In [7]:
```
import cv2
import numpy as np
```

Draw-Function

In [8]:
```
#------------------------------------------------------------- Global Variabl
text=None
x1=None
y1=None
# permission variable
allow_pen=False
draw=False
allow_eraser=False
allow_rect=False
allow_circle=False
allow_line=False
allow_text=False

# ------------------------------------------------------------- Draw Functic
def drawfun(event,x,y,flags,user):
    global allow_pen,x1,y1,draw,allow_eraser,allow_circle,allow_rect,allow_lir

    #----------------------for pen
    if allow_pen:
        if event ==1:
            draw=True
            x1=x
            y1=y
        elif event==0:
            if draw:
                col_x=colors()
                cv2.line(white,(x1,y1),(x,y),col_x[0],col_x[1])
                x1=x
                y1=y
        elif event==4:
            draw=False
            x1=None
            y1=None

    #---------------------------- for eraser
    if allow_eraser:
        if event ==1:
            draw=True
            x1=x
            y1=y
        elif event==0:
            if draw:
                rad=cv2.getTrackbarPos("Eraser","Paint")
                cv2.circle(white,(x1,y1),rad,(255,255,255),-1)
                x1=x
                y1=y
```

```
        elif event==4:
            draw=False
            x1=None
            y1=None

    # ------------------------------------for Rectangle
    if allow_rect:
        if event ==1:
            x1=x
            y1=y
        elif event==4:
            col_x=colors()
            cv2.rectangle(white,(x1,y1),(x,y),col_x[0],col_x[1])
            x1=None
            y1=None
    # ------------------------------------ for circle
    if allow_circle:
        if event==1:
            x1=x
            y1=y
        elif event==4:
            col_x=colors()
            rad=cv2.getTrackbarPos("Circle Size","Paint")
            cv2.circle(white,(x1,y1),rad,col_x[0],col_x[1])
            x1=None
            y1=None

    # ----------------------------------------for line
    if allow_line:
        if event==1:
            x1=x
            y1=y
        elif event ==4:
            pen_x=colors()
            cv2.line(white,(x1,y1),(x,y),pen_x[0],pen_x[1])
            x1=None
            y1=None
    # -------------------------fro text
    if allow_text:
        fs=cv2.getTrackbarPos("Font Size","Paint")
        pen_x=colors()
        if event==1:
            cv2.putText(white,text,(x,y),cv2.FONT_ITALIC,fs,pen_x[0],pen_x[1])
            text=None
```

