Tessellations - with starter code
Lesson Prep and Teacher Notes

Materials needed:

- Starter code for participants
- Graph paper printed for participants
- Slide deck of instructions
- CodeX!

Opening:

- OPTIONAL: Start with a website on tessellations and have a short discussion on what makes a tessellation.
    - https://www.mathsisfun.com/geometry/tessellation.html
    - Or just do a google image search with “tessellations” and see a lot of cool examples.
- Start with the slide deck. Slide 2 gives examples.
- Slide 3 - goals of the mission

Learn about drawing rectangles on the CodeX (filled and outlined)

- Slide 4 goes over the commands to draw a rectangle
- SUGGESTION: Take your time and review the values involved: x, y, width and height
- OPTIONAL: Hand out mini-graph papers to each person. Have them practice drawing rectangles and identifying the values needed to draw them. They can even draw their own “brick” to use in the first tessellation if they are more advanced (not beginners).

Shape #1: Brick (single rectangle)

- Slide 5 has students start their file and copy and paste starter code into their file.
- NOTE: Have a way to give the starter code to the participants (Google doc, copy and paste from slides, etc.)
- The code will draw one row of bricks across the top of the display.
- Slide 6 goes over the values of the brick
- Slide 7 goes over the code for creating one row of bricks
- Slide 8 goes over calling the function (its arguments)
- SUGGESTION: Take time to go over the slides and make sure they are comfortable with the loop and how it works and the function call.
- NOTE: The starter code includes a list of colors, and the loop will iterate through the list. This code is given and will be repeated in each tessellation. Participants are welcome to modify the list of colors, but they shouldn’t change the code.

Tessellation for Shape #1

- Slide 9 goes over the code to add a second loop. The inside loop doesn’t change, just needs to be indented. Slow down and go over as necessary.
- Slide 10 shows the intended output

Offsetting the tessellation for Shape #1

- Slide 11 shows a more interesting variation of the tessellation.
- This part uses a Boolean variable to “toggle” between a regular row and an offset row. If participants aren’t familiar with Boolean, it can be explained, or you can change it to a number with two values, but that could be more complicated because you can’t just say “not toggle”.
- Slide 12 gives instructions
- Slide 13 shows the intended output

Shape 2 – one row.

- This shape is derived from Tetris.
- Slides 14 and 15 introduce the shape. We are calling it “skew”. The code is given to them, but they can change the size of the two bricks if they want to.
- The program includes a menu that maps a button press with a tessellation. Slide 16 has participants add to the menu for the skew.
- Slide 17 shows what one row of skewed shapes should look like. The code isn’t given to them, but it is pretty much identical to the bricks single row (or the inside loop of the final tessellation).
- A good place to start is to copy and paste the inside loop and paste it into tessell_skew() and just change “brick” to “skew”. They can try the code and adjust the number of loops as needed. You can see from the picture on the slide that looping 5 times gives the desired output.
    -
```
def tessell_skew(x, y, index):
    for column in range(5):
        skew(x, y, my_colors[index])
        x = x + 50
        index = index + 1
        if index == len(my_colors):
            index = 0
```

Tessellation Shape 2

- Slide 18
- Participants add another loop to create multiple rows of skews. The code is not given to them, but it is pretty much the same as as tessell_bricks(). They can use it as a guide or copy and paste and change the function call to “skew”.
- NOTE: This may take some trial and error because the height of this shape is double (two rectangles). Hopefully productive struggle.

Fix the gap – shape 2

- The tessellation leaves a gap on the left side. An easy fix for this is with the toggle and starting argument. Just follow the two instructions on Slide 19.
- Slide 20 shows the desired output.

Shape 3 – half cross

NOTE This shape can be skipped if you don’t have time. Nothing new here (except the gap).**

- This is another Tetris shape that is pretty easy to tessellate.
- Slides 21 and 22 introduce the shape and go over the math.
- Slide 23 – add to the menu
- Slide 24 – write code to draw one row of half crosses (similar to the other two shapes).

- **There is a gap! Slide 25 shows an easy fix. This time change the starting y argument instead of x. If you skip this shape, be aware of this alternate way to fix a gap.**
- Complete the tessellation by adding a second loop and toggle variable. This is the third time doing this code, so hopefully they don't have a lot of struggles. (Slide 26)

Shape 4 – isosceles triangle

- This shape requires three lines, not rectangles
- Slide 27 goes over drawing lines and the values needed.
- SUGGESTION: Go over this as needed. If participants have graph paper, have them draw lines and practice identifying the values needed to draw them.
- ANOTHER SUGGESTION: Have them draw a triangle (any shape, not necessarily isosceles) and identify the values for all three lines.
- Slide 28 has participants draw an isosceles triangle in the upper left corner and identify the values. You can just use the ones on the slide if you don't want them to do their own. Depends on the experience of the participants.
- Slide 29:
  There is a lot of instructions on this slide!
  - Instructions say to create a function to draw the triangle. The code is included in the starter code, so you can just review the code already there, or they can draw their own.
  - Instructions also indicate to add to the menu. Don't skip this step!
  - Add code to tessell_triangle() to draw a row.
- Slide 30 – add another row to tessell the triangles.
  - This code is like all the others
  - It may be difficult to see because the lines are thin, but there are many gaps in this tessellation – all the flipped triangles!
- Slide 31 – define a function for the flipped triangle
  - The function definition is given in the starter code – they will need to delete "pass"
  - The code is almost identical to the first triangle, with just the y values needing to be flipped. This is explained on the slide.
  - The code is not given, but they should be able to figure it out.

```
def flipped(x, y, color):
    display.draw_line(x, y, x+60, y, color)
    display.draw_line(x, y, x+30, y-30, color)
    display.draw_line(x+30, y-30, x+60, y, color)
```

- Slide 32 – use the flipped function
  - This is another use for "toggle", which makes the complete tessellation fairly straightforward.
  - On the bottom half of the slide, you can see the code where I made 1 pixel adjustments to drawing the triangles. This helps see the individual triangles because otherwise the sides overlap and it looks like of the same as without the flipped triangle.
  - Also notice that y doesn't need to be reset the for the second row, because the vertex is the same as the first row.
  - Adjust the number of rows as needed
- Slide 33 – shows intended output

Your own shape

If time permits, participants can try drawing and tessellating their own shape. They can keep it simple and do one of the shapes, but a different size, or try a different shape altogether. Some suggestions are given on Slide 35.

- Slide 34 – introduce doing their own shape
- Slide 35 – suggestions for a shape
- Slide 36 – use graph paper to design the shape and calculate the values needed
- Slide 37 – instructions for the tessellations.
    - These are the same steps they have been doing for the first four tessellations, so they are not spelled out in great detail.

Closing:

Options for closing the lesson, as time permits:

- For teachers, go over slides 38-43
    - They give possible extensions for geometry, algebra and art
- Use a website for more information.
    - This website explains how tessellations work: https://science.howstuffworks.com/math-concepts/tessellations.htm
    - This website gives directions for making an interesting free-style tessellation with paper. http://mathengaged.org/resources/activities/art-projects/tessellations/