Coding Graphics Workshop

Saturday, 13 July 2019

Andi Dinata – CodeClubJr
KFC Lippo Cikarang

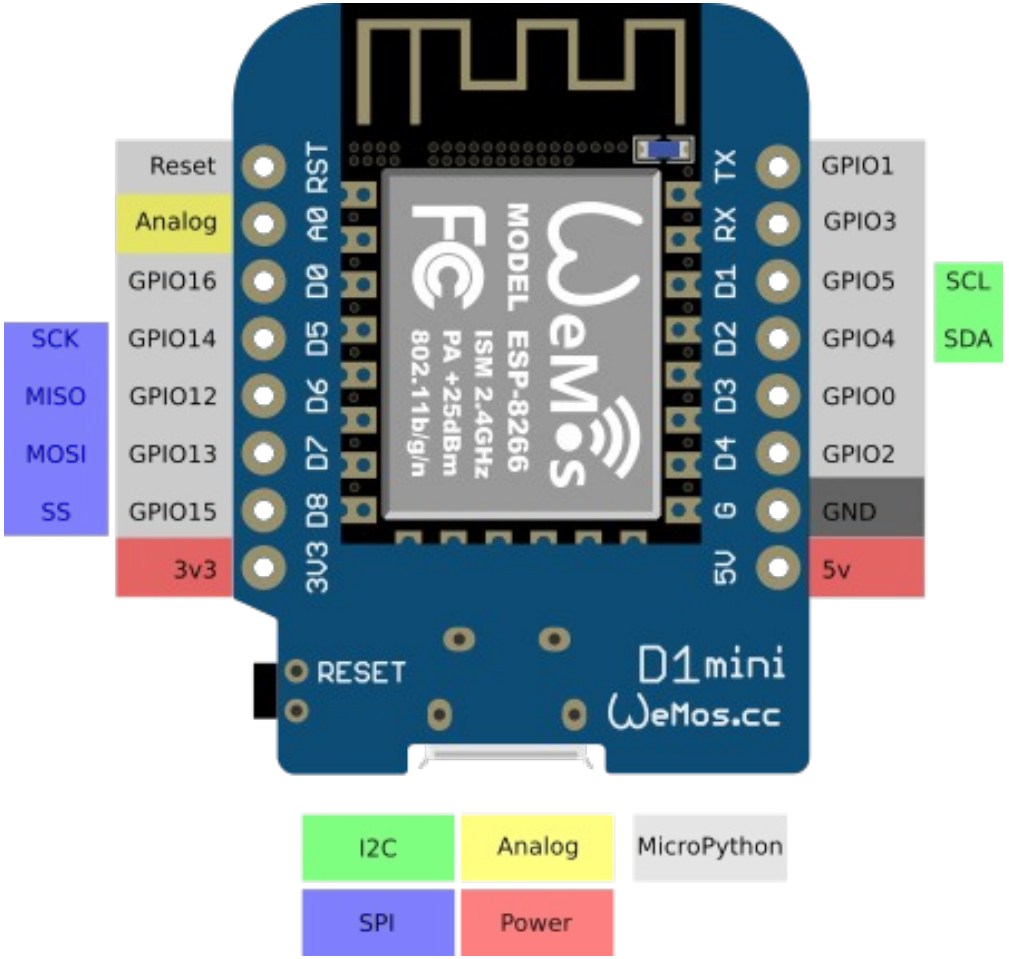
Content

- Led Matrix MAX7219 and ESP8266
- Introduction to 2D Coordinate (X,Y)
- PIXEL
- SHAPE
- DRAW ART
- ANIMATION

House Rules

- 120 minutes is compact
- Question, question, question
- Listen, listen, listen
- One by one
- Practice, practice, practice
- Quiz at the end

WEMOS D1 Mini



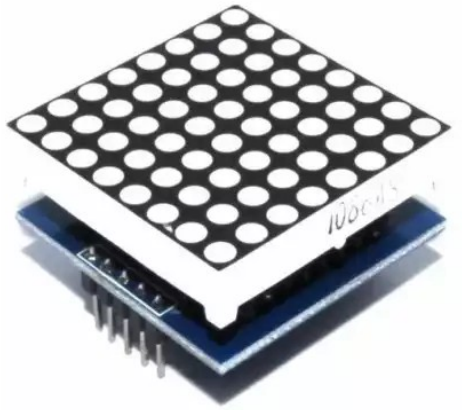
- Microcontroller ESP8266
- 80 or 160 MHz
- 4MB Flash RAM
- 32 Kb RAM
- 9 Digital Pin
- 1 Analog Pin
- 5 and 3.3 Volt
- WIFI
- USB
- Windows/Mac : need driver
- Linux: built-in

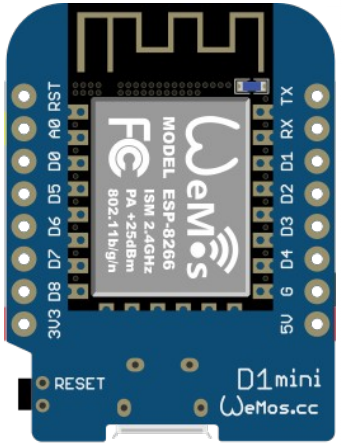
LED MATRIX

- 64 individual led
- MAX7219 controller
- SPI (=Serial Peripheral Interface) sonnection
- 3.3 Volt

Connections

HARDWARE





Micro USB Cable
Data exchange and Power

LED MATRIX

As object that receive and execute instruction from microcontroller

MICROCONTROLLER

Produces instruction to led matrix instantly based on the code input

COMPUTER

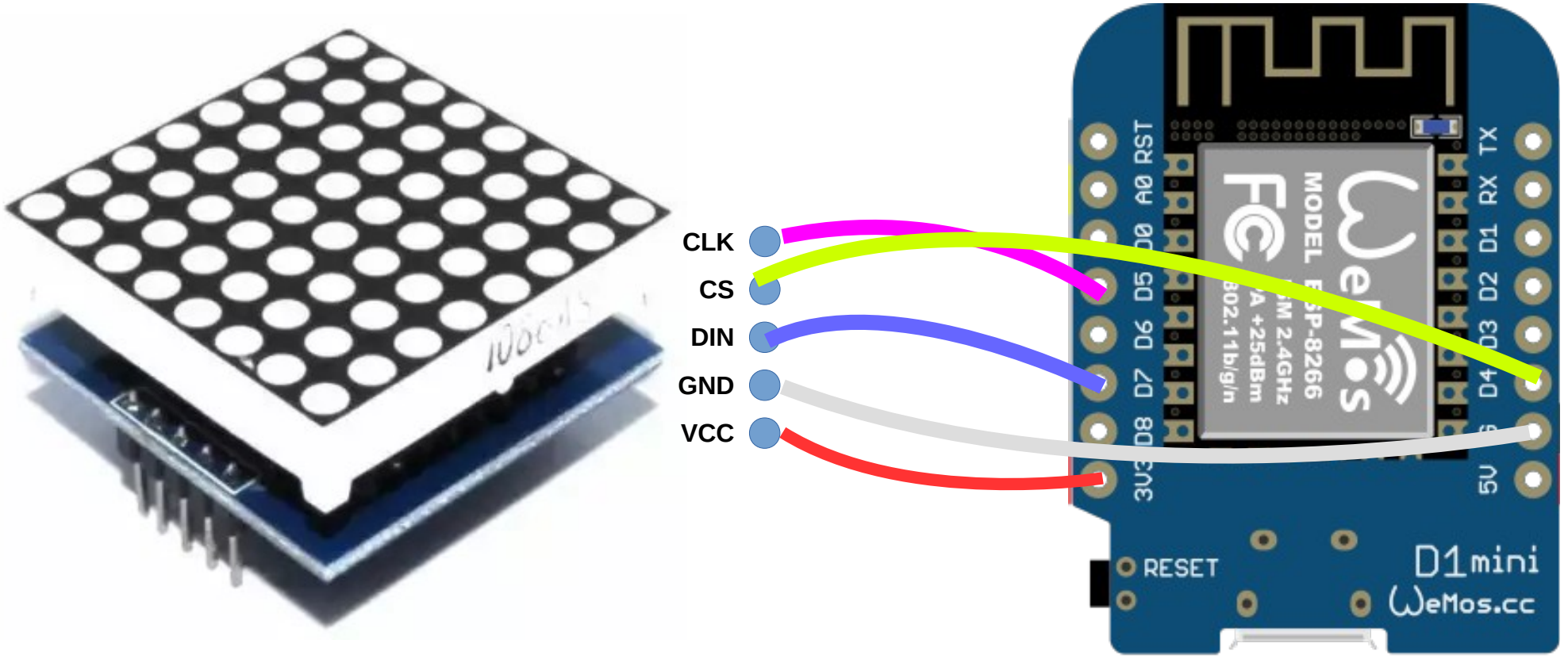
To send micropython code over USB to microcontroller from and IDE software

SOFTWARE

MicroPython
runs here

Thonny IDE

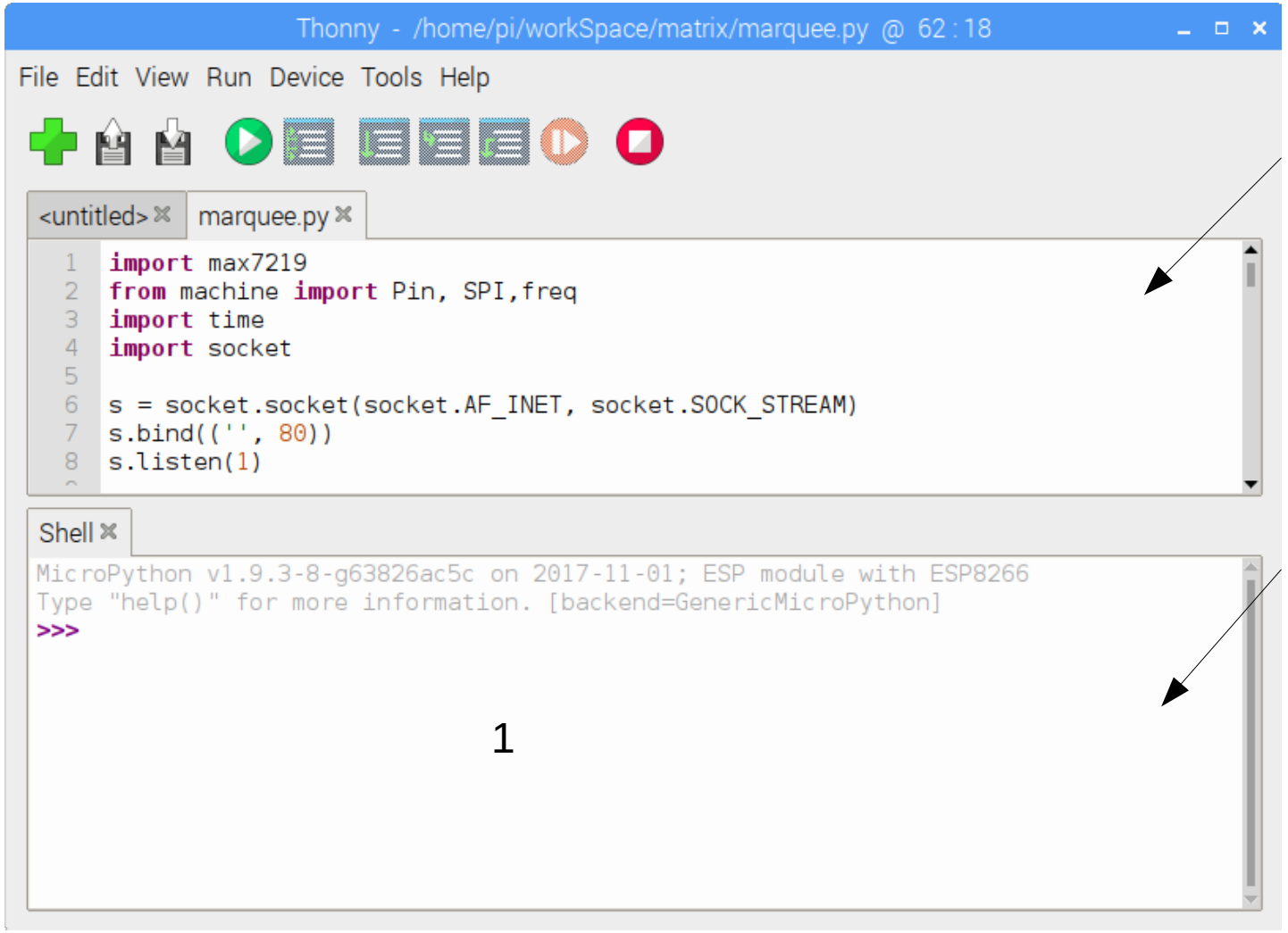
WIRING – JUMPER CABLE



Led Matrix	ESP8266
CLK	D7
CS	D4
DIN	D5
GND	G
VCC	3V3

Thonny IDE

EDITOR **SHELL**



EDITOR Area

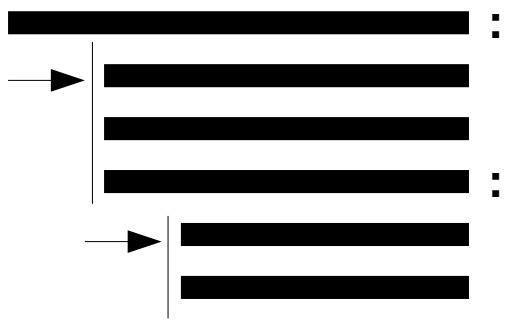
* Code written here needs to be sent to microcontroller in order to run the program (F5 Key)
* Code can be saved as a file in the computer
* Code lines is numbered for tracking

SHELL Area

* Code written here will be executed by microcontroller, directly
* Code is not saved. Very useful to test the code / part of the code
* Prompt >>>

Micropython is language
Elements to remember

1. The first number is 0 (zero)
   count 1 to 10 will result 0 1 2 3 4 5 6 7 8 9
2. Case sensitive
   Capital and non-capital letters are different. A≠a
3. The use of space
4. 3 types of data: **integer**, **float**, **string**
   Integer = whole number, e.g. 1,2,3,100
   float = decimal number e.g. 1.0, 2.5, 100.0
   string = text data, written in “ “ (double-quote)
5. Function ended with ( ) sign
6. Code after : sign is written indented



When you forget...don’t worry

Quick guide will be deployed, type at SHELL

```
>>> display.iforgot()
```

Quick Guide

Hardware initiation
--- Copy and paste when writing a program

```
import max7219
from machine import Pin, SPI,freq
import time
tile=1
freq(160000000)
spi=SPI(1, baudrate=10000000, polarity=0, phase=0)
display=max7219.Matrix8x8(spi, Pin(2), tile)
display.brightness(3)
display.fill(0)
display.show()
```

---end of hardware initiation---

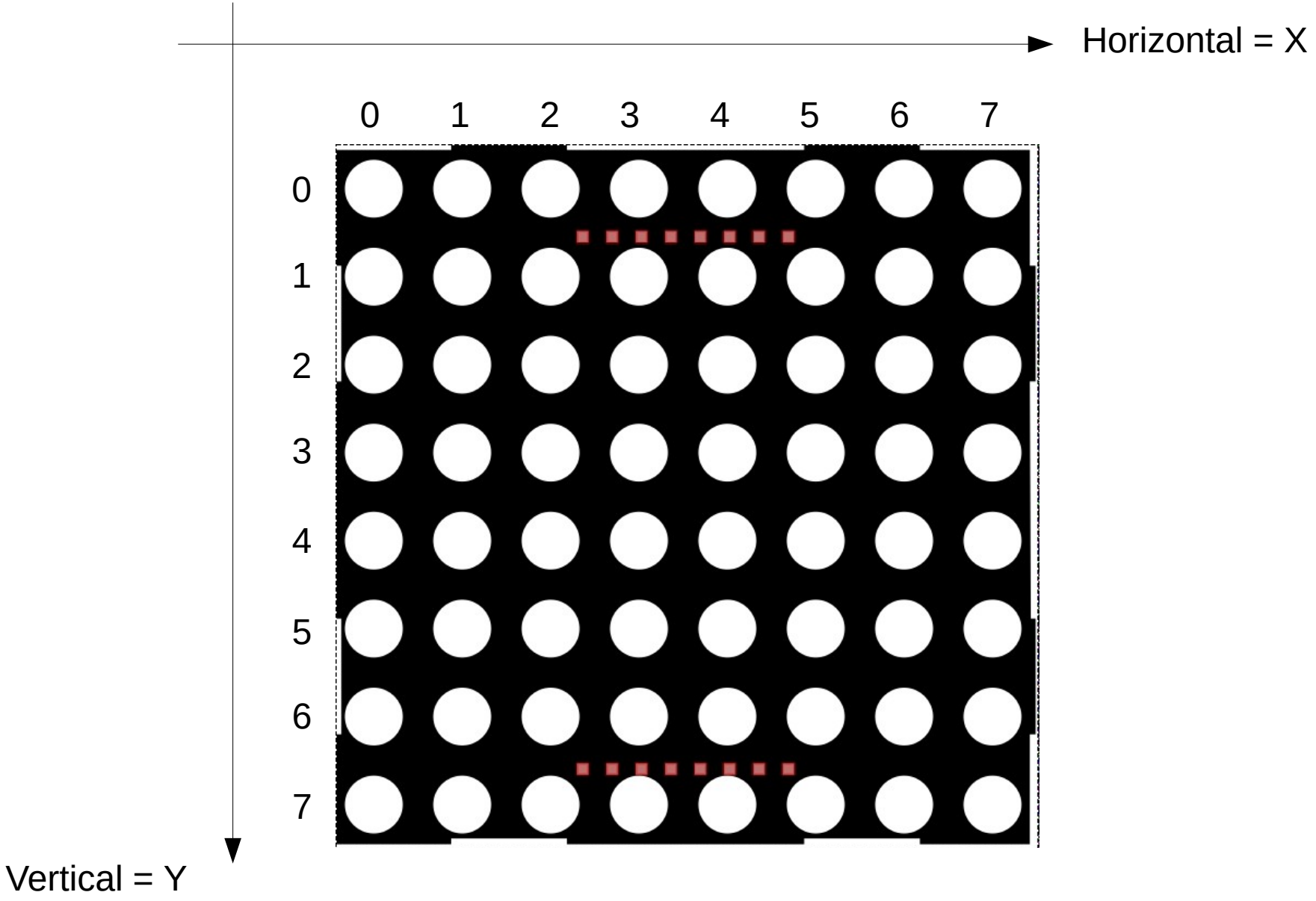
When you forget...don’t worry


```
display.fill(0)
display.show()
display.pixel(posx,posy,color)
display.text("a letter",posx,posy,color)
display.hline(posx,posy,length,color)
display.vline(posx,posy,length,color)
display.line(posx,posy,posx1,posx2,color)
display.rect(posx,posy,width,length,color)
display.fill_rect(posx,posy,width,length,color)
display.scroll(posx,posy)
display.scroll_left("anything you want to write")
display.scroll_right("anything you want to write")
display.rollup("a letter")
display.rolldown("a letter")
```

When you forget...don’t worry

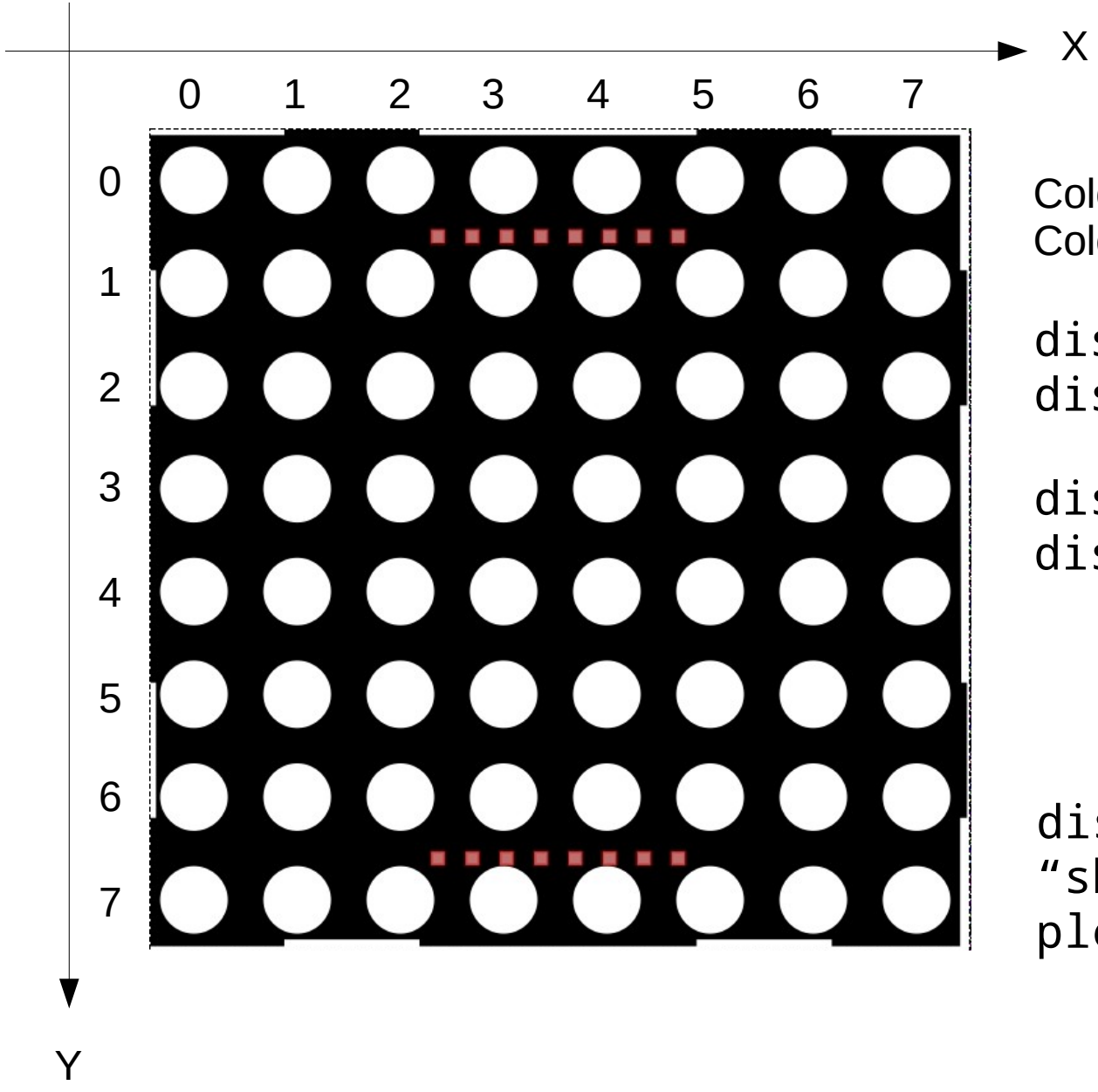

```
copy this empty matrix template and paste
matrix=[
    [0,0,0,0,0,0,0,0],
    [0,0,0,0,0,0,0,0],
    [0,0,0,0,0,0,0,0],
    [0,0,0,0,0,0,0,0],
    [0,0,0,0,0,0,0,0],
    [0,0,0,0,0,0,0,0],
    [0,0,0,0,0,0,0,0],
    [0,0,0,0,0,0,0,0]]
```

Led Matrix 8x8 = 64 individual led



Fill Color

Fill all led with same color



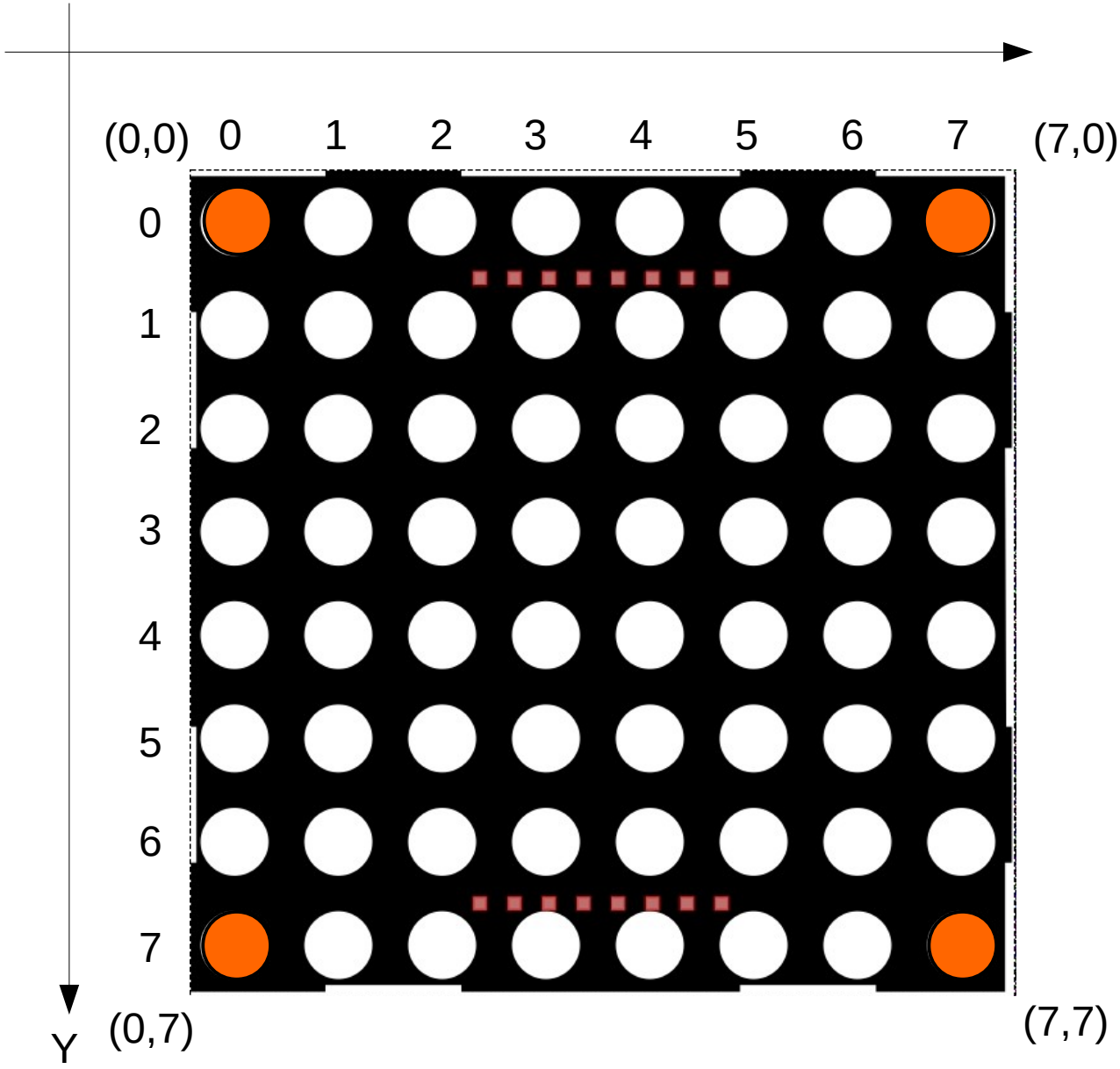
Color value 1 = Led ON
Color value 0 = Led OFF

```
display.fill(1)
display.show()

display.fill(0)
display.show()
```

```
display.show() means
“show it on the display
please”
```