Main Function

In [9]:
```
# Created the Name Window ----- Note used same name at all the points
cv2.namedWindow("Paint",cv2.WINDOW_NORMAL)
cv2.setMouseCallback("Paint",drawfun)
```

```python

# Dummpy Funtion for TrackBar
def zx(a):
    pass


# ----------------------------------------------------  4 TrackBar For Pen, R
cv2.createTrackbar("Blue","Paint",0,255,zx)
cv2.createTrackbar("Green","Paint",0,255,zx)
cv2.createTrackbar("Red","Paint",0,255,zx)
cv2.createTrackbar("Thickness","Paint",1,10,zx) # if got thickness error then

def colors():
    b=cv2.getTrackbarPos("Blue","Paint")
    g=cv2.getTrackbarPos("Green","Paint")
    r=cv2.getTrackbarPos("Red","Paint")
    th=cv2.getTrackbarPos("Thickness","Paint")
    return [(b,g,r),th]



# ------------------------------------------------------------------------  1 Tra
cv2.createTrackbar("Eraser","Paint",1,100,zx)
                                                # its get position we will use

# ------------------------------------------------------------------------  1 Tra
cv2.createTrackbar("Circle Size","Paint",1,350,zx)
                                                    # its get position we will

cv2.createTrackbar("Font Size","Paint",1,10,zx)



# ------------------------------------------------------------------------  While

white=np.full((700,700,3),255,dtype=np.uint8)

grid=True

count=1

while True:
    # Instruction Section Decoration

    cv2.rectangle(white,(0,550),(700,700),(0,0,0),-1)
    ins=  "c = Circle  r = Rectange   e = Eraser   d = Clear Page"
    ins2= "t = text   p =Pen   q =Close  l =Line  g =H-grid"
    ins3= "NOTE:C,R,E,L,T and P used to Deactivate the function"
    cv2.putText(white,ins,(15,580),cv2.FONT_ITALIC,0.8,(0,0,255),2)
    cv2.putText(white,ins2,(15,630),cv2.FONT_ITALIC,0.8,(0,0,255),2)
    cv2.putText(white,ins3,(15,680),cv2.FONT_ITALIC,0.8,(0,0,255),2)
```

```
cv2.imshow("Paint",white)

# Break Key
if (cv2.waitKey(1) & 255)==ord('q'):
    break

# Page Clear Key
if (cv2.waitKey(1) & 255)==ord('d'):
    grid=True
    white=np.full((700,700,3),255,dtype=np.uint8)



# Allow Pen Key
if (cv2.waitKey(1) & 255)==ord('p'):
    allow_eraser=False # when pen activaeted eraser should deactivate
    allow_rect=False
    allow_circle=False
    allow_line=False

    allow_pen=True
    print("Pen Activated")

# Remove Pen key
if (cv2.waitKey(1) & 255)==ord('P'):
    allow_pen=False
    print("Pen Deactivated")



# Allow Eraser Key
if (cv2.waitKey(1) & 255)==ord('e'):
    allow_pen=False # when eraser activated all should deactivated
    allow_rect=False
    allow_circle=False
    allow_line=False

    allow_eraser=True
    print("Eraser Activated")

# Remove Eraser Key
if (cv2.waitKey(1) & 255)==ord('E'):
    allow_eraser=False
    print("Eraser Deactivated")

# Allow Rectangel
if (cv2.waitKey(1) & 255)==ord('r'):
    allow_pen=False # when rectangle activated all should deactivated
    allow_eraser=False
```

```
        allow_circle=False
        allow_line=False

        allow_rect=True
        print("Rectangle Activated")

    # Deactivate Rectangle
    if (cv2.waitKey(1) & 255)==ord('R'):
        allow_rect=False
        print("Rectangle Deactivated")

    # Circle Activate
    if (cv2.waitKey(1) & 255)==ord('c'):
        allow_eraser=False # dectivate all other things while one thing start
        allow_pen=False
        allow_rect=False
        allow_line=False

        allow_circle=True
        print("Circle Activated")

    # circle deactivate
    if (cv2.waitKey(1) & 255)==ord('C'):
        allow_circle=False
        print("Circle Deactivated")


    # Line Activate
    if (cv2.waitKey(1) & 255)==ord('l'):
        allow_eraser=False # dectivate all other things while one thing start
        allow_pen=False
        allow_rect=False
        allow_circle=False

        allow_line=True
        print("line Activated")

    # Line deactivate
    if (cv2.waitKey(1) & 255)==ord('L'):
        allow_line=False
        print("line Deactivated")


    # text activated
    if (cv2.waitKey(1) & 255)==ord('t'):
        allow_eraser=False # dectivate all other things while one thing start
        allow_pen=False
        allow_rect=False
        allow_circle=False
        allow_line=False

        allow_text=True
        print("text activated")
```

```
        text=input()

    # text deactivated
    if (cv2.waitKey(1) & 255)==ord('T'):
        allow_text=False
        print("text Deactivated")

    # to save the frame
    #if cv2.waitKey(4):
    #    cv2.imwrite(r"C:\CV\ComputerVision\Scripts\Created Images\Animation F
    #    count+=1


    # Grid Line
    if (cv2.waitKey(1) & 255)==ord('g') and grid:
        count=0
        while True:
            cv2.line(white,(0,count),(700,count),(0,0,0),1)
            count=count+10
            if count==700:
                grid=False
                break

cv2.destroyAllWindows()
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: