# Rational

I’ve been a computer programmer for over a half century. I’ve found it fun, instructive and profitable. It saddens me that many people have missed that opportunity. I’m not concerned with those who are already interested and motivated to do programming. There are many available avenues, online courses, books, and schools. I do not want to compete withs these. I’m looking to aid those who may not yet know what computer programming is or whether they might be interested in the skill.

# Class Plan

1. Provide the student enough computer programming tools, including Python, to do programming.
2. Provide a few programing examples.
3. Have the student run these examples on their computer.
4. Have the student make a small program modification.
5. Have the student run the modified program.
6. Discus the program and expand our knowledge of programming.

# Rational for the Blind

As experience with teaching the subject increased, it became obvious that care was needed to keep the interest of the early students. Unless and until the student visualizes the power of programming it is very hard to create enough energy to overcome the necessary details of the discipline. I have addressed this by using as many visual / graphical programming examples as possible. Python has a module turtle which provides a wealth of graphical possibilities. Great for the sighted, but not so for others. How can we avoid leaving the blind student behind? We have developed software which converts simple graphics produced by our samples in to low resolution text pictures suitable output to braille. I’ll include a short example later in this document.

# Current Course Offerings (Taken form Announcements)

# One Session Overview to Programming Using Python

This is a single, one-off session that will provide an overview of computer programming with Python. Students need no prior computer programming experience though basic computer experience is assumed. The popular Python programming language is used as an example.  Included in this session are a number of simple but realistic examples in Python which the student views, runs, modifies and runs again. This session emphasizes student participation - **learning by doing**. During this session the student is shown IDLE, a simple but powerful Python programming development tool.

Programming areas demonstrated in this session include the following:

Programming concepts:

* program - telling the computer what to do
* program file – store a program
* programming language – rules for the telling

Python parts:

* program structure / layout: comments, indentation, import
* calculation / comparisons: +, -, \*,  \*\*, /, //, %, (), +=, \*=
* variables – named places to store stuff
* control flow:
  + if, else
  + loops: while, for

# Introduction to Programming using Python (7 Week Online Series)

Join us for this **online**introductory course series for those new to programming with instructor Charles (Ray) Smith. Learn what computer programming is all about from the programmer's point of view and learn how to write and run programs **you** build using the programming language Python. Minimal computer experience is useful, but no previous computer programming experience is necessary - this series is designed to help you decide if programming is for you!

For this interactive, seven-week course series geared towards adults, a personal laptop will be required, as well as the ability to install Python on said laptop. As an alumnus of Massachusetts Institute of Technology, Ray has a background in programming that has spanned the professional world from MIT and GenRad to IBM and eDial/Alcatel-Lucent and is interested in expanding individual understanding of how coding informs the world in which we live.

# TurtleBraille – graphical braille output from Python turtle

Our software translates an approximately 1024x1024 color graphic into a 40x25 text map, with the colors represented by the first letter of the color, e.g., the color red is represented by the character “r”. The following views attempt to provide how our software facilitate the blind programmer in providing an approximate “picture” of what the sighted programmer sees when they run the same program.

Simple display program to create a simple graphic.

# spokes.py

# Display a star with spokes

from turtle\_braille\_link import \* # Set link to library

#from turtle import \* # Bring in turtle graphic functions

for i in range(7): # Do things 7 times

if i == 0:

color("red")

elif i == 1:

color("orange")

elif i == 2:

color("yellow")

elif i == 3:

color("green")

elif i == 4:

color("blue")

elif i == 5:

color("indigo")

else:

color("violet")

forward(300)