SHELL

Led Matrix 8x8 – Led Location



Every led has “address” or “location”

Location is indicated by (X,Y) position or coordinate

X first, then Y second
(0,0) = top left corner
(7,0) = top right corner
(0,7)= bottom left corner
(7,7) = bottom right corner

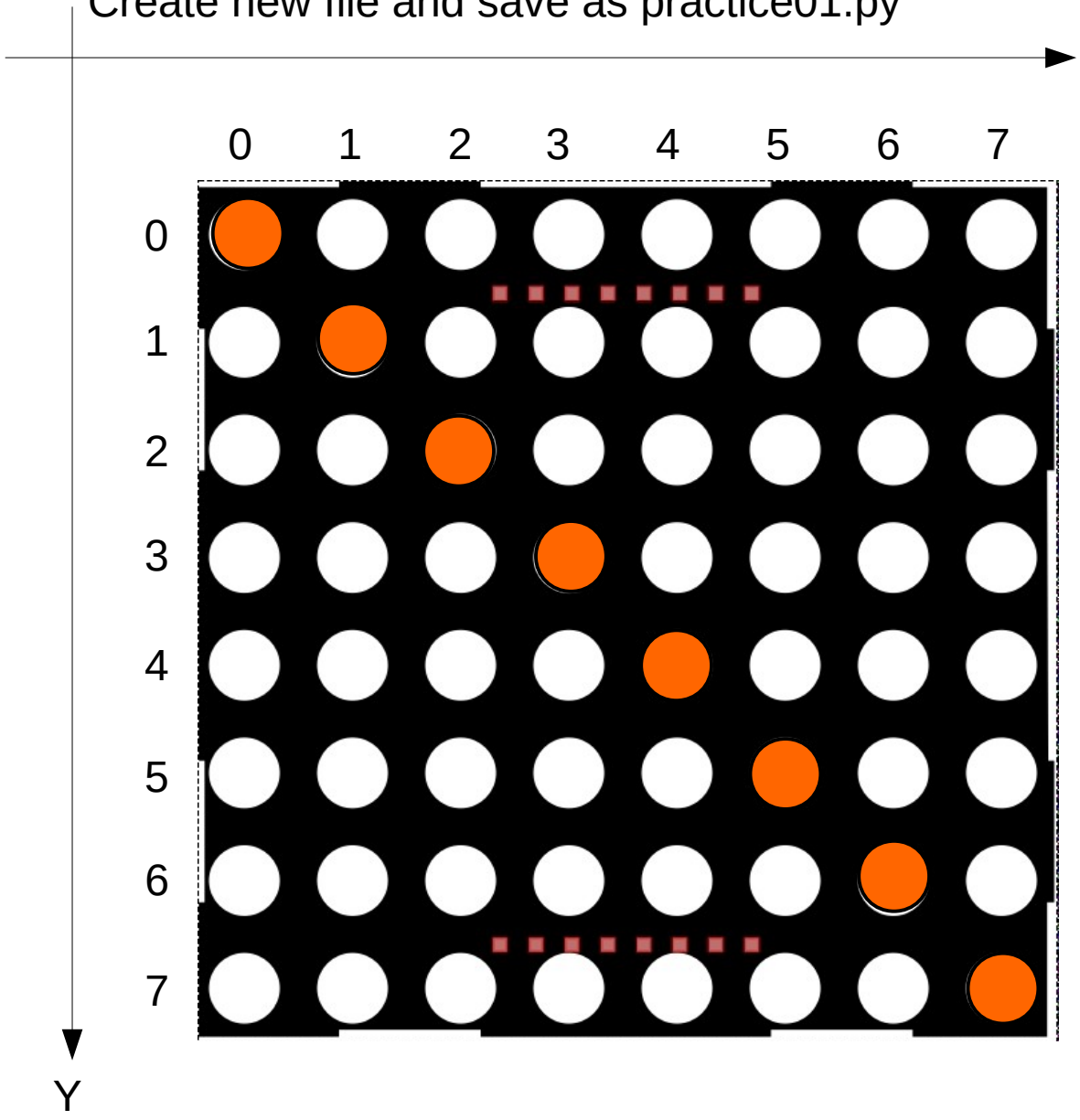
Remember:
(X,Y)

```
display.pixel(x,y,color)

display.pixel(0,0,1)
display.pixel(7,0,1)
display.pixel(0,7,1)
display.pixel(7,7,1)
display.show()
```

EDITOR

Practice #1 – Draw Diagonal Line

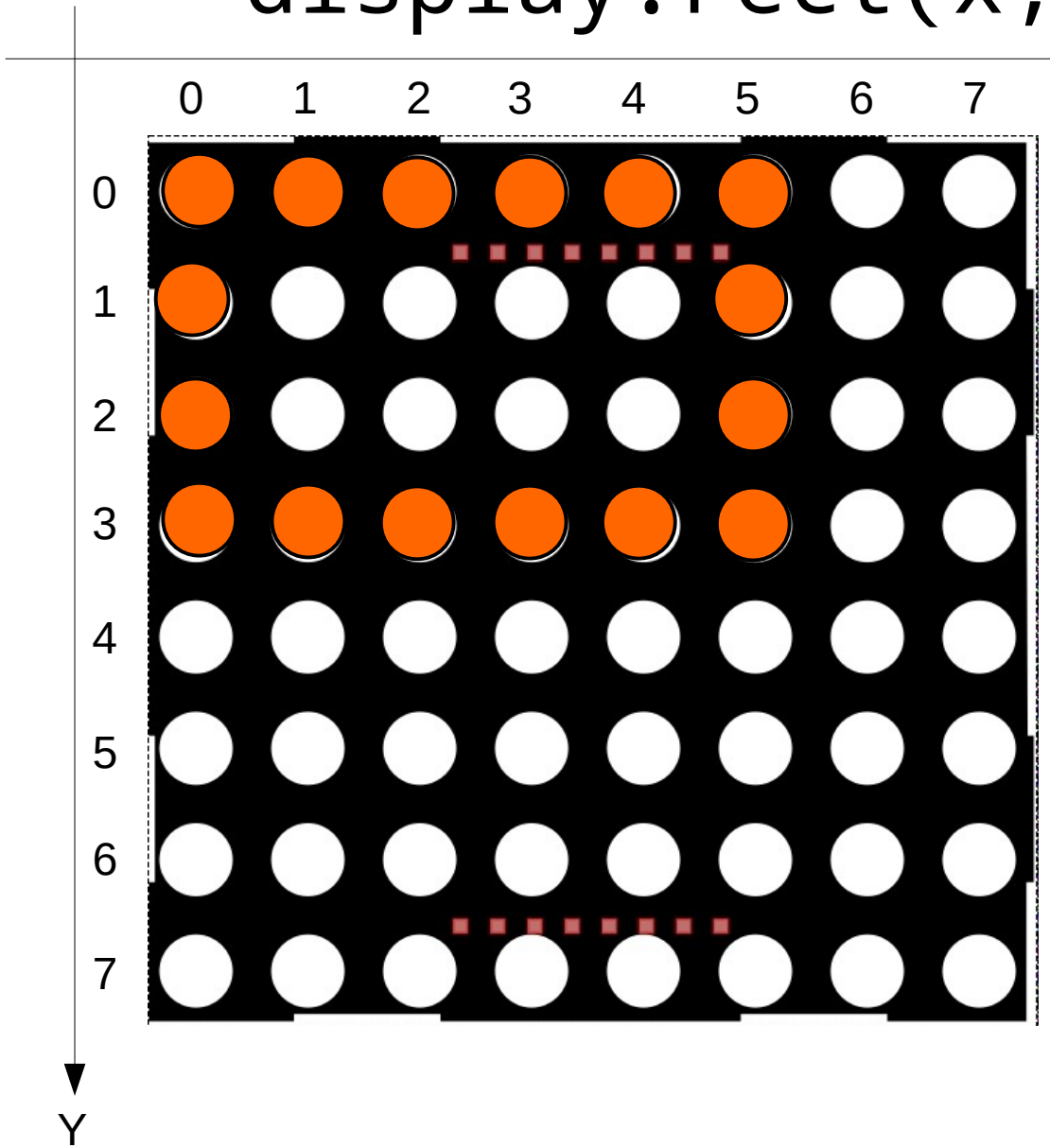
Create new file and save as practice01.py




```
display.pixel(0,0,1)
display.pixel(1,1,1)
display.pixel(2,2,1)
display.pixel(3,3,1)
display.pixel(4,4,1)
display.pixel(5,5,1)
display.pixel(6,6,1)
display.pixel(7,7,1)
display.show()
```

Draw Rectangle

```
display.rect(x,y,w,l,color)
```

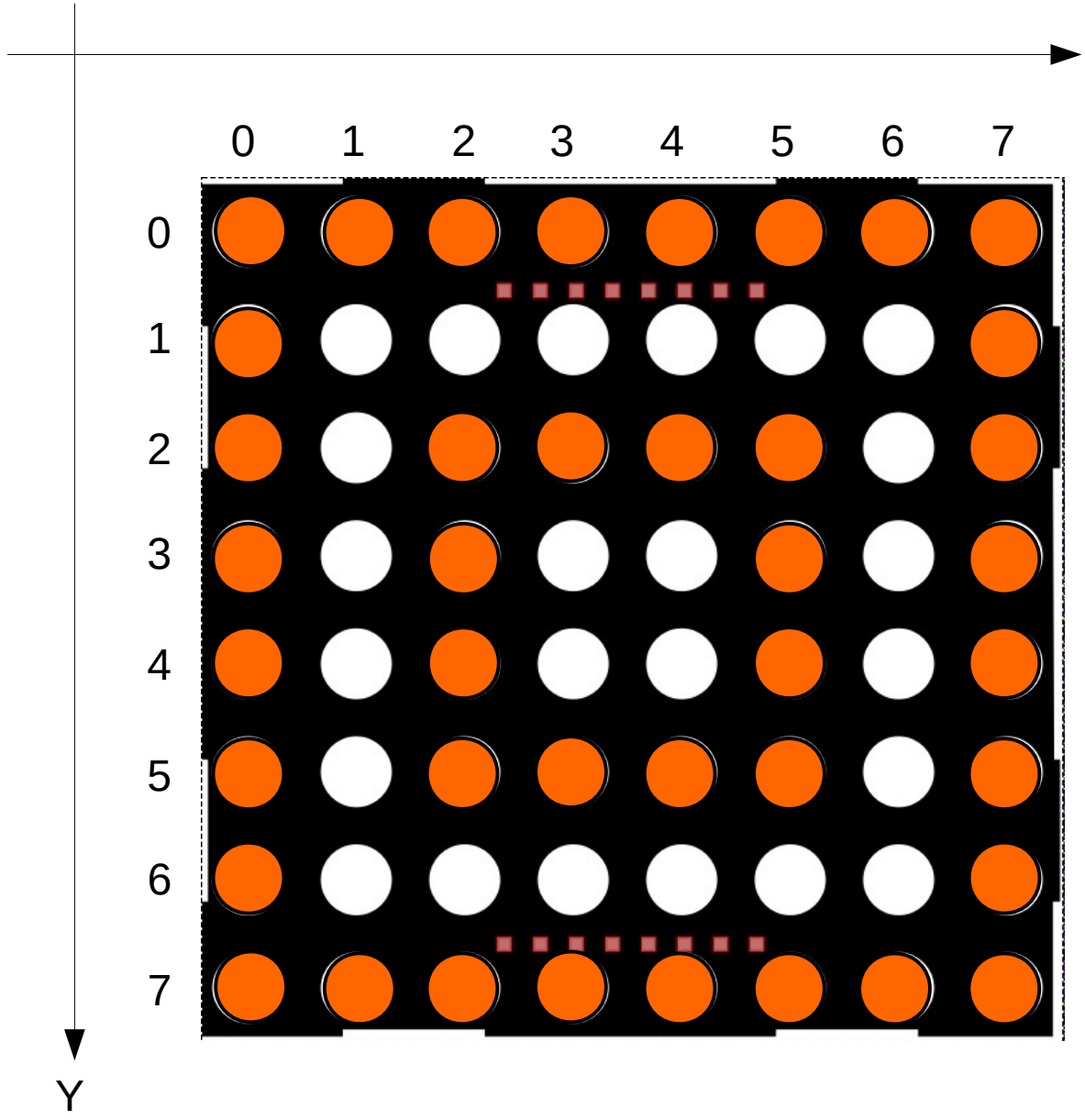



```
x = Position X
y = Position Y
w = width
l = length
color = 1 (ON),
color = 0 (OFF)

display.rect(0,0,6,4,1)
display.show()
```

EDITOR

Practice #2 – Draw Rectangle

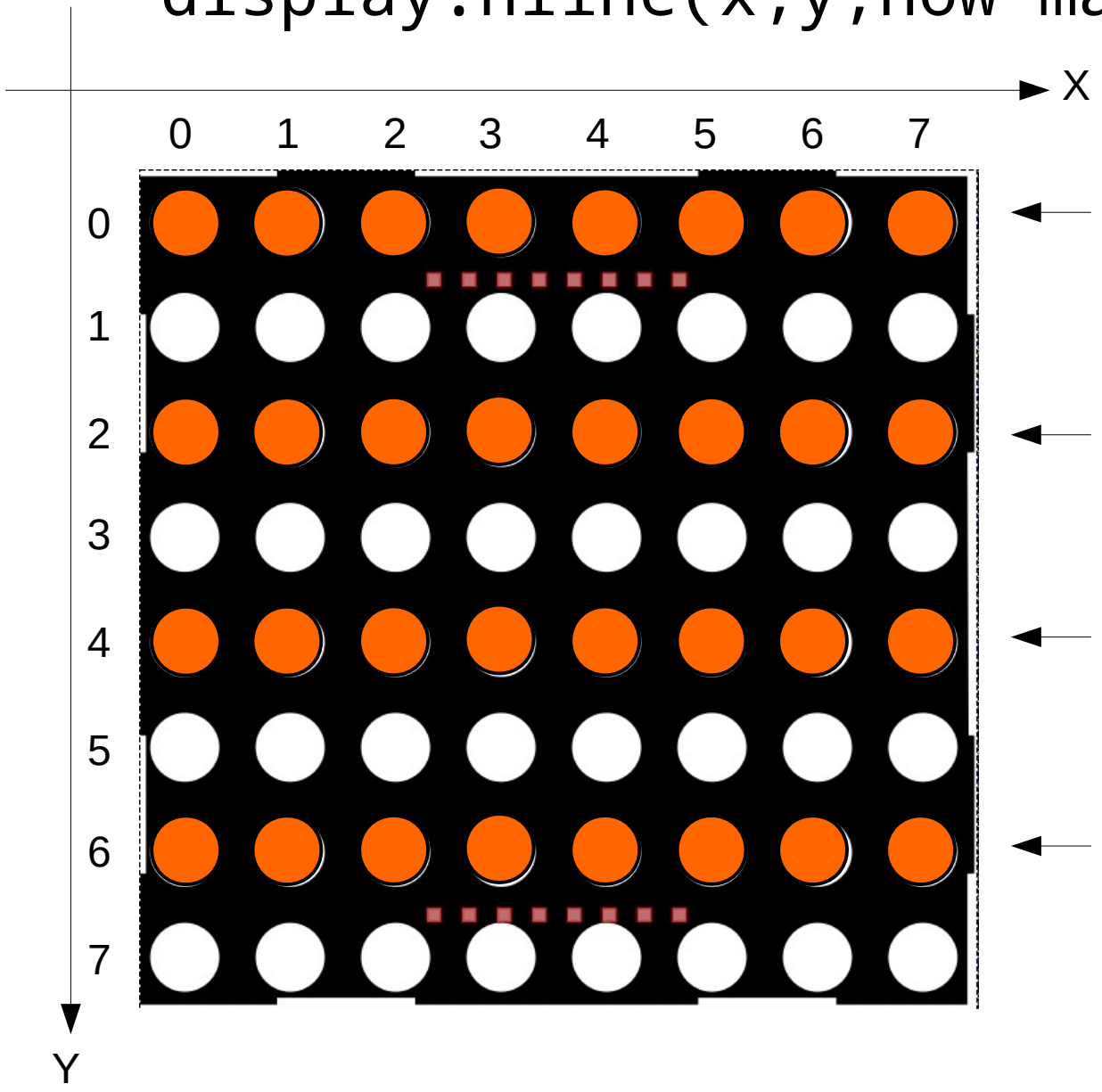
Create new file and save as practice02.py




```
display.rect(0,0,8,8,1)
display.rect(2,2,4,4,1)
display.show()
```

Draw Horizontal Line

```
display.hline(x,y,how many pixels,color)
```




```
display.hline(0,0,8,1)
display.show()
```

```
display.hline(0,2,8,1)
display.show()
```

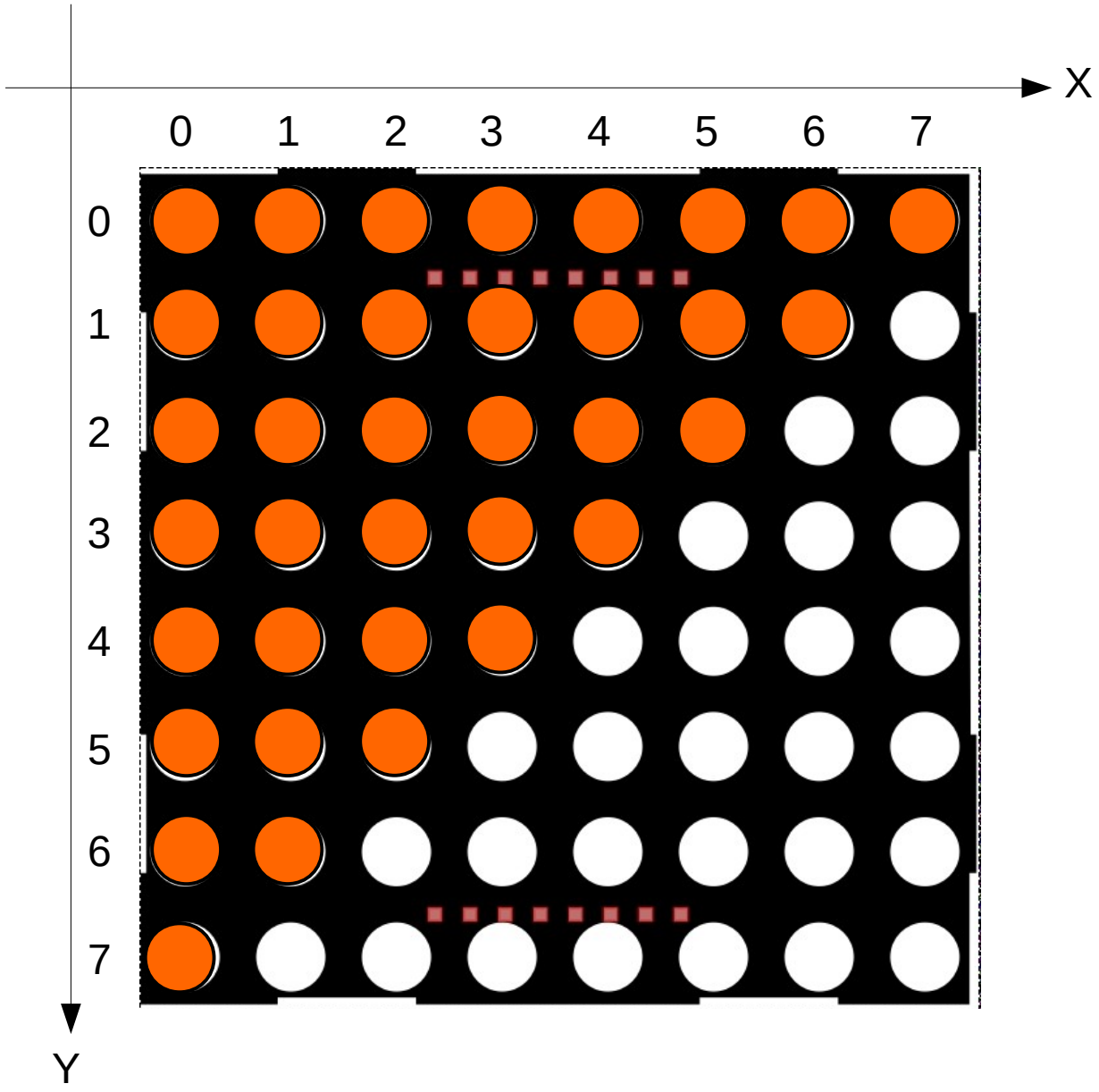
```
display.hline(0,4,8,1)
display.show()
```

```
display.hline(0,6,8,1)
display.show()
```

EDITOR

Practice #3 – Horizontal Lines

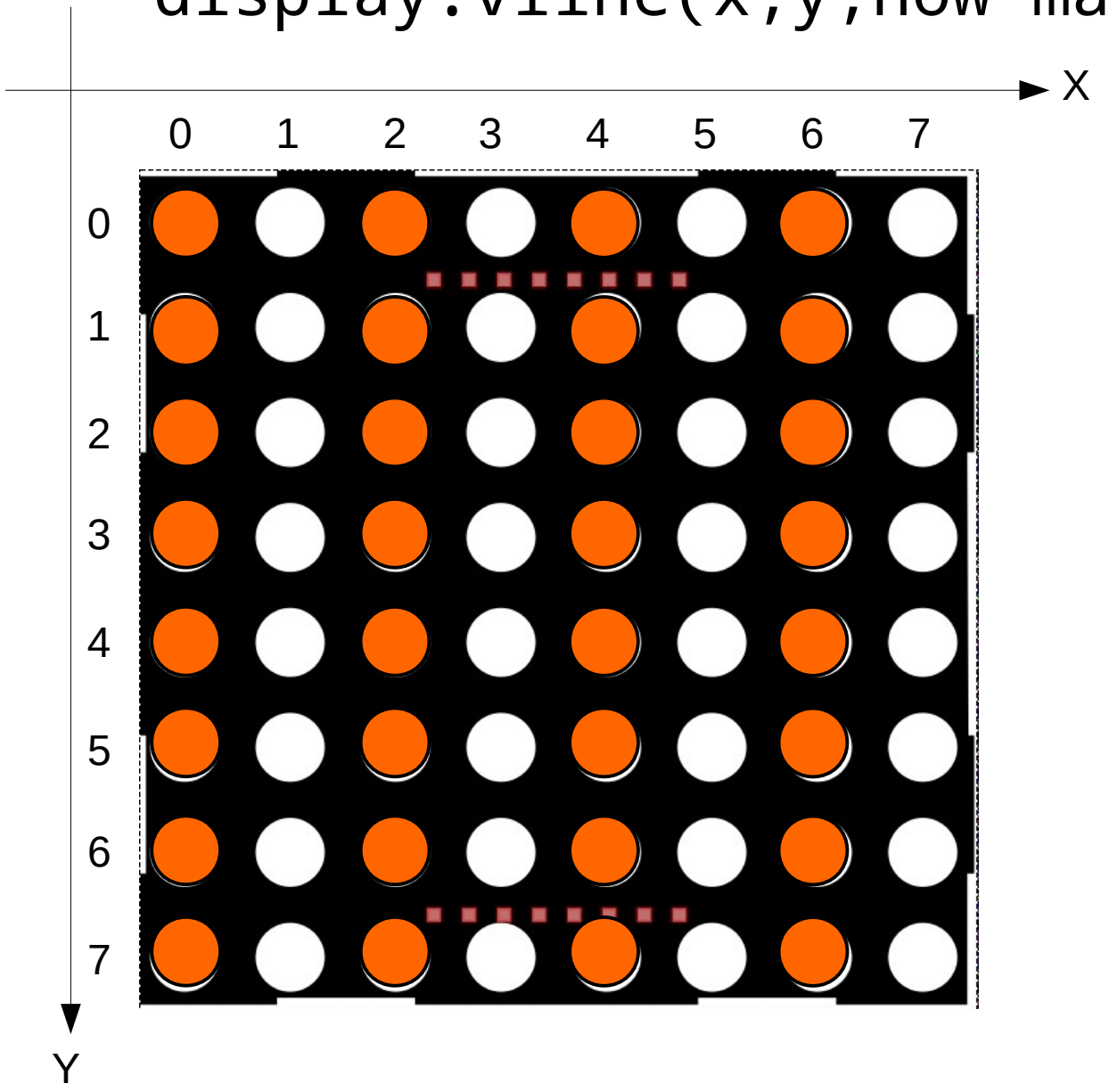
Create new file and save as practice03.py




```
display.hline(0,0,8,1)
display.hline(0,1,7,1)
display.hline(0,2,6,1)
display.hline(0,3,5,1)
display.hline(0,4,4,1)
display.hline(0,5,3,1)
display.hline(0,6,2,1)
display.hline(0,7,1,1)
display.show()
```

Draw Vertical Line

```
display.vline(x,y,how many pixels,color)
```




```
display.vline(0,0,8,1)
display.show()
```

```
display.vline(2,0,8,1)
display.show()
```

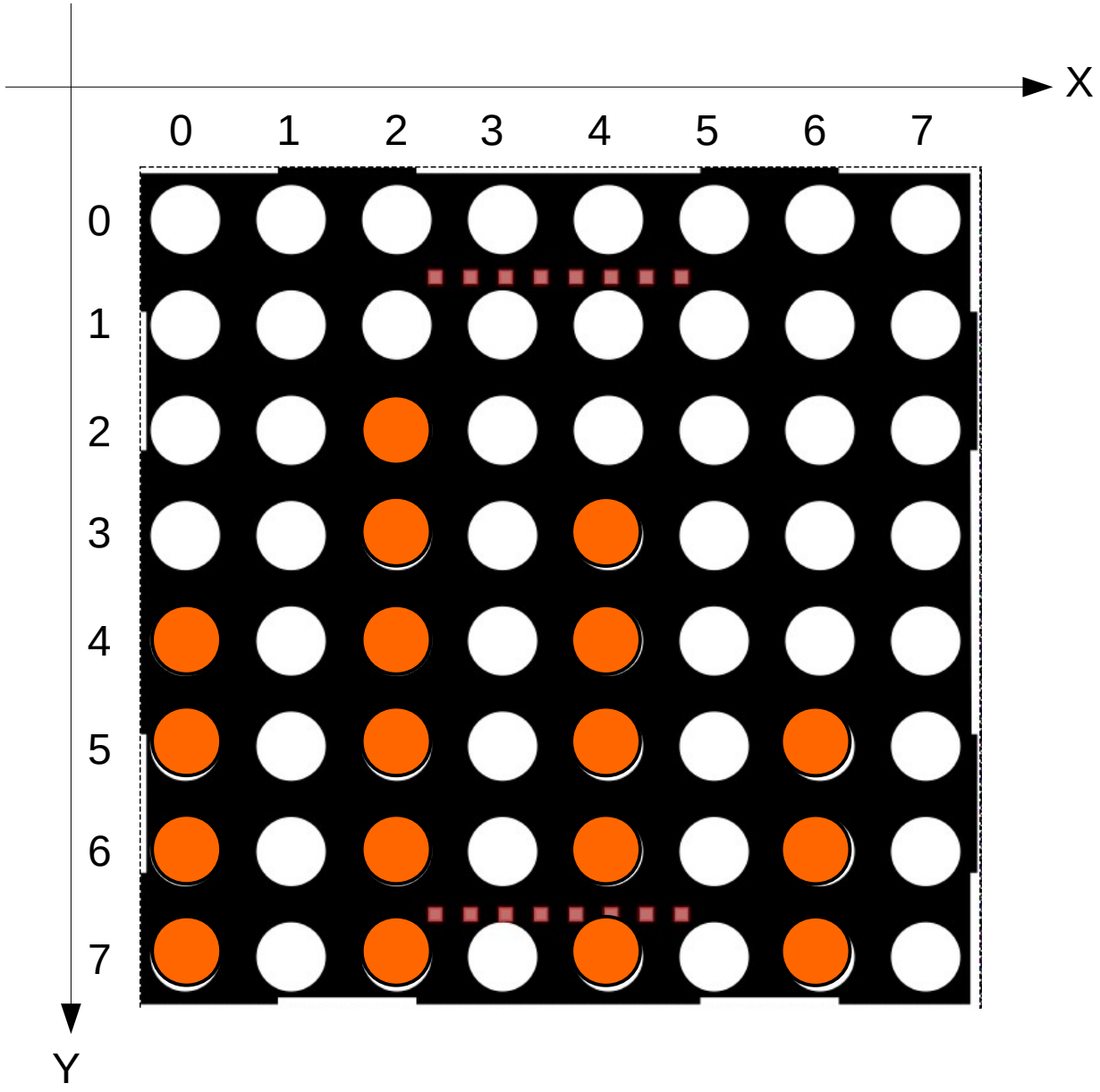
```
display.vline(4,0,8,1)
display.show()
```

```
display.vline(6,0,8,1)
display.show()
```

EDITOR

Practice #4 Vertical Line

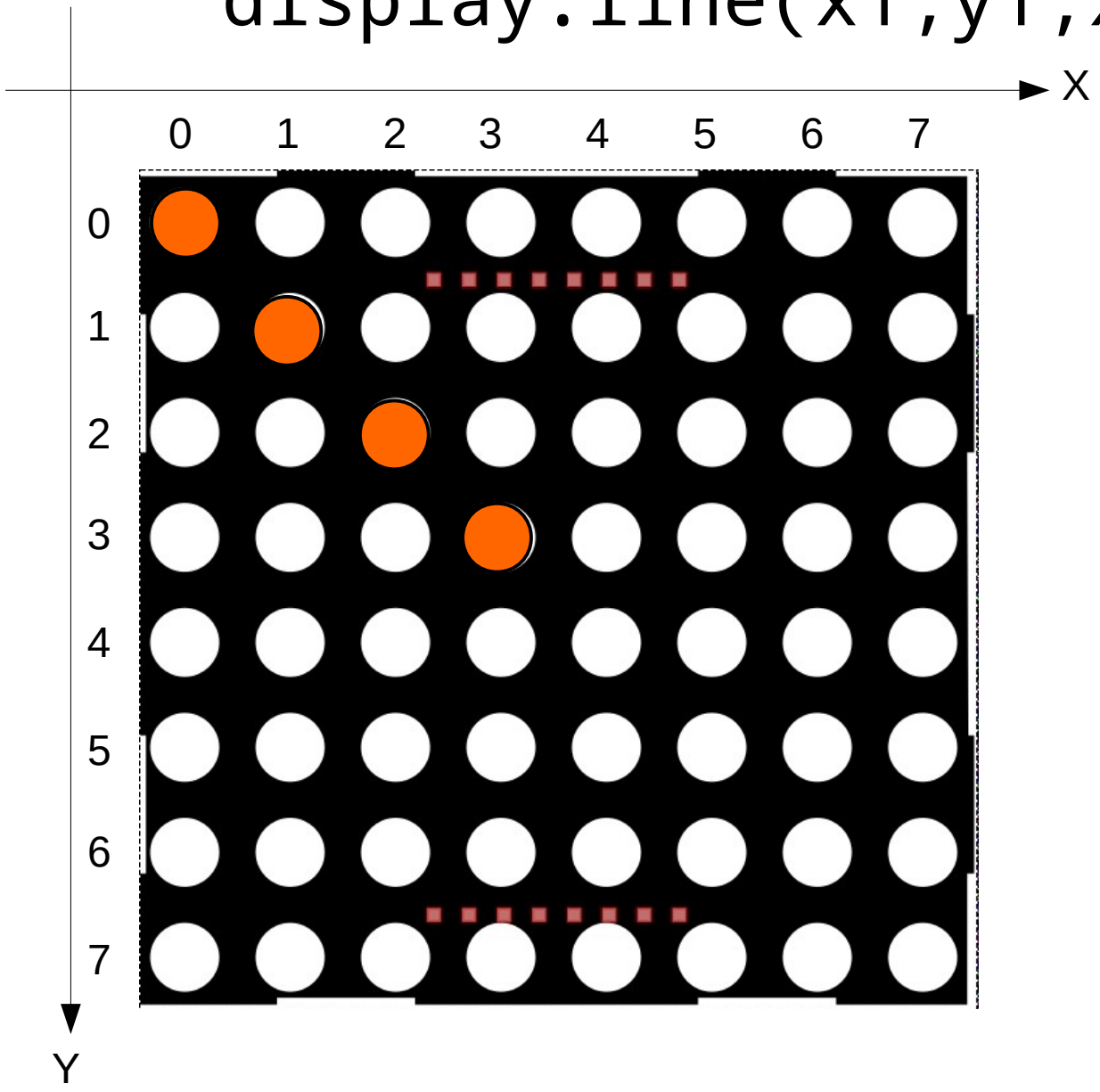
Create new file and save as practice04.py




```
display.vline(0,4,4,1)
display.vline(2,2,6,1)
display.vline(4,3,5,1)
display.vline(6,5,3,1)
display.show()
```

Draw Line

```
display.line(x1,y1,x2,y2,color)
```

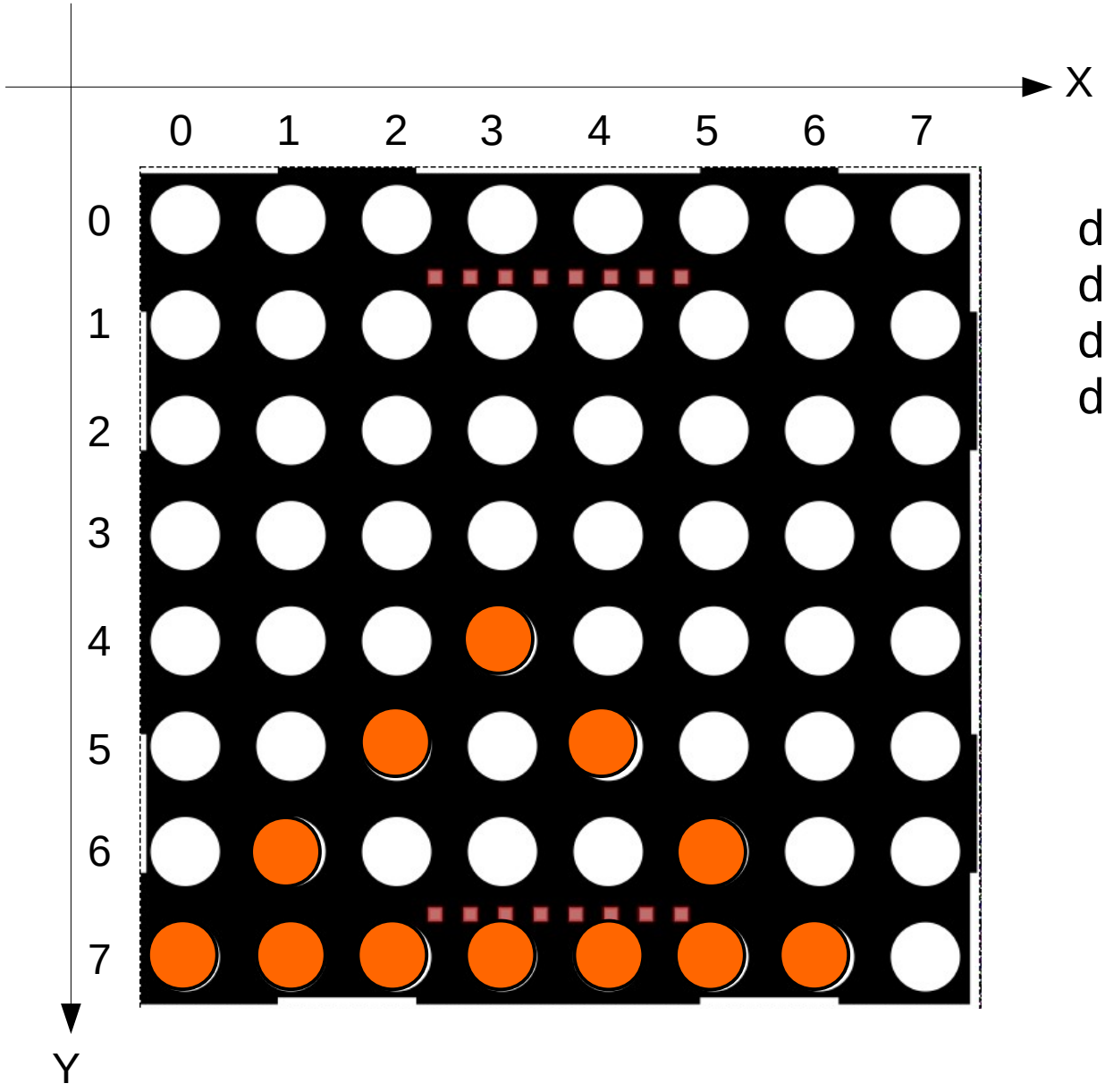



```
display.line(0,0,3,3,1)
display.show()
```

EDITOR

Practice #5

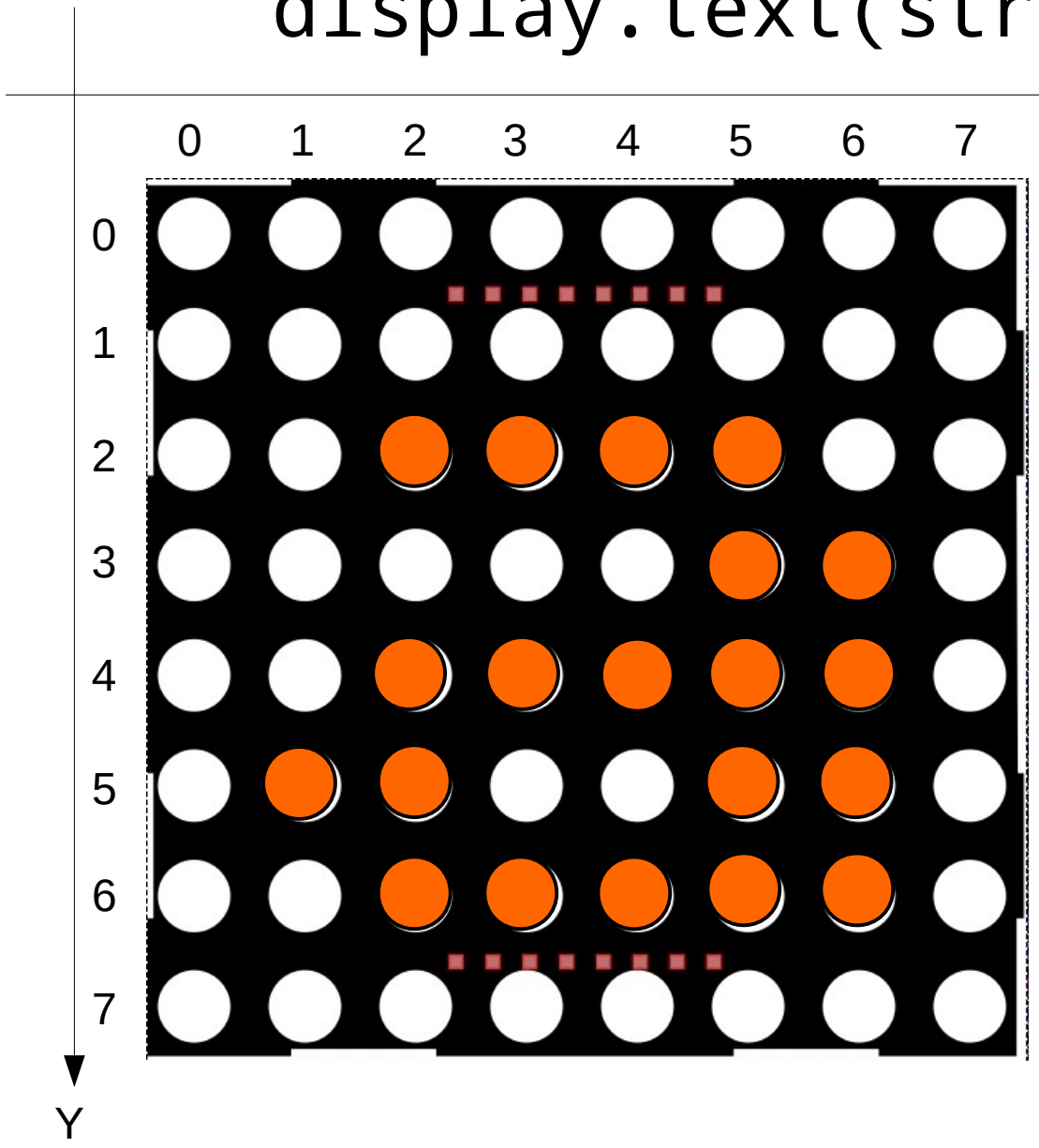
Create new file and save as practice05.py




```
display.line(0,7,3,4,1)
display.line(3,4,6,7,1)
display.line(0,7,6,7,1)
display.show()
```

Draw Text

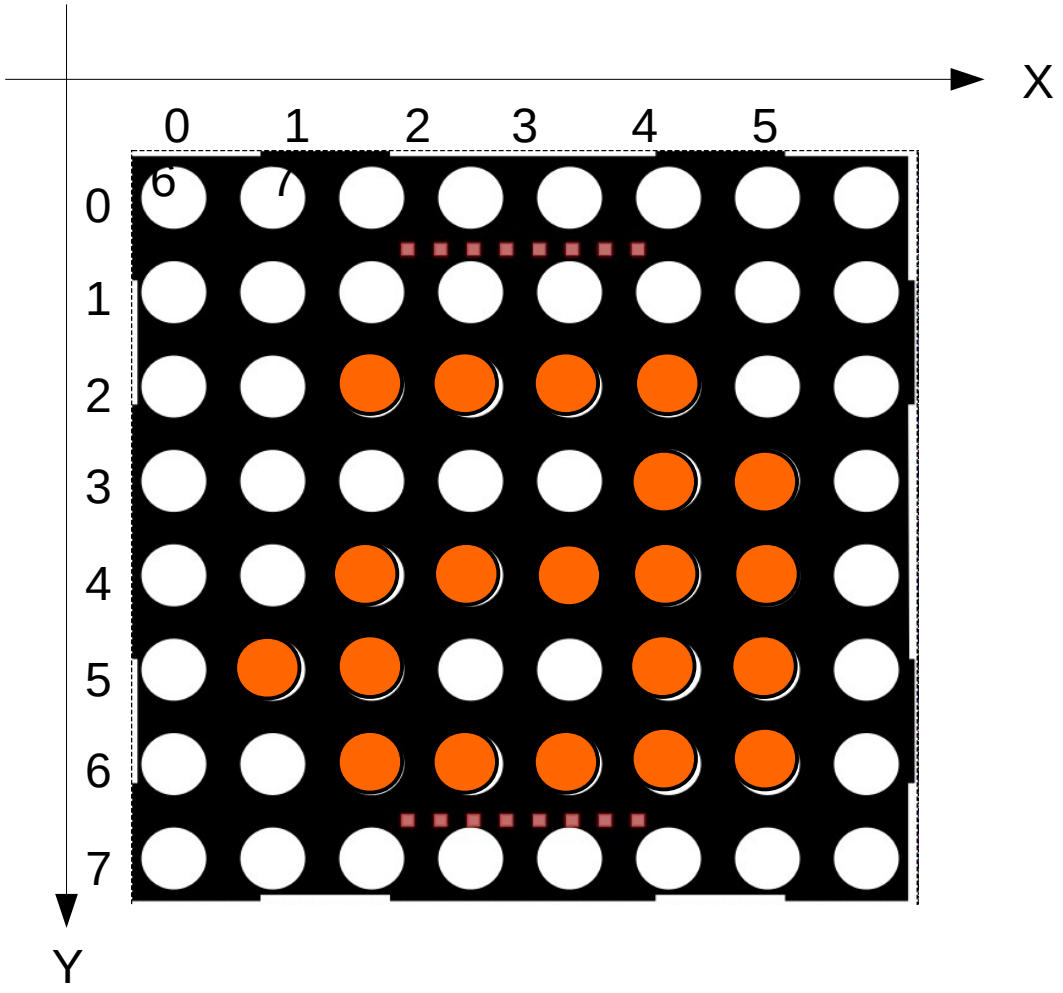
```
display.text(string,x,y,color)
```




```
display.text(“a”,0,0,1)
display.show()

display.text(“5”,0,0,1)
display.show()

display.text(“X”,0,0,1)
display.show()

display.text(“M”,0,0,1)
display.show()
```

Text Animation – Roll Up and Down

```
display.rollup(string)
display.rolldown(string)
```

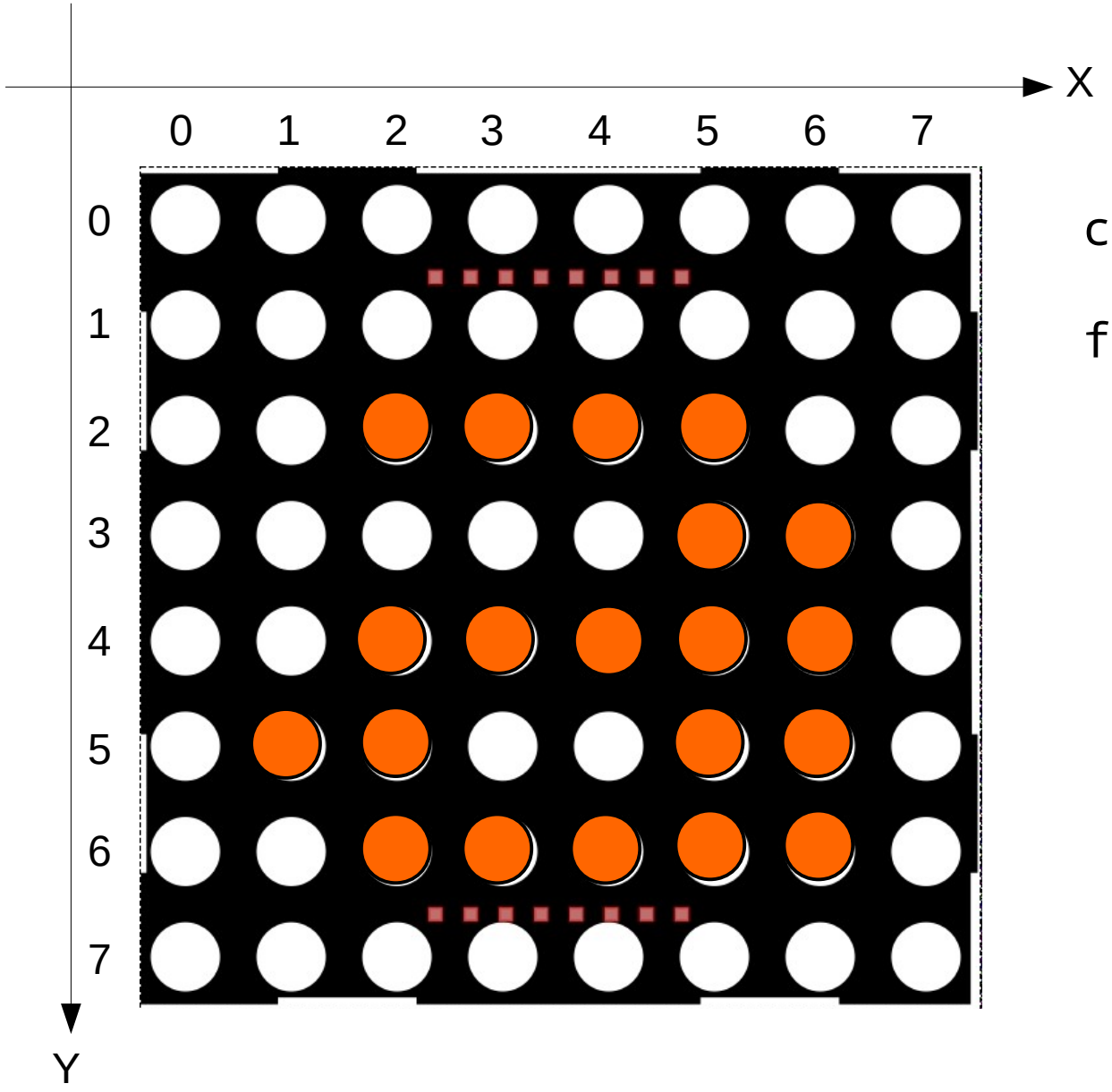



```
display.rollup(“a”)

display.rolldown(“a”)
```

for 8x8 led matrix,
one letter only

EDITOR

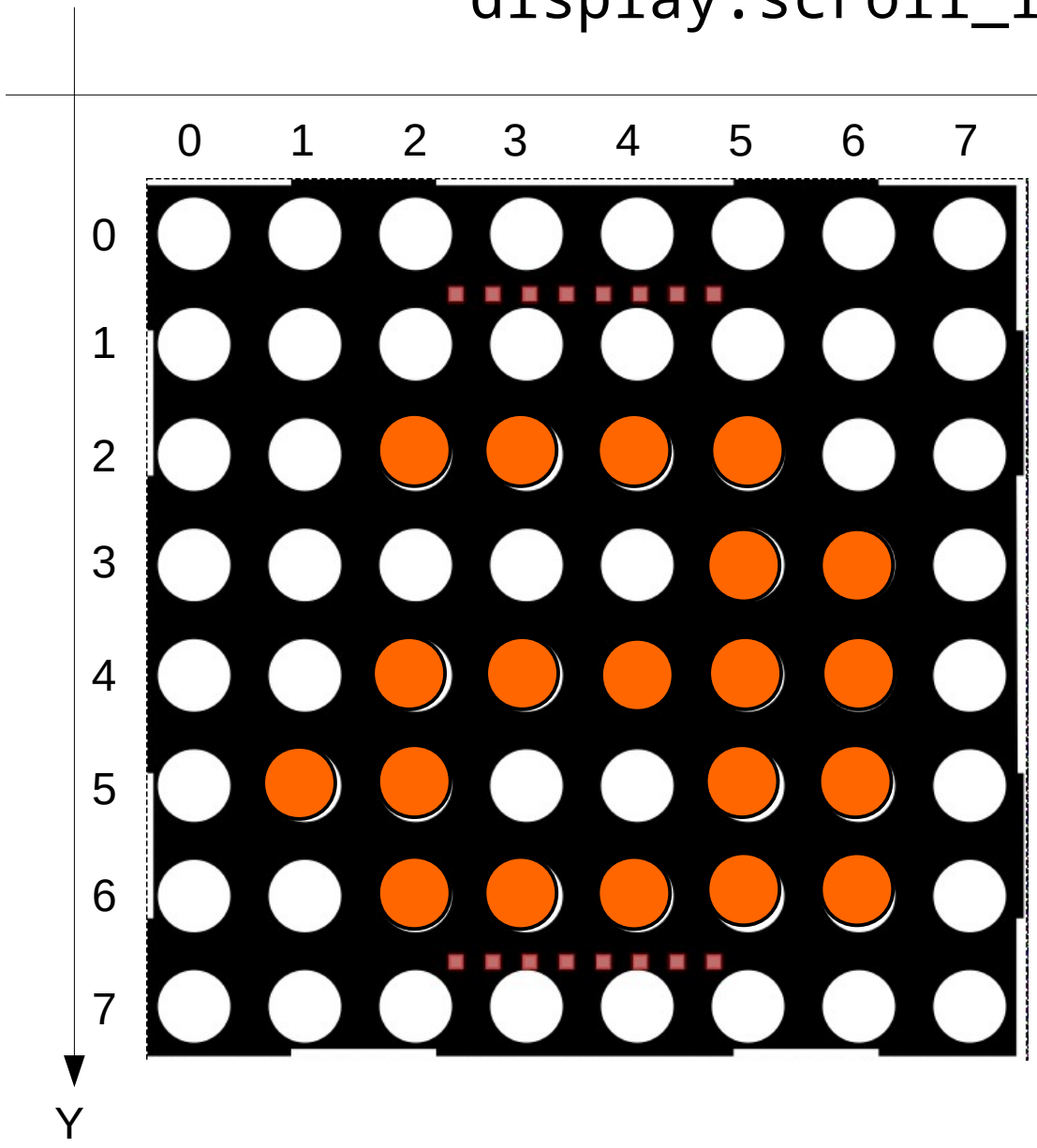
Text Animation – Roll Up




```
chicken=”KFC”

for i in chicken:
    display.rollup(i)
```

Text Animation – Scroll Left and Right

```
display.scroll_left(string)
```



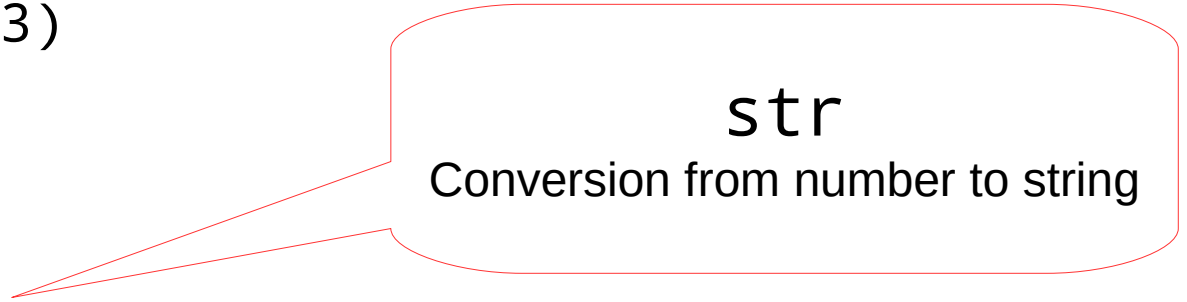

```
display.scroll_left(“hello”)

display.scroll_right(“hello”)
```

EDITOR

Practice #6 – Traffic Countdown

Create new file and save as practice06.py

```
import max7219
from machine import Pin, SPI,freq
import time
tile=1
freq(160000000)
spi=SPI(1, baudrate=10000000, polarity=0, phase=0)
display=max7219.Matrix8x8(spi, Pin(2), tile)
display.brightness(3)
display.fill(0)
display.show()

for i in range(10):
   display.rollup(str(9-i))

display.scroll_left("silahkan jalan")
```

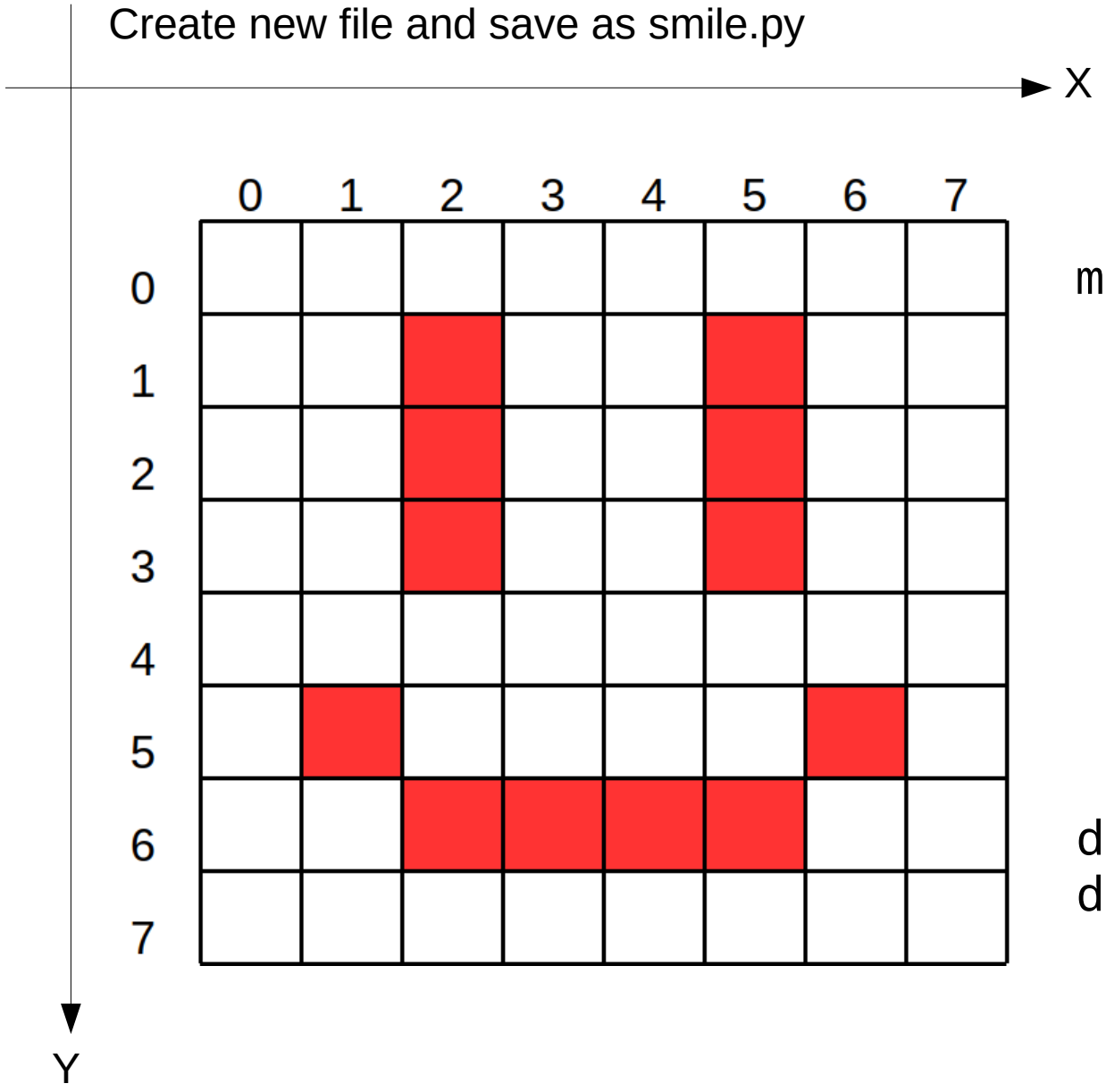


EDITOR

Pixel Art

`display.draw(matrix)`

Create new file and save as smile.py



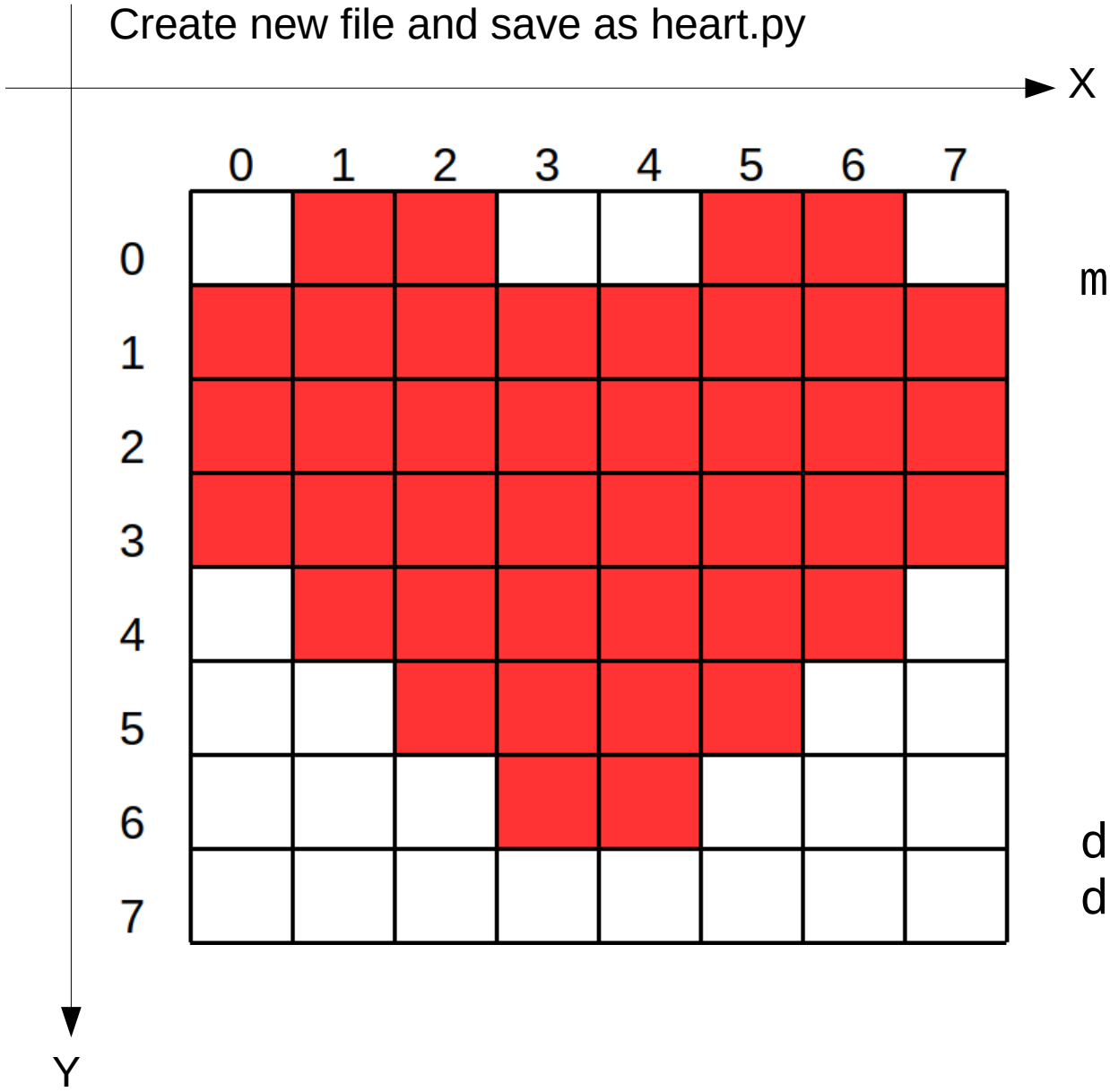

```
matrix=[
    [0,0,0,0,0,0,0,0],
    [0,0,1,0,0,1,0,0],
    [0,0,1,0,0,1,0,0],
    [0,0,1,0,0,1,0,0],
    [0,0,0,0,0,0,0,0],
    [0,1,0,0,0,0,1,0],
    [0,0,1,1,1,1,0,0],
    [0,0,0,0,0,0,0,0]]

display.draw(matrix)
display.show()
```

EDITOR

Pixel Art

`display.draw(matrix)`

Create new file and save as heart.py



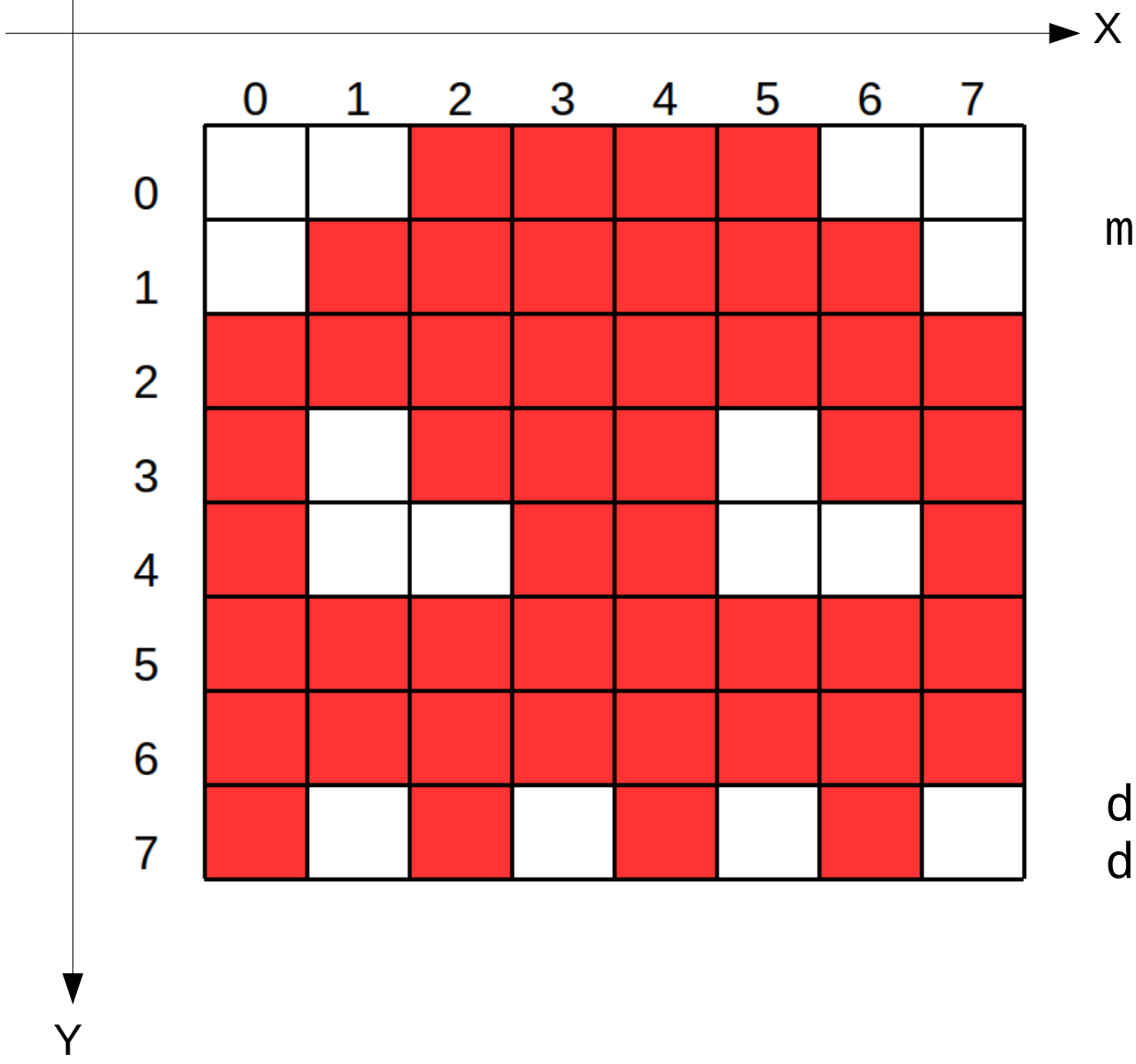

```
matrix=[
      [0,1,1,0,0,1,1,0],
      [1,1,1,1,1,1,1,1],
      [1,1,1,1,1,1,1,1],
      [1,1,1,1,1,1,1,1],
      [0,1,1,1,1,1,1,0],
      [0,0,1,1,1,1,0,0],
      [0,0,0,1,1,0,0,0],
      [0,0,0,0,0,0,0,0]]

display.draw(matrix)
display.show()
```

EDITOR

Pixel Animation

`display.draw(matrix)`

Create new file and save as animate.py

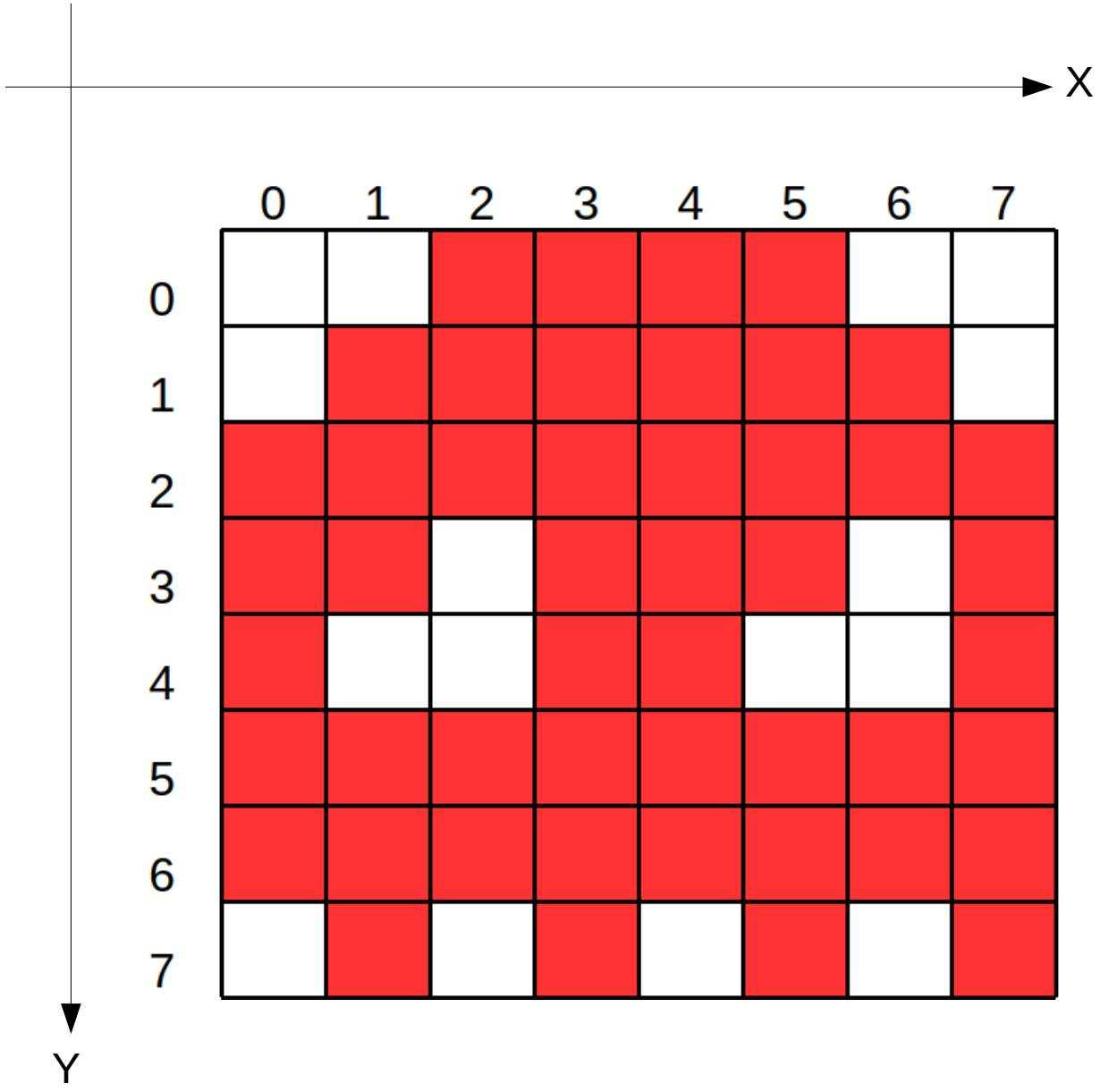



```
matrix1=[
     [0,0,1,1,1,1,0,0],
     [0,1,1,1,1,1,1,0],
     [1,1,1,1,1,1,1,1],
     [1,0,1,1,1,0,1,1],
     [1,0,0,1,1,0,0,1],
     [1,1,1,1,1,1,1,1],
     [1,1,1,1,1,1,1,1],
     [1,0,1,0,1,0,1,0]]

display.draw(matrix1)
display.show()
```

EDITOR

Pixel Animation

`display.draw(matrix)`




```
matrix2=[
      [0,0,1,1,1,1,0,0],
      [0,1,1,1,1,1,1,0],
      [1,1,1,1,1,1,1,1],
      [1,1,0,1,1,1,0,1],
      [1,0,0,1,1,0,0,1],
      [1,1,1,1,1,1,1,1],
      [1,1,1,1,1,1,1,1],
      [0,1,0,1,0,1,0,1]]

display.draw(matrix2)
display.show()
```

EDITOR

Pixel Animation

```
display.draw(matrix)
```

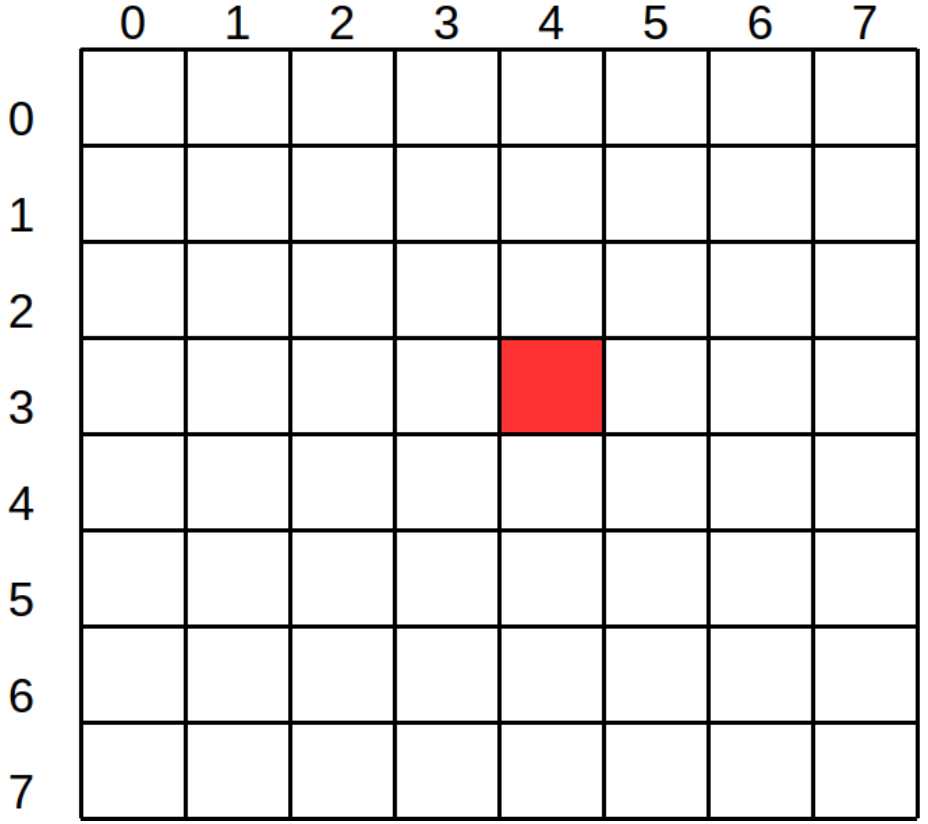
```
matrix1=[
    [0,0,1,1,1,1,0,0],
    [0,1,1,1,1,1,1,0],
    [1,1,1,1,1,1,1,1],
    [1,0,1,1,1,0,1,1],
    [1,0,0,1,1,0,0,1],
    [1,1,1,1,1,1,1,1],
    [1,1,1,1,1,1,1,1],
    [1,0,1,0,1,0,1,0]]

matrix2=[
    [0,0,1,1,1,1,0,0],
    [0,1,1,1,1,1,1,0],
    [1,1,1,1,1,1,1,1],
    [1,1,0,1,1,1,0,1],
    [1,0,0,1,1,0,0,1],
    [1,1,1,1,1,1,1,1],
    [1,1,1,1,1,1,1,1],
    [0,1,0,1,0,1,0,1]]
```

```
for i in range(10):
    display.fill(0)
    display.draw(matrix1)
    display.show()
    time.sleep(0.5)

    display.fill(0)
    display.draw(matrix2)
    display.show()
    time.sleep(0.5)
```

QUIZ

How to display one red dot on the display

```
display._____(X,Y,1)
display.____()
```



How to display one red dot from this picture

```
display._____(_,_,_)
display.____()
```

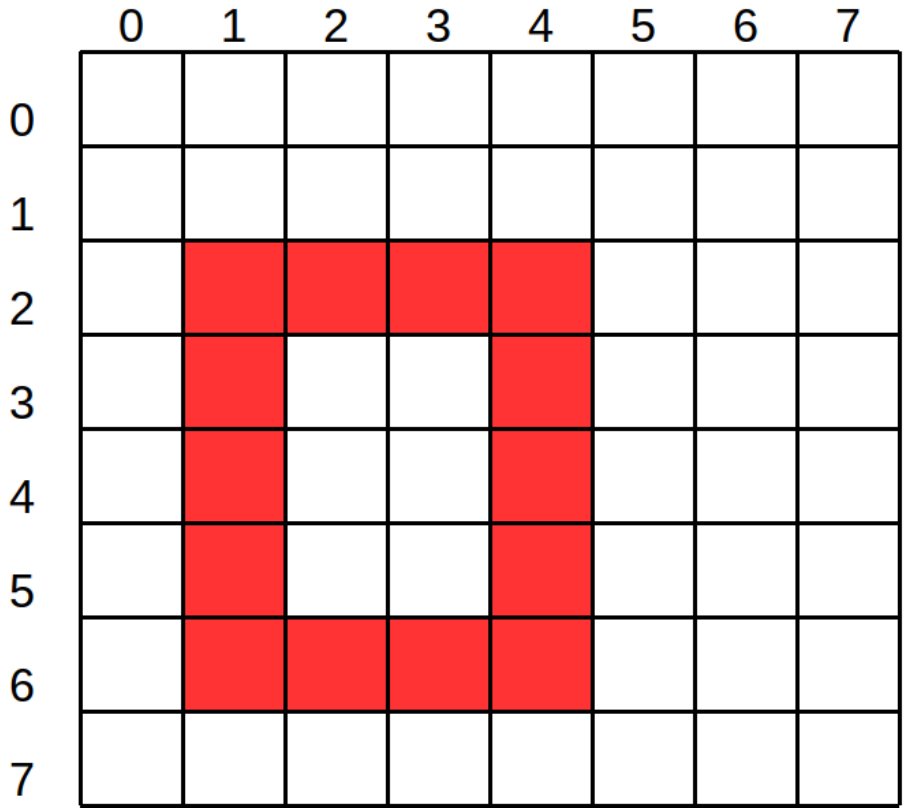
How to remove one red dot from this picture

```
display._____(_,_,_)
display.____()
```

QUIZ

How to draw rectangle on the display

```
display._______(X,Y,1)
display._____()
```



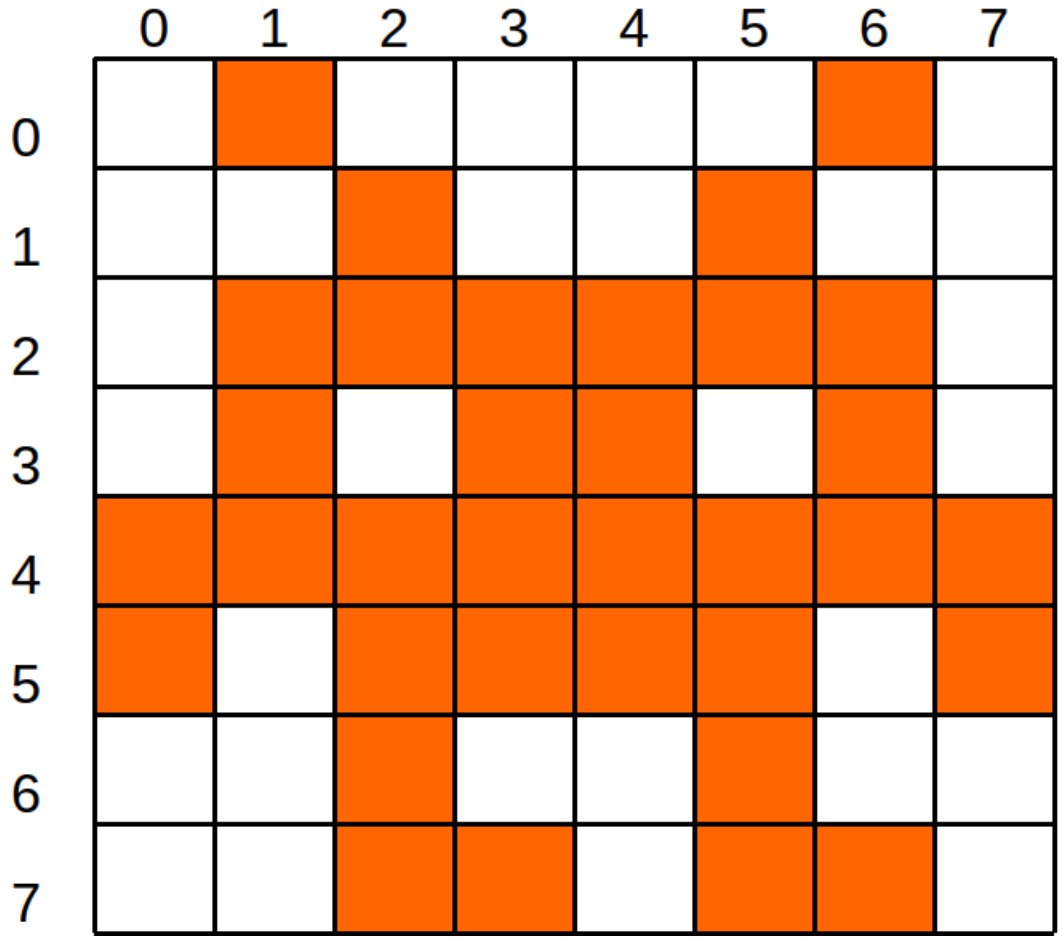
QUIZ

Scroll text “hello world” from right to left

A. Display.scroll_left(“hello world”)

- B. display.scroll_left(“hello world”)
- C. Display.scroll left(“hello world”)

QUIZ




```
matrix=[
    [0,0,0,0,0,0,0,0],
    [0,0,0,0,0,0,0,0],
    [0,0,0,0,0,0,0,0],
    [0,0,0,0,0,0,0,0],
    [0,0,0,0,0,0,0,0],
    [0,0,0,0,0,0,0,0],
    [0,0,0,0,0,0,0,0],
    [0,0,0,0,0,0,0,0]]

display._____(matrix)
display.show()
```