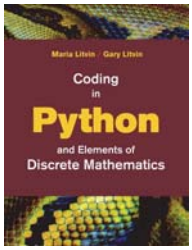


Coding in **Python** and Elements of Discrete Mathematics

Maria Litvin
Phillips Academy, Andover, Massachusetts

Gary Litvin
Skylight Software, Inc.

Skylight Publishing
Andover, Massachusetts

Skylight Publishing
9 Bartlet Street, Suite 70
Andover, MA 01810

web: http://www.skylit.com
e-mail: sales@skylit.com
support@skylit.com

Library of Congress Control Number: 2019905086

ISBN 978-0-9972528-4-2

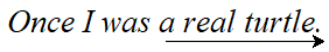
1 2 3 4 5 6 7 23 22 21 20 19

Printed in the United States of America

The available font names are those installed in your operating system, but in a portable program it is advisable to use only common fonts that are available in most systems, such as `'Arial'`, `'Times'`, and `'Courier'`, or just write `None` for the font name to use the default font.

By default, the left end of the baseline of text will be at the current turtle position. An optional parameter, `align='center'` or `align='right'`, will place the center or the right end of the baseline at the current position. The optional parameter, `move=True` will move the turtle to the end of the baseline (and draw if the pen is down). For example:

```
t.write('Once I was a real turtle.',
       font=('times', 20, 'italic'), align='center', move=True)
```



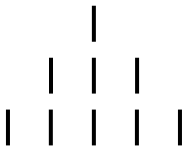
If the text string contains `'\n'` characters, `write` will correctly display multiple lines.

Section 9.3 ~ Exercises

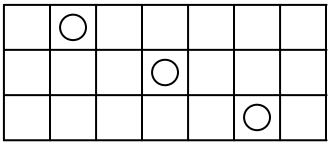
1. Draw a "hamburger" button

✓

2. The game of *Nim* is, theoretically, played with piles of stones, but it is commonly played with rows of sticks instead. Draw a configuration with three rows of sticks:

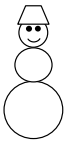


3. Another, *isomorphic* (mathematically identical) representation of Nim is tokens moving from left to right on a rectangular board. Draw the Nim configuration with three tokens:



(It is identical to the three rows of sticks in the previous question.) ✓

4.■ Draw a snowman:



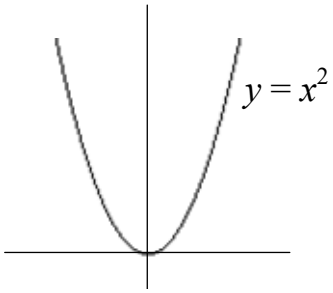
5. In the Tower of Hanoi puzzle, you need to transfer a pyramid of disks from one peg to another, using the third peg as a "spare." You can only move one disk at a time, and you may place it only on top of a larger disk or on the base. Draw a two-dimensional sketch of the puzzle with five disks:

6.■ Draw a stop sign:

Don't worry about an exact font match — Arial will do for this exercise. ✓

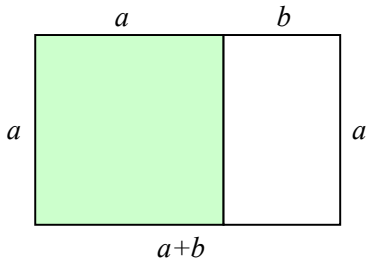
7.■ Display the code that draws a hexagon to the right of the hexagon it draws (see Example 2 in the previous section). Use the Courier font for the code.

8.■ Draw a fairly smooth graph of the parabola $y = x^2$ with a label to its right:



ⵑ Hint: generate the segment of the parabola for $-3 \le x \le 3$ but scale the graph by a factor of 30 or 40. ⵑ ✓

9.◆ Draw a diagram that illustrates the *golden ratio* (actually taken from the next chapter of this book). Add the equation to the right of the rectangle:



$$\frac{a+b}{a} = \frac{a}{b}$$

9.4 Colors

In turtle graphics a virtual turtle draws on virtual paper with a virtual pen. No pen exists, of course. What you see on your computer screen is ultimately determined by the contents of the video memory (VRAM) on the *graphics adapter* card or the graphics processor chip. VRAM represents a rectangular array of *pixels* (picture elements). Each pixel has a particular color, which can be represented as a mix of red, green, and blue components, each with its own intensity. A typical graphics adapter uses eight bits to represent each of the red, green, and blue (RGB) values (in the range from 0 to 255). The image on the screen is produced by setting the color of each pixel in VRAM. The video hardware scans the whole video memory continuously and refreshes the image on the screen.

A graphics processor is what we call a *raster* device: each individual pixel can be set separately from other pixels. (This is different from a *vector* device, such as a plotter, which actually draws lines on paper directly from point *A* to point *B*, with a pen of a particular color.) To draw a red line or a circle on a raster device, you need to set just the right group of pixels to the red color. That's where a graphics package helps: you certainly don't want to program all those functions for setting pixels yourself.

A graphics package has to provide functions for setting colors. Python's `turtle` inherits screen and color handling from the `tkinter` package (`Tk` interface), which is Python's standard toolkit for GUI (Graphical User Interface) development. `tkinter` uses names assigned to several hundred selected colors. (These names are standard in web app development environments.) You can find some of the named colors with their RGB components in hex and/or decimal form on many web sites, for example, https://trinket.io/docs/colors. A complete list of named colors is available at https://www.tcl.tk/man/tcl8.4/TkCmd/colors.htm.

Table 9-4 summarizes `turtle`'s color functions.

Function	Action

`color(c)`
`color(c1, c2)`
`color()`
	Set pen color and fill color to `c`.
	Set pen color to `c1` and fill color to `c2`.
	Return turtle's current pen color and fill color (each as an RGB tuple or name).

`pencolor(c)` `pencolor()`	Set pen color. Return turtle's current pen color.
`fillcolor(c)` `fillcolor()`	Set fill color. Return turtle's current fill color.
`pensize(w)`	Set the width of strokes to `w`; the default is 1.
`Screen().colormode(256)`	Set RGB tuples scale to 0-255.

Table 9-4. `turtle` color handling functions

The `color` function has two forms: `color(c)` sets the color `c` as both the pen color and the fill color. `color(c1, c2)` sets `c1` as the pen color and `c2` as the fill color.