CS Bridge, Lecture 9

Graphics and Nested Loops

KOÇ UNIVERSITY

İZMİR DEMOKRASİ ÜNİVERSİTESİ 2016

LELAND STANFORD JUNIOR UNIVERSITY 1891

Creating Graphical Objects

canvas`.create_line(`x_0, y_0, x_1, y_1`)` Creates a new line connecting (x0, y0) and (x1, y1).
canvas`.create_rectangle(`x_0, y_0, x_1, y_1`)` Creates a new rectangle on the canvas the size of this bounding box.
canvas`.create_oval(`x_0, y_0, x_1, y_1`)` Creates a new oval on the canvas contained within this bounding box.
canvas`.create_text(`*x, y, text*`)` Creates text on the canvas with the specified contents, centered at (x, y).

canvas`.create_image(`*x, y, filepath*`)` Creates a new image on the canvas from the specified file, with top-left corner at (x, y).

Operations on Graphical Objects

canvas`.moveto(`*object, x, y*`)` Sets the location of obj to the specified coordinates.
canvas`.set_color(`*object, color*`)` Sets the outline and fill color (if applicable) of the object.
canvas`.set_outline_color(`*object, color*`)` Sets the outline color of the object.
canvas`.set_fill_color(`*object, color*`)` Sets the fill color of the object.
canvas`.set_font(`*object, font, size*`)` Sets the font and font size for the given text object.
canvas`.delete(`*object*`)` Deletes the object from the canvas

turkey21.csbridge.org/en/index.html

CS Bridge | Resources | Projects | Examples | Lectures | Bonus | Sponsors

References
Karel Reader
Karel Reference

Python Reader
Python Reference

Graphics Reference

Course Information
General Info
Code of Conduct
[Student] Course FAQs
Section Info

Software
Install PyCharm
How to use PyCharm

Submission/Office Hours

to Computer Science

21
o August 19th, online

There's not ge at the moment. Take a break, have some tea, and we'll see you at the next lecture

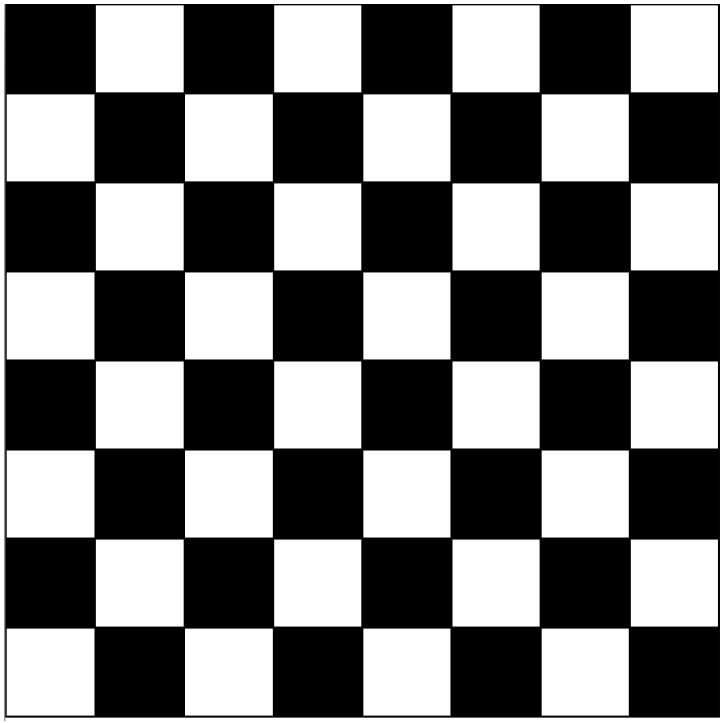
The Ide se

The point of t teach you the fundamentals of computer programming to the point where you can go a is taught by a collaboration of instructors from Stanford University, Koç University, an ity. You will learn to program using material from Stanford's Introduction to Computer Sci

Recap

Drawing with Loops

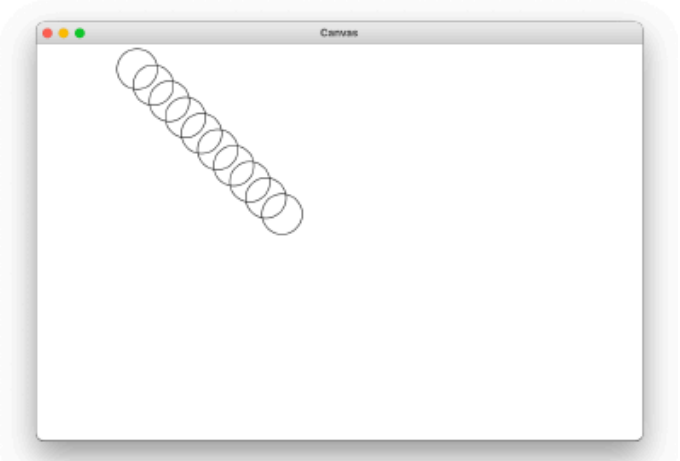
Goal for today: draw checker board **64 squares**



Drawing with Loops

```
def main():
    canvas = Canvas()
    for i in range(10):
        circle_x = 100 + 20 * i
        circle_y = 5 + 20 * i
        canvas.create_oval(circle_x, circle_y, circle_x + 50, circle_y + 50)
    canvas.mainloop()
```

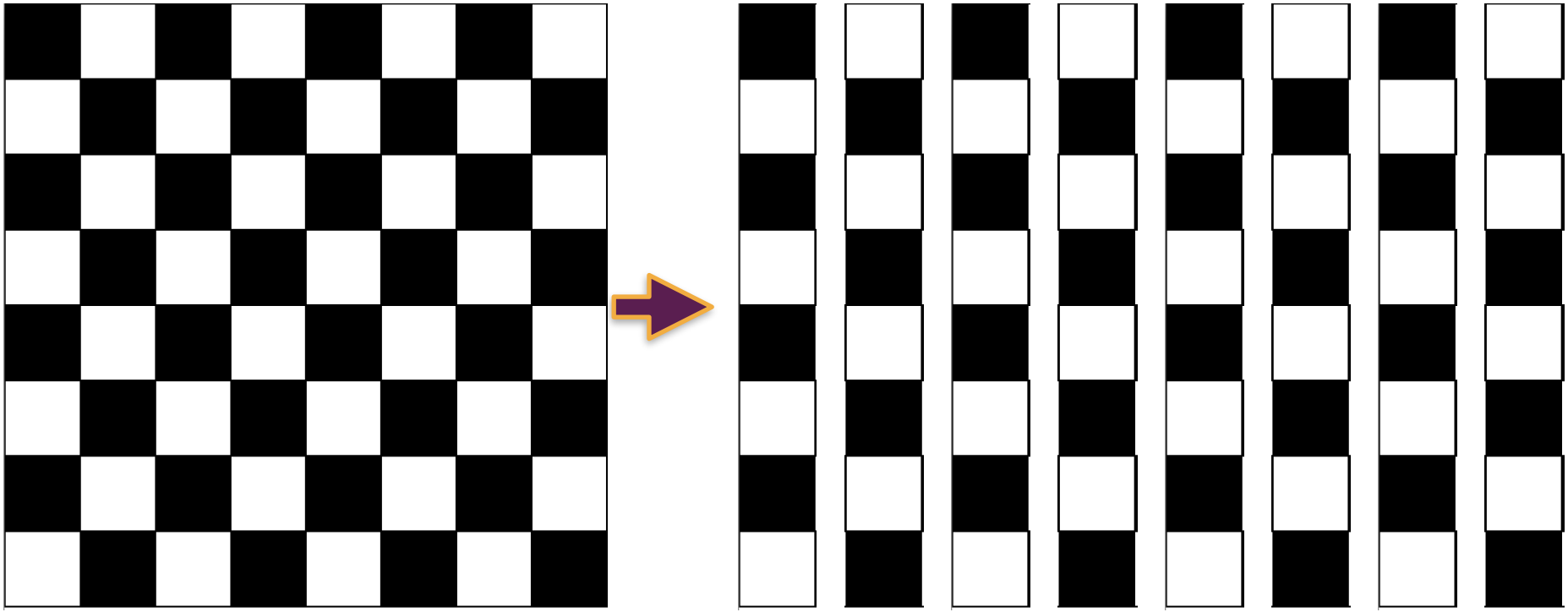
Loop variable is used to control location
Both circle_x and circle_y



Recap

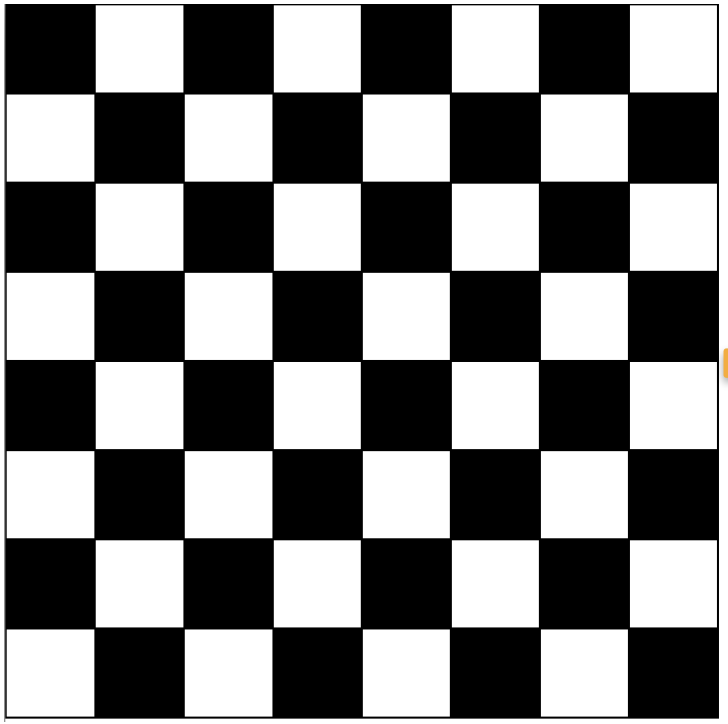
Drawing with Loops

Decomposition → **draw_column()**

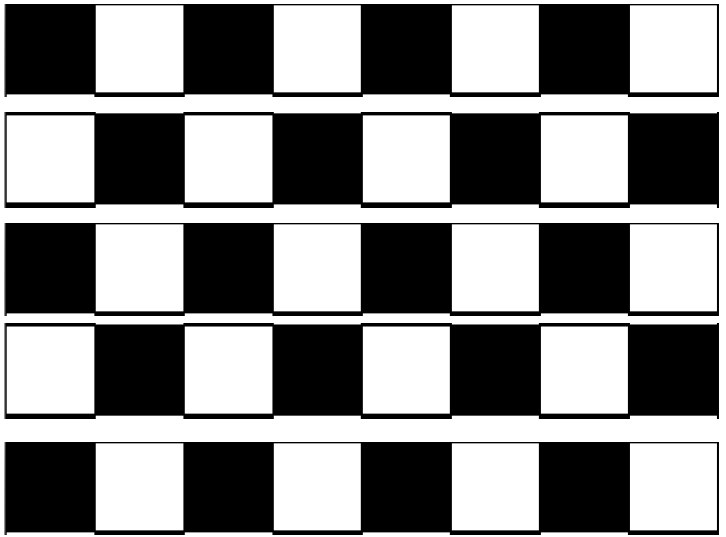


Drawing with Loops

Decomposition



draw_row()



...

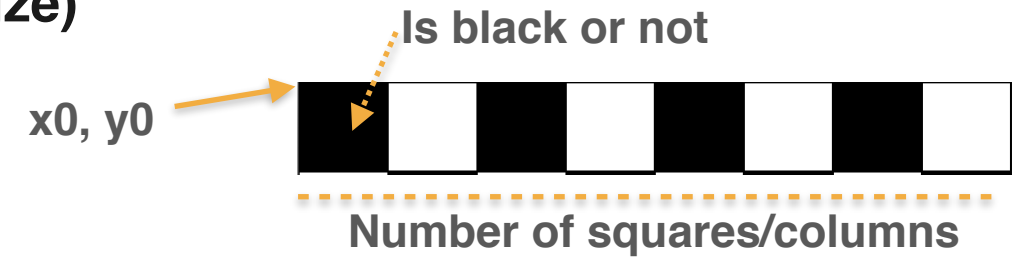
Drawing with Loops

Decomposition
draw_row()

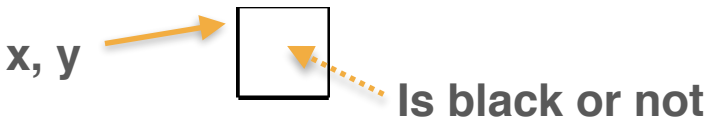
draw_square()

What should be the parameters of each of these functions?
(Assume constant square size)

draw_row(??)



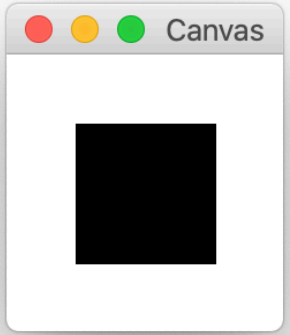
draw_square(??)



Draw square without using a function

```
from graphics import Canvas

SQUARE_SIZE = 60

def main():
    canvas = Canvas(120, 120)
    x = 30
    y = 30
    is_black = True
    square = canvas.create_rectangle(x, y, x + SQUARE_SIZE, y + SQUARE_SIZE)
    if is_black:
        color = 'black'
    else:
        color = 'white'
    canvas.set_fill_color(square, color)

    canvas.mainloop()
```

As the next step, let's implement a draw_square() function and call it

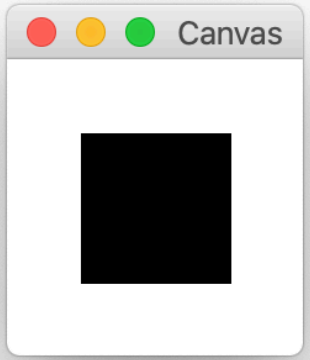


drawSquare_noFunc.py

Draw square via defining and calling a function

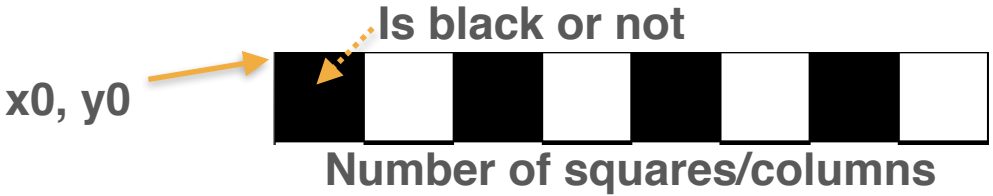
```
from graphics import Canvas

SQUARE_SIZE = 60

def main():
    canvas = Canvas(120, 120)
    draw_square(canvas, 30, 30, True)
    canvas.mainloop()

def draw_square(canvas, x, y, is_black):
    square = canvas.create_rectangle(x , y, x + SQUARE_SIZE, y + SQUARE_SIZE)
    if is_black:
        color = 'black'
    else:
        color = 'white'
    canvas.set_fill_color(square, color)
```