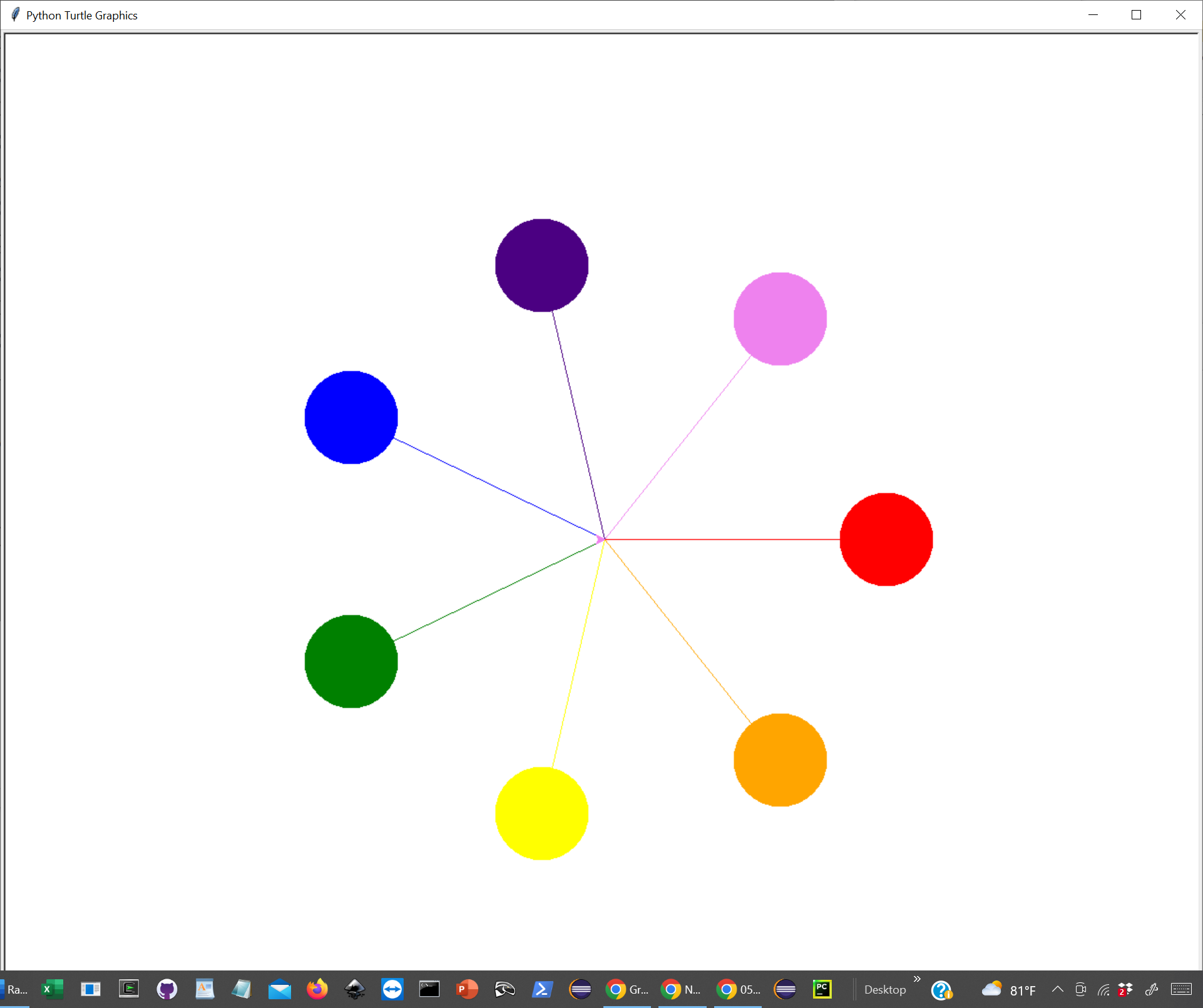
dot(100)

backward(300)

right(360/7)

done()

Python turtle window output



Program text output to send to braille embosser

Note the harsh compromise first letter of color displayed

Trace levels from properties file C:\Users\raysm\Desktop\one\_session\_braille\my\_work\01\_intro\spokes.properties

loadTraceFlags:

turtle braille support

Braille Display - Braille Print Output

**,,,,,,,,,,,,,iii**

**,,,,,,,,,,,,iiiiii**

**,,,,,,,,,,,,iiiiii,,,,,,,,vvv**

**,,,,,,,,,,,,iiiii,,,,,,,vvvvvv**

**,,,,,,,,,,,,,,,i,,,,,,,,vvvvvv**

**,,,bbb,,,,,,,,,ii,,,,,,,,vvvvv**

**,,bbbbb,,,,,,,,,i,,,,,,vvv**

**,,bbbbbb,,,,,,,,i,,,,,vv**

**,,bbbbbbbb,,,,,,ii,,,vv**

**,,,,,,,,,,bbbb,,,i,,v,,,,,,,,,,,rr**

**,,,,,,,,,,,,,bbbbivv,,,,,,,,,,rrrrrr**

**,,,,,,,,,,,,,,,,bvvrrrrrrrrrrrrrrrrr**

**,,,,,,,,,,,,,ggg,yoo,,,,,,,,,,rrrrrr**

**,,,,g,,,,ggggg,,,y,,o,,,,,,,,,,,rr**

**,,gggggggg,,,,,,yy,,,oo**

**,,gggggg,,,,,,,,y,,,,,oo**

**,,ggggg,,,,,,,,,y,,,,,,ooo**

**,,,,g,,,,,,,,,,yy,,,,,,,,ooooo**

**,,,,,,,,,,,,,,,y,,,,,,,,oooooo**

**,,,,,,,,,,,,yyyyyy,,,,,,oooooo**

**,,,,,,,,,,,,yyyyyy,,,,,,,,ooo**

**,,,,,,,,,,,,yyyyyy**

**,,,,,,,,,,,,,yyy**

Adjustments made from comments provided by Susan Sullivan and her 8’th graders Jake and Logan from Perkins School for the Blind:

* 1. Commas to replace leading blanks so the perspective was not lost due to the Braille software compressing blanks. The comma was chosen because of its sparce (one dot) would best approximate a blank region
  2. Leading blank areas toward left and toward top are removed to facilitate the locating of meaningful graphics.

Development/Diagnostic window to help me see the braille view

