Challenges for Python LED Exercises

1. Blink.py
   1. Original intent:
      1. Understand the “strip[0]” position
      2. Understand RGB numbering
      3. Understand “sleep(1)” (1 = seconds)
      4. Introduce “while True:” looping concept
   2. Challenge:
      1. Select a different position
      2. Select a different color
      3. Select a different delay
2. Red-green-blue.py
   1. Original intent:
      1. Reinforce above concepts
   2. Challenge:
      1. Change brightness
      2. Make lights blink (turn off) before changing colors
      3. Add new colors (blending RGB)
3. Dimmer.py
   1. Original intent:
      1. Reinforce above
      2. Introduce ‘stepping’ parameter in for loop, and thus the concept of counting backwards
   2. Challenge:
      1. Select a different rate at which to increase and decrease brightness (different step value) – light up only the even-numbered LEDs (or odd)
      2. Have different lights dim/brighten – make each of the lights in the strip brighten/dim in sequence
4. Move.py
   1. Original intent:
      1. Reinforce above
      2. Introduce config.NUMBER\_PIXELS & using that to control the end of loop
         1. Note that NUMBER\_PIXELS indicates the quantity of LEDs in the strip, but that the LEDs are indexed/identified starting with 0
   2. Challenge:
      1. *Make the move backwards (well .. this is done in the next exercise)*
      2. Illuminate every third LED on the strip
      3. Make the lights appear to “run faster” backwards than forwards
5. Up-and-down.py
   1. Original intent:
      1. Reinforce above
   2. Challenge:
      1. See challenges in Move.py
6. Color-wipe.py
   1. Original intent:
      1. Reinforce above
   2. Challenge:
      1. Make it so that the entire strip seems to change from R to G to B at once
      2. Make a stop light – top 1/3 = red, mid = yellow, lower 1/3 = green
7. Random.py
   1. Original intent:
      1. Reminder about “randint” function
      2. Reinforce RGB, looping
   2. Challenge:
      1. Have two LEDs lit at the same time, one randomly in the lower half of the strip and one in the upper half.
      2. Have two LEDs lit at the same time, each the same random distance from the center point
8. Color-wheel.py
   1. Original intent:
      1. Reminder about creating functions
      2. Use of “return”
   2. Challenge:
      1. Either make entire strip become a color wheel
      2. Or … make each LED in the strip become a color wheel, one by one with a slight pause between
9. Rainbow.py
   1. Original intent:
      1. Reinforce functions/return
   2. Challenge:
      1. Make it build rainbow backwards (so end result is VIBGYOR)
10. Rainbow-cycle.py
    1. Original intent:
       1. More of rainbow.py, but moving in continuous loop
    2. Challenge:
       1. Make it cycle forwards, then backwards