Calling the function



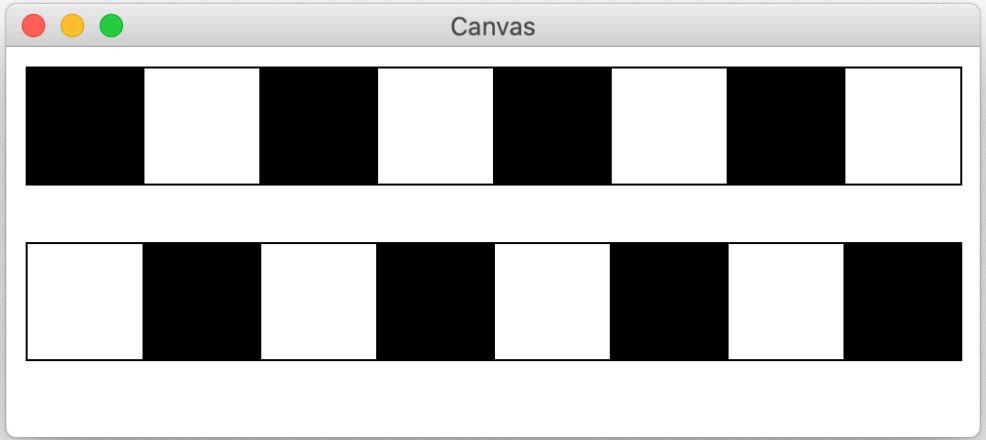
drawSquare.py

Implement draw_row() that calls the draw_square() function



NUM_COLS = 8

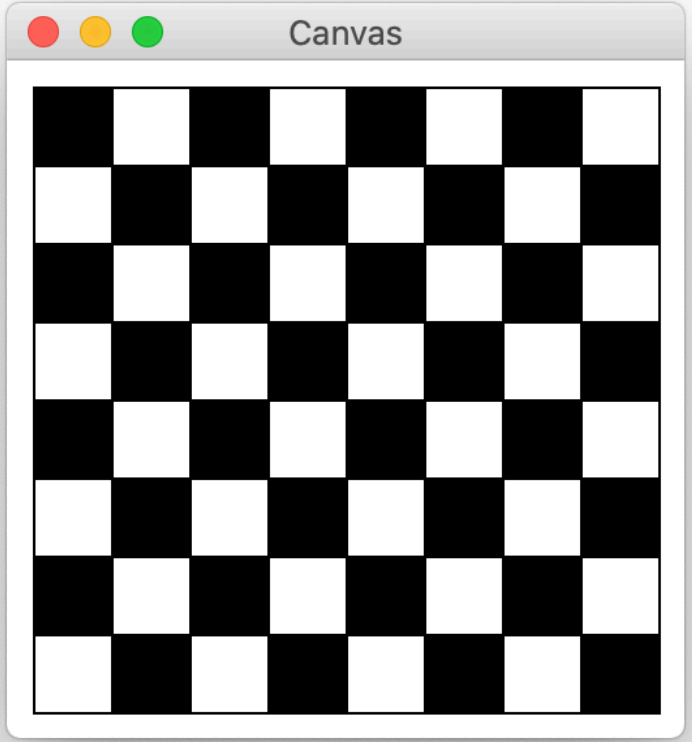
```
def main():
    canvas = Canvas(500, 200)
    is_black = True
    draw_row(canvas, NUM_COLS, 10, 10, is_black)
    is_black = False
    draw_row(canvas, NUM_COLS, 10, 100, is_black)
    canvas.mainloop()

def draw_row(canvas, num_squares, x0, y0, is_black):
    for i in range(num_squares):
        x = x0 + i * SQUARE_SIZE
        draw_square(canvas, x, y0, is_black)
        is_black = not is_black
```



drawRow.py

Draw checker board calling draw_row()

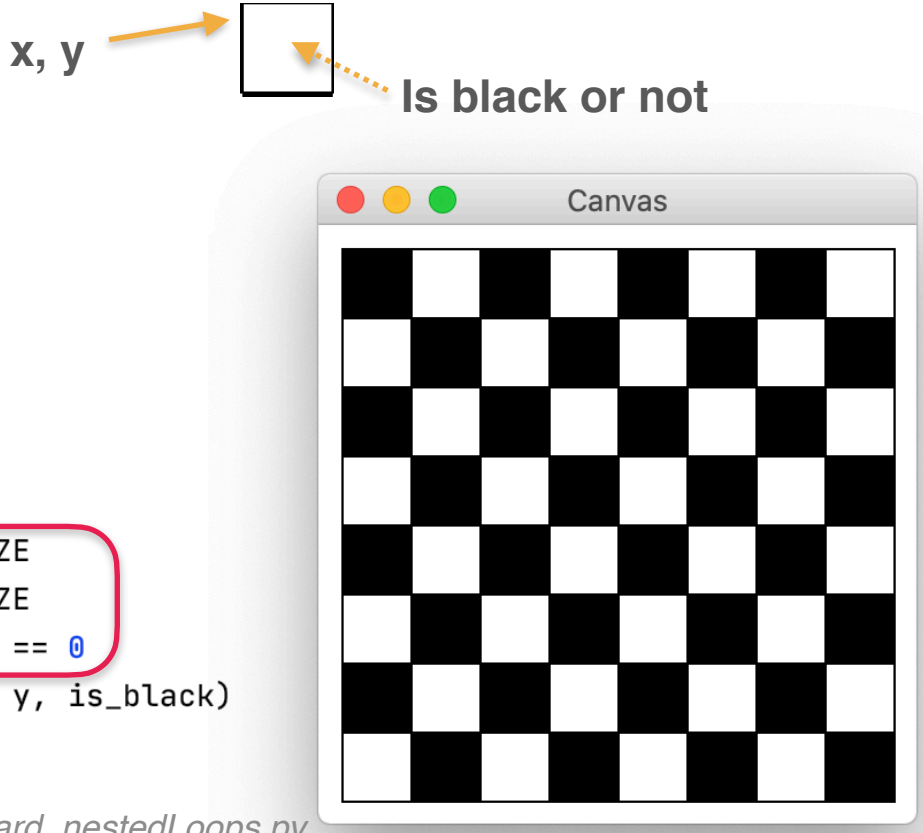
```
SQUARE_SIZE = 30
NUM_ROWS = 8
NUM_COLS = 8

def main():
    canvas = Canvas(260, 260)
    is_black = True
    x0 = 10
    y0 = 10
    for i in range(NUM_ROWS):
        y = y0 + i * SQUARE_SIZE
        draw_row(canvas, NUM_COLS, x0, y, is_black)
        is_black = not is_black
    canvas.mainloop()
```



checkerboard_viaRows.py

Drawing checker board using nested loops

```
SQUARE_SIZE = 30
NUM_ROWS = 8
NUM_COLS = 8

def main():
    canvas = Canvas(260, 260)
    is_black = True
    x0 = 10
    y0 = 10
    for i in range(NUM_ROWS):
        for j in range(NUM_COLS):
            x = x0 + j * SQUARE_SIZE
            y = y0 + i * SQUARE_SIZE
            is_black = (i + j) % 2 == 0
            draw_square(canvas, x, y, is_black)

    canvas.mainloop()
```

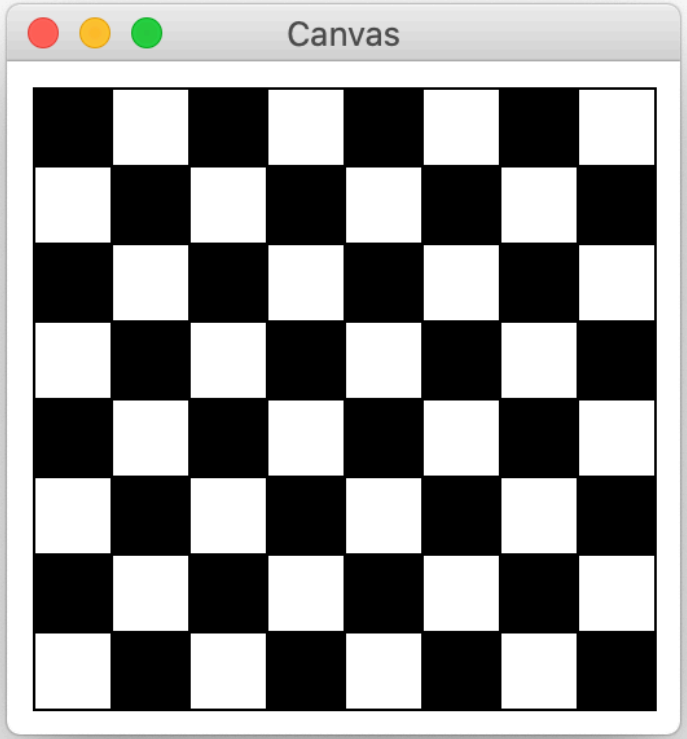


checkerboard_nestedLoops.py

Drawing checker board using nested loops

```
SQUARE_SIZE = 30
def main():
    canvas = Canvas(260, 260)
    is_black = True
    x0 = 10
    y0 = 10
    for i in range(NUM_ROWS):
        for j in range(NUM_COLS):
            x = x0 + j * SQUARE_SIZE
            y = y0 + i * SQUARE_SIZE
            is_black = (i + j) % 2 == 0
            draw_square(canvas, x, y, is_black)

    canvas.mainloop()
```

i = 0

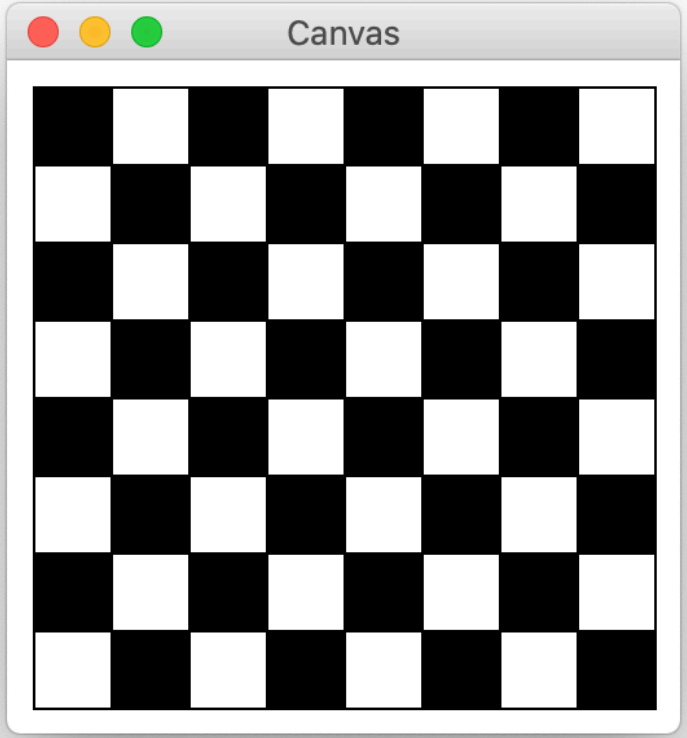


Drawing checker board using nested loops

```
SQUARE_SIZE = 30
def main():
    canvas = Canvas(260, 260)
    is_black = True
    x0 = 10
    y0 = 10
    for i in range(NUM_ROWS):
        for j in range(NUM_COLS):
            x = x0 + j * SQUARE_SIZE
            y = y0 + i * SQUARE_SIZE
            is_black = (i + j) % 2 == 0
            draw_square(canvas, x, y, is_black)

    canvas.mainloop()
```

i = 0

j = 0



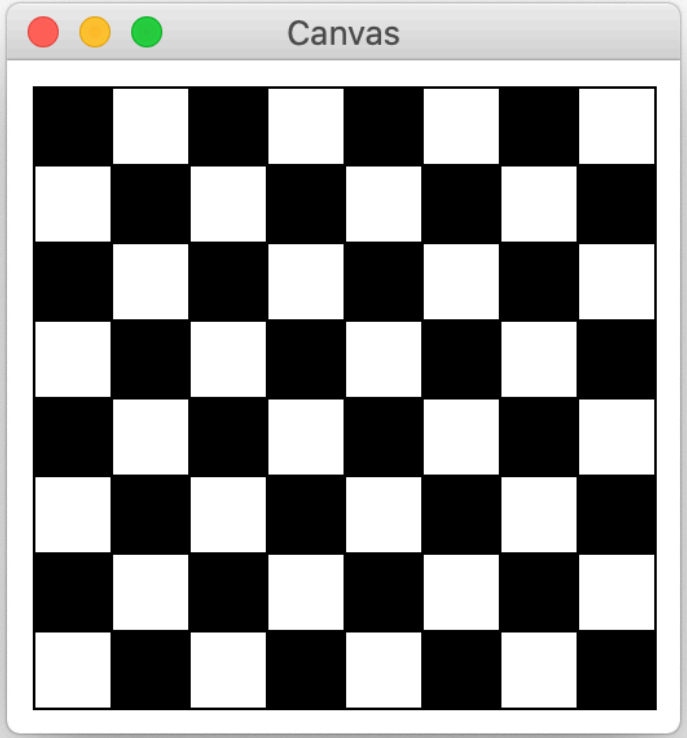
Drawing checker board using nested loops

```
SQUARE_SIZE = 30
def main():
    canvas = Canvas(260, 260)
    is_black = True
    x0 = 10
    y0 = 10
    for i in range(NUM_ROWS):
        for j in range(NUM_COLS):
            x = x0 + j * SQUARE_SIZE
            y = y0 + i * SQUARE_SIZE
            is_black = (i + j) % 2 == 0
            draw_square(canvas, x, y, is_black)

    canvas.mainloop()
```

i = 0

j = 0

x = 10



Drawing checker board using nested loops

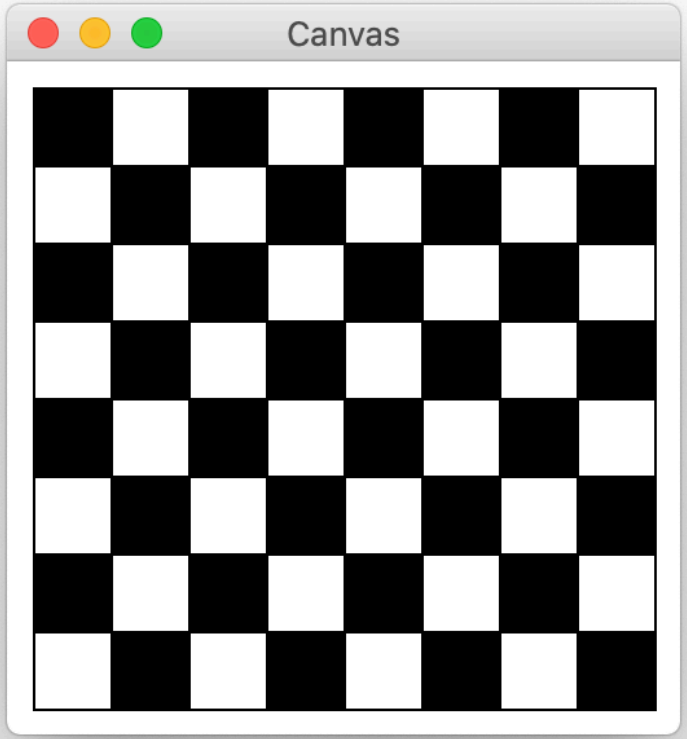
```
SQUARE_SIZE = 30
def main():
    canvas = Canvas(260, 260)
    is_black = True
    x0 = 10
    y0 = 10
    for i in range(NUM_ROWS):
        for j in range(NUM_COLS):
            x = x0 + j * SQUARE_SIZE
            y = y0 + i * SQUARE_SIZE
            is_black = (i + j) % 2 == 0
            draw_square(canvas, x, y, is_black)

    canvas.mainloop()
```

i = 0

j = 0

x = 10

y = 10



Drawing checker board using nested loops

```
SQUARE_SIZE = 30
def main():
    canvas = Canvas(260, 260)
    is_black = True
    x0 = 10
    y0 = 10
    for i in range(NUM_ROWS):
        for j in range(NUM_COLS):
            x = x0 + j * SQUARE_SIZE
            y = y0 + i * SQUARE_SIZE
            is_black = (i + j) % 2 == 0
            draw_square(canvas, x, y, is_black)

    canvas.mainloop()
```

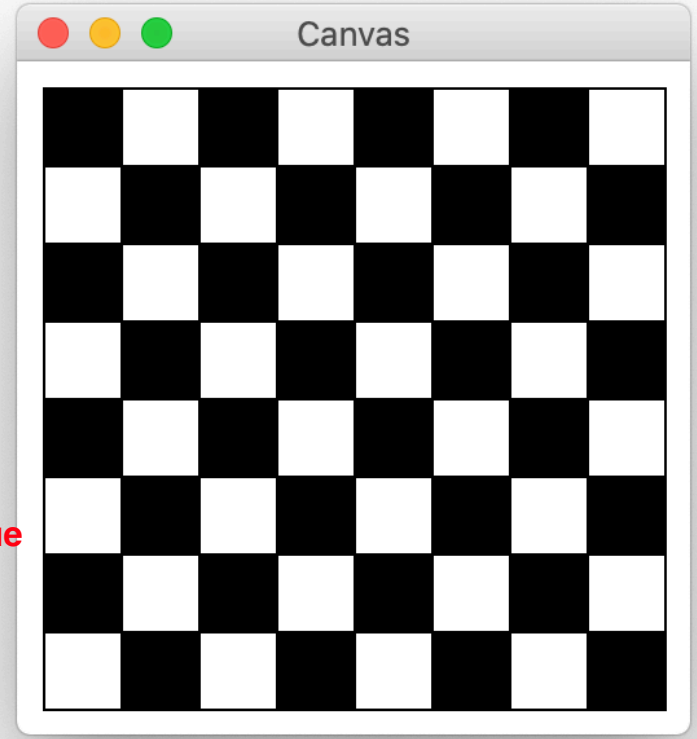
i = 0

j = 0

x = 10

y = 10

is_black = True



Drawing checker board using nested loops

```
SQUARE_SIZE = 30
def main():
    canvas = Canvas(260, 260)
    is_black = True
    x0 = 10
    y0 = 10
    for i in range(NUM_ROWS):
        for j in range(NUM_COLS):
            x = x0 + j * SQUARE_SIZE
            y = y0 + i * SQUARE_SIZE
            is_black = (i + j) % 2 == 0
            draw_square(canvas, x, y, is_black)

    canvas.mainloop()
```

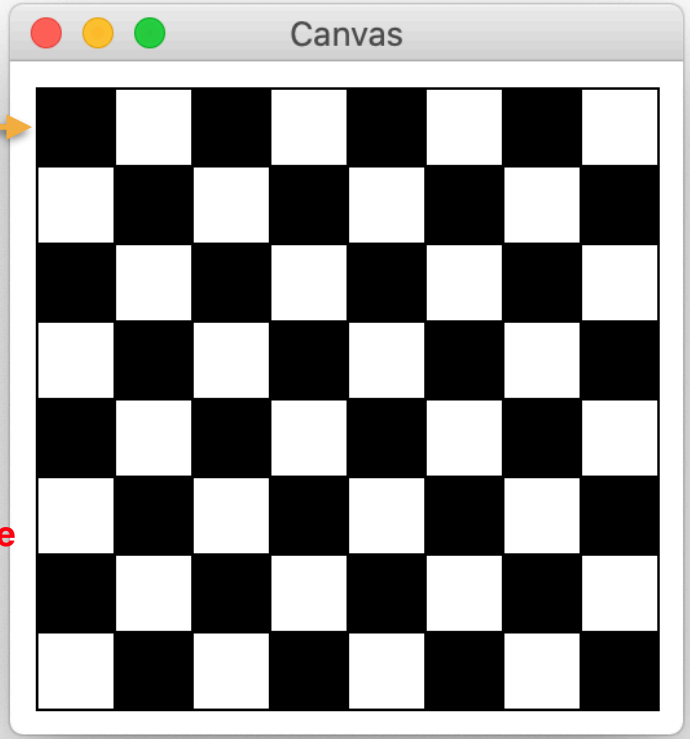
i = 0

j = 0

x = 10

y = 10

is_black = True

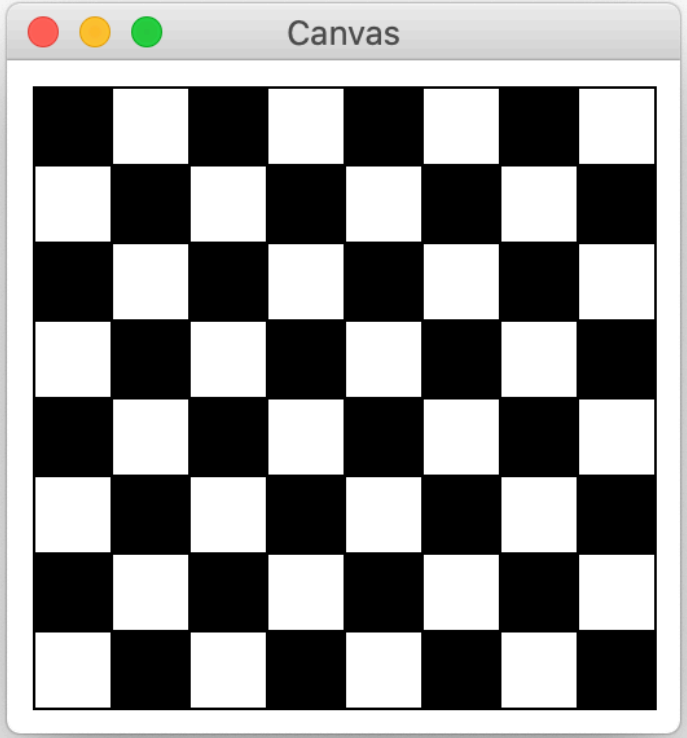


Drawing checker board using nested loops

```
SQUARE_SIZE = 30
def main():
    canvas = Canvas(260, 260)
    is_black = True
    x0 = 10
    y0 = 10
    for i in range(NUM_ROWS):
        for j in range(NUM_COLS):
            x = x0 + j * SQUARE_SIZE
            y = y0 + i * SQUARE_SIZE
            is_black = (i + j) % 2 == 0
            draw_square(canvas, x, y, is_black)

    canvas.mainloop()
```

i = 0

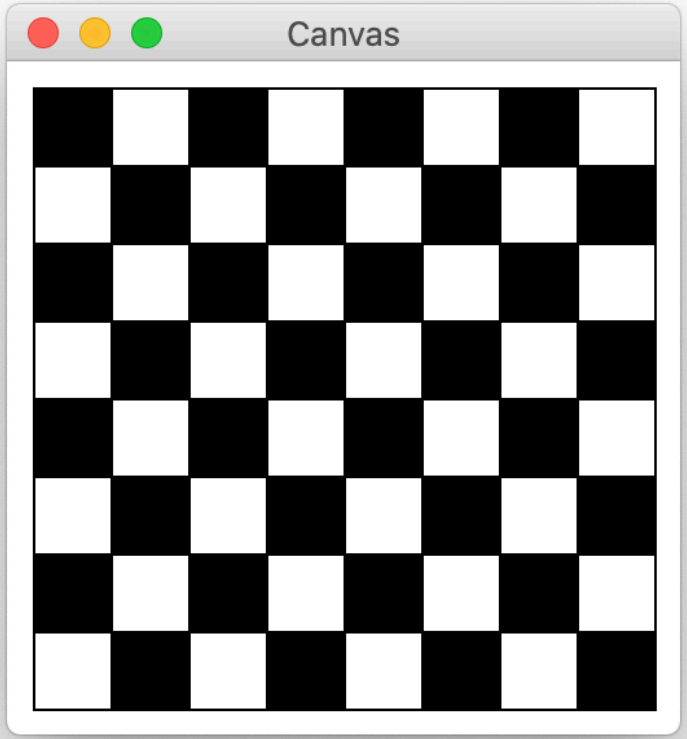
j = 1



Drawing checker board using nested loops

```
SQUARE_SIZE = 30
def main():
    canvas = Canvas(260, 260)
    is_black = True
    x0 = 10
    y0 = 10
    for i in range(NUM_ROWS):
        for j in range(NUM_COLS):
            x = x0 + j * SQUARE_SIZE
            y = y0 + i * SQUARE_SIZE
            is_black = (i + j) % 2 == 0
            draw_square(canvas, x, y, is_black)

    canvas.mainloop()
```

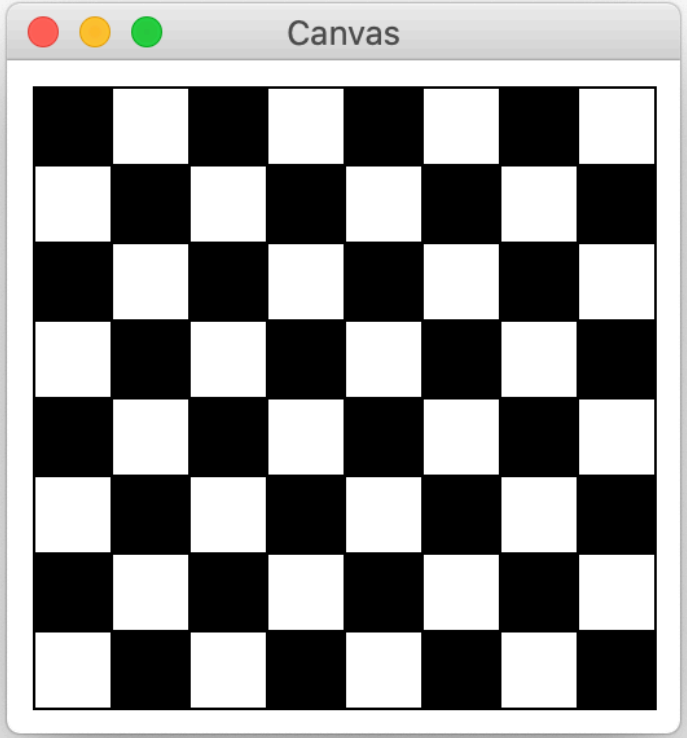
i = 0
j = 1
x = 40



Drawing checker board using nested loops

```
SQUARE_SIZE = 30
def main():
    canvas = Canvas(260, 260)
    is_black = True
    x0 = 10
    y0 = 10
    for i in range(NUM_ROWS):
        for j in range(NUM_COLS):
            x = x0 + j * SQUARE_SIZE
            y = y0 + i * SQUARE_SIZE
            is_black = (i + j) % 2 == 0
            draw_square(canvas, x, y, is_black)

    canvas.mainloop()
```

i = 0
j = 1
x = 40
y = 10



Drawing checker board using nested loops

```
def main():
    canvas = Canvas(260, 260)
    is_black = True
    x0 = 10
    y0 = 10
    for i in range(NUM_ROWS):
        for j in range(NUM_COLS):
            x = x0 + j * SQUARE_SIZE
            y = y0 + i * SQUARE_SIZE
            is_black = (i + j) % 2 == 0
            draw_square(canvas, x, y, is_black)

    canvas.mainloop()
```

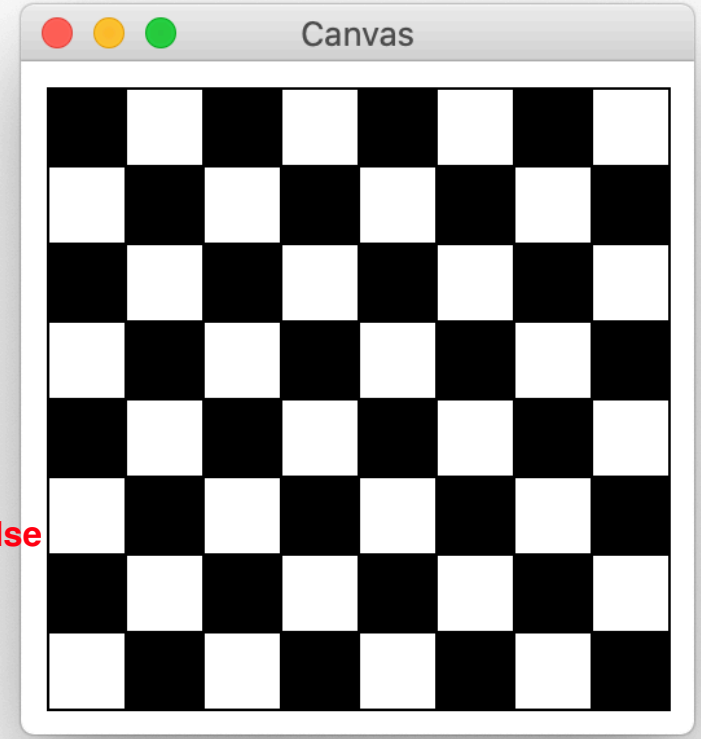
i = 0

j = 1

x = 40

y = 10

is_black = False



Drawing checker board using nested loops

```
def main():
    canvas = Canvas(260, 260)
    is_black = True
    x0 = 10
    y0 = 10
    for i in range(NUM_ROWS):
        for j in range(NUM_COLS):
            x = x0 + j * SQUARE_SIZE
            y = y0 + i * SQUARE_SIZE
            is_black = (i + j) % 2 == 0
            draw_square(canvas, x, y, is_black)

    canvas.mainloop()
```

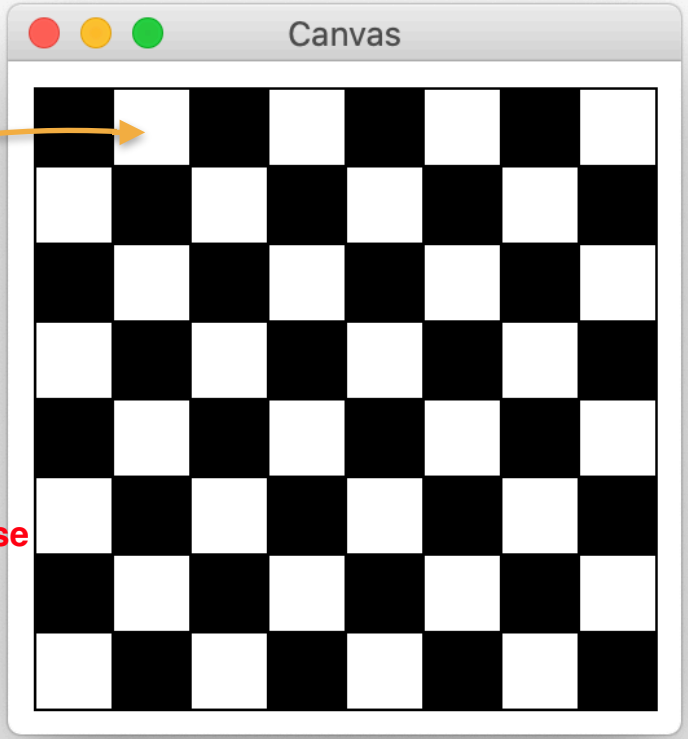
i = 0

j = 1

x = 40

y = 10

is_black = False

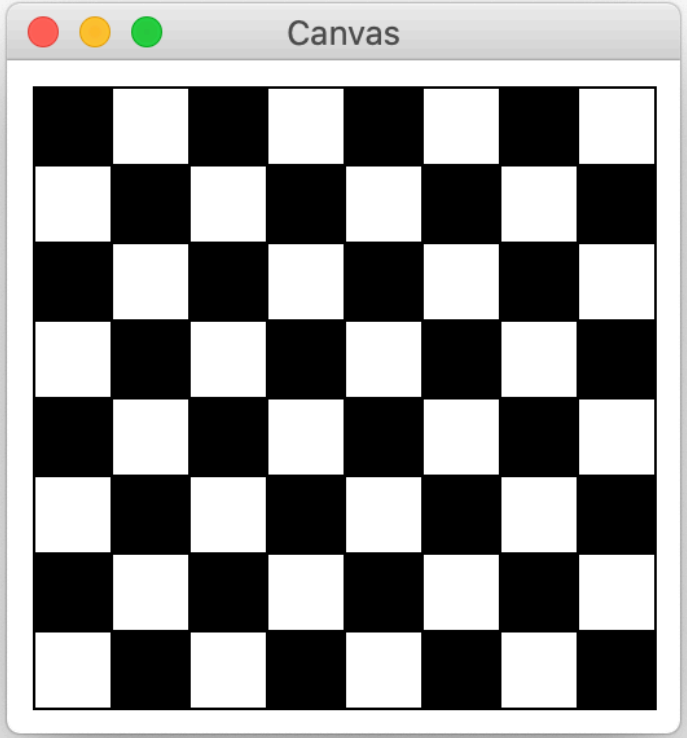


Drawing checker board using nested loops

```
SQUARE_SIZE = 30
def main():
    canvas = Canvas(260, 260)
    is_black = True
    x0 = 10
    y0 = 10
    for i in range(NUM_ROWS):
        for j in range(NUM_COLS):
            x = x0 + j * SQUARE_SIZE
            y = y0 + i * SQUARE_SIZE
            is_black = (i + j) % 2 == 0
            draw_square(canvas, x, y, is_black)

    canvas.mainloop()
```

i = 0

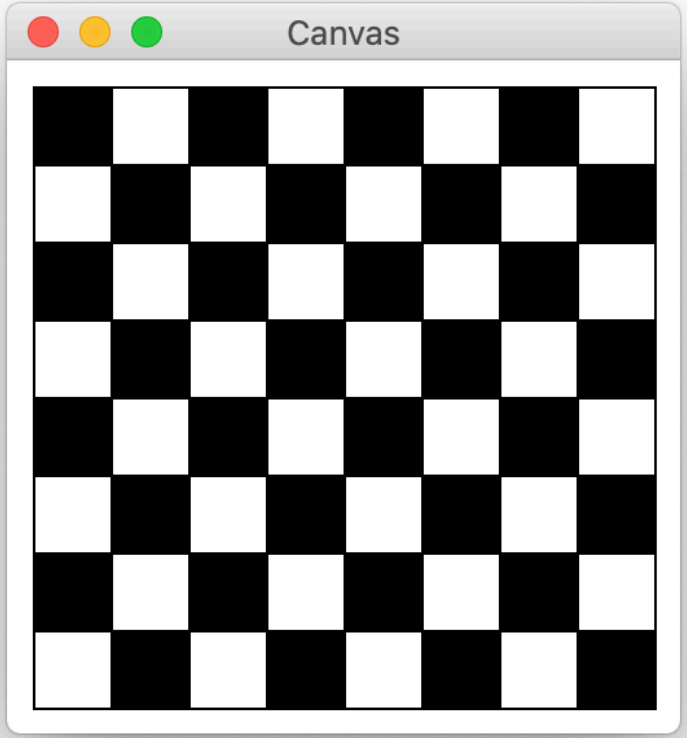
j = 2



Drawing checker board using nested loops

```
SQUARE_SIZE = 30
def main():
    canvas = Canvas(260, 260)
    is_black = True
    x0 = 10
    y0 = 10
    for i in range(NUM_ROWS):
        for j in range(NUM_COLS):
            x = x0 + j * SQUARE_SIZE
            y = y0 + i * SQUARE_SIZE
            is_black = (i + j) % 2 == 0
            draw_square(canvas, x, y, is_black)

    canvas.mainloop()
```

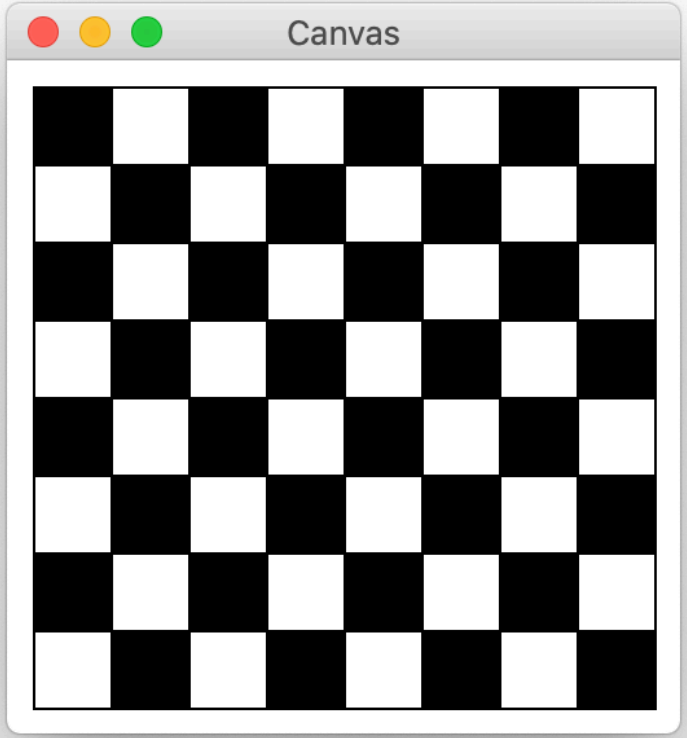
i = 0
j = 2
x = 70



Drawing checker board using nested loops

```
SQUARE_SIZE = 30
def main():
    canvas = Canvas(260, 260)
    is_black = True
    x0 = 10
    y0 = 10
    for i in range(NUM_ROWS):
        for j in range(NUM_COLS):
            x = x0 + j * SQUARE_SIZE
            y = y0 + i * SQUARE_SIZE
            is_black = (i + j) % 2 == 0
            draw_square(canvas, x, y, is_black)

    canvas.mainloop()
```

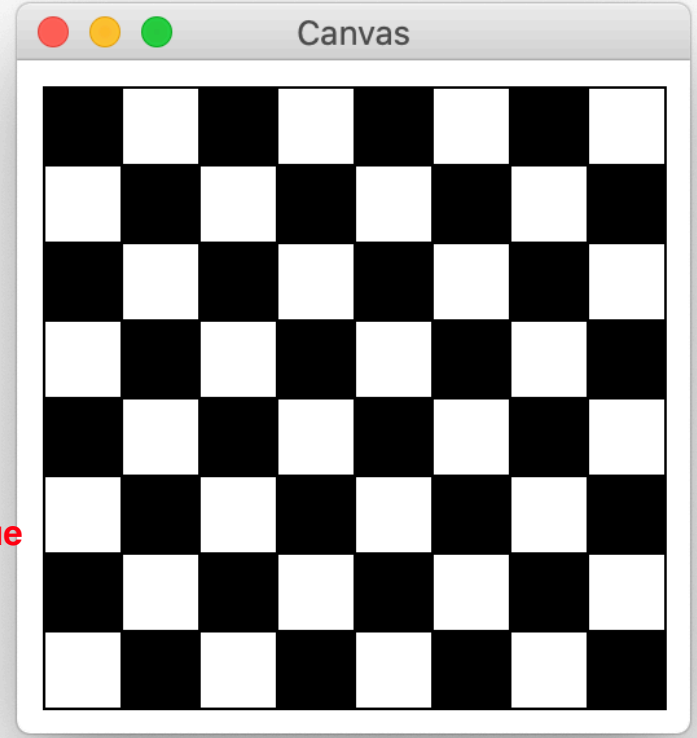
i = 0
j = 2
x = 70
y = 10



Drawing checker board using nested loops

```
def main():
    canvas = Canvas(260, 260)
    is_black = True
    x0 = 10
    y0 = 10
    for i in range(NUM_ROWS):
        for j in range(NUM_COLS):
            x = x0 + j * SQUARE_SIZE
            y = y0 + i * SQUARE_SIZE
            is_black = (i + j) % 2 == 0
            draw_square(canvas, x, y, is_black)

    canvas.mainloop()
```

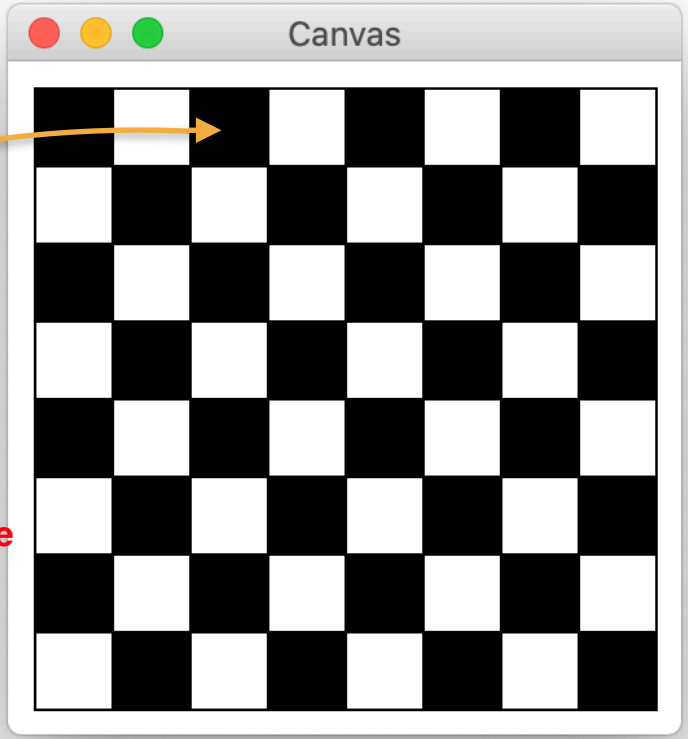
i = 0
j = 2
x = 70
y = 10
is_black = True



Drawing checker board using nested loops

```
def main():
    canvas = Canvas(260, 260)
    is_black = True
    x0 = 10
    y0 = 10
    for i in range(NUM_ROWS):
        for j in range(NUM_COLS):
            x = x0 + j * SQUARE_SIZE
            y = y0 + i * SQUARE_SIZE
            is_black = (i + j) % 2 == 0
            draw_square(canvas, x, y, is_black)

    canvas.mainloop()
```

i = 0
j = 2
x = 70
y = 10
is_black = True



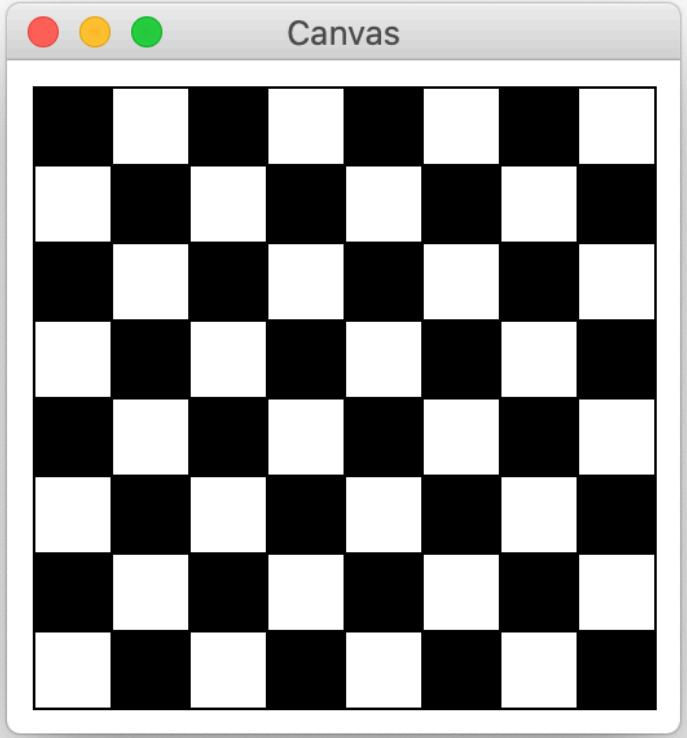
Many loops pass….

Drawing checker board using nested loops

```
SQUARE_SIZE = 30
def main():
    canvas = Canvas(260, 260)
    is_black = True
    x0 = 10
    y0 = 10
    for i in range(NUM_ROWS):
        for j in range(NUM_COLS):
            x = x0 + j * SQUARE_SIZE
            y = y0 + i * SQUARE_SIZE
            is_black = (i + j) % 2 == 0
            draw_square(canvas, x, y, is_black)

    canvas.mainloop()
```

i = 5

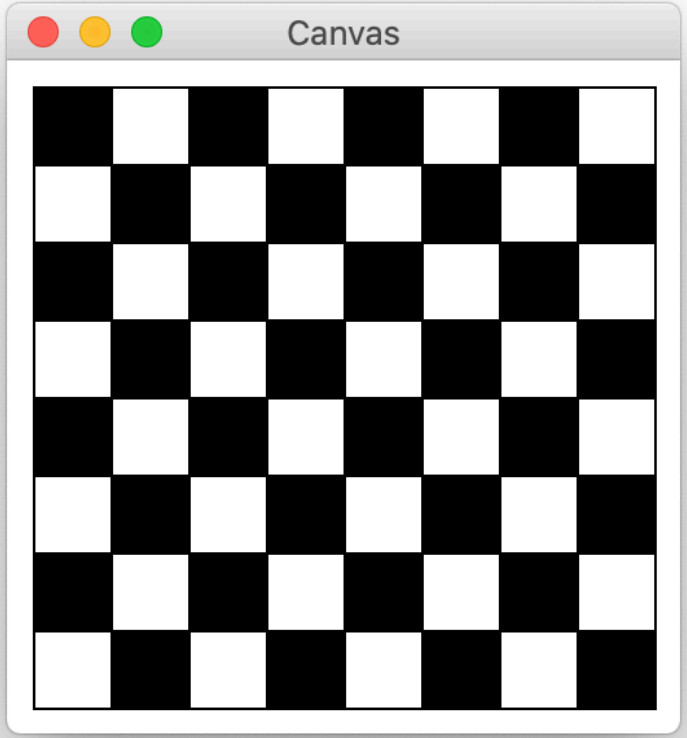
j = 2



Drawing checker board using nested loops

```
SQUARE_SIZE = 30
def main():
    canvas = Canvas(260, 260)
    is_black = True
    x0 = 10
    y0 = 10
    for i in range(NUM_ROWS):
        for j in range(NUM_COLS):
            x = x0 + j * SQUARE_SIZE
            y = y0 + i * SQUARE_SIZE
            is_black = (i + j) % 2 == 0
            draw_square(canvas, x, y, is_black)

    canvas.mainloop()
```

i = 5
j = 2
x = 70
y = 160



Drawing checker board using nested loops

```
def main():
    canvas = Canvas(260, 260)
    is_black = True
    x0 = 10
    y0 = 10
    for i in range(NUM_ROWS):
        for j in range(NUM_COLS):
            x = x0 + j * SQUARE_SIZE
            y = y0 + i * SQUARE_SIZE
            is_black = (i + j) % 2 == 0
            draw_square(canvas, x, y, is_black)

    canvas.mainloop()
```

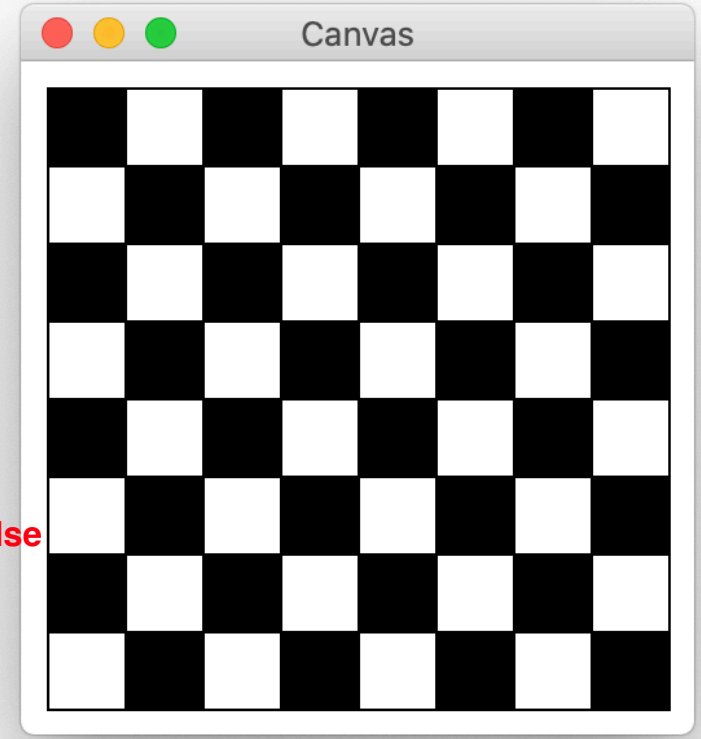
i = 5

j = 2

x = 70

y = 160

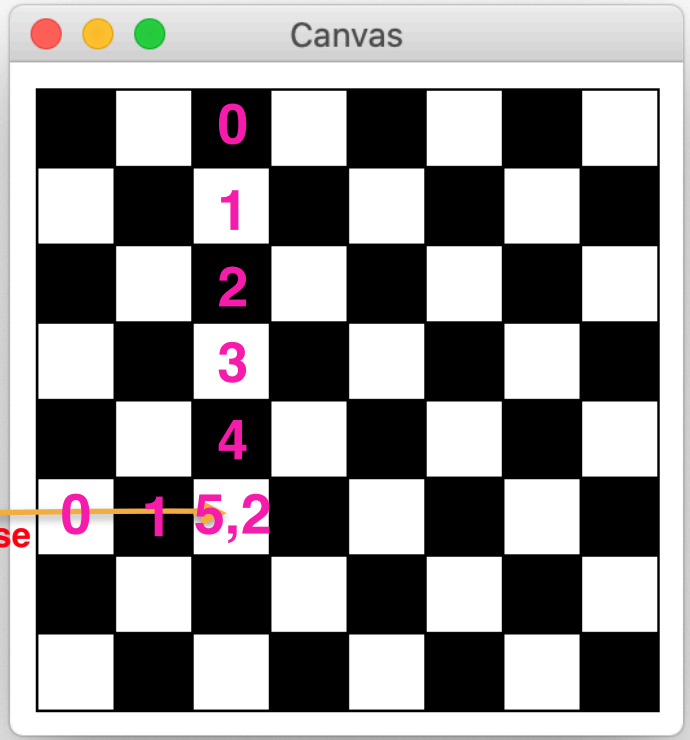
is_black = False



Drawing checker board using nested loops

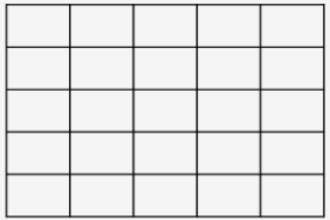
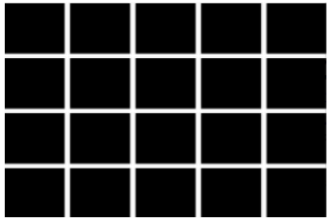
```
def main():
    canvas = Canvas(260, 260)
    is_black = True
    x0 = 10
    y0 = 10
    for i in range(NUM_ROWS):
        for j in range(NUM_COLS):
            x = x0 + j * SQUARE_SIZE
            y = y0 + i * SQUARE_SIZE
            is_black = (i + j) % 2 == 0
            draw_square(canvas, x, y, is_black)

    canvas.mainloop()
```

i = 5
j = 2
x = 70
y = 160
is_black = False



Rest of this morning

Day 6: Loops and Animation

Morning Project [here]		
	Print Indices Quickstart SL Notes	For Loops
	Optical Illusion Project SL Notes	For Loops