET – Drawing Mouse Cursor for Window

In this article we will see how we can draw mouse cursor for window in PYGLET module in python. Pyglet is easy to use but powerful library for developing...

2 min read

## PYGLET – Drawing Label

In this article we will see how we can draw a label on the window in PYGLET module in python. Pyglet is easy to use but powerful library for developing...

2 min read

## PYGLET – Drawing Multiple Sprites

In this article, we will see how we can draw multiple sprites on the window in PYGLET module in python. Pyglet is easy to use but powerful library for...

3 min read

**Article Tags :** [Python](#) [Python-PyGame](#)

**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

## 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

NextJS  
Bootstrap  
Web Design

Python Interview Question  
Django

## 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  
OOAD  
System Design Bootcamp  
Interview Questions

## School Subjects

Mathematics  
Physics  
Chemistry  
Biology  
Social Science  
English Grammar  
Commerce  
World GK

## 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  
Puzzles

## GeeksforGeeks Videos

DSA  
Python  
Java  
C++  
Web Development  
Data Science  
CS Subjects

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved