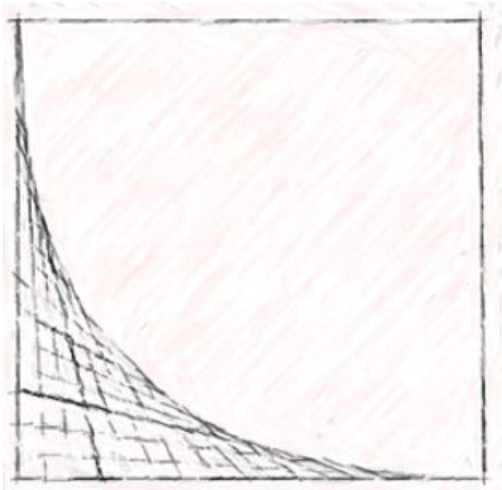
THE REIMAGINED CLASSROOM

PAINTING POETRY WITH PYTHON

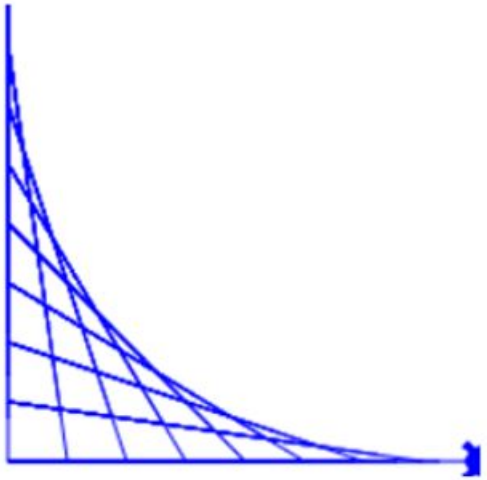
LESSON PREVIEW

Tension: *The state of being stretched tight*



```python
import turtle
tina = turtle.Turtle()
tina.color("blue")
tina.shape("turtle")
tina.speed(10)
tina.pensize(1)
x=0
y=200

while x<225:
    tina.penup()
    tina.goto(0,y)
    tina.pendown()
    tina.goto(x,0)
    x=x+25
    y=y-25
```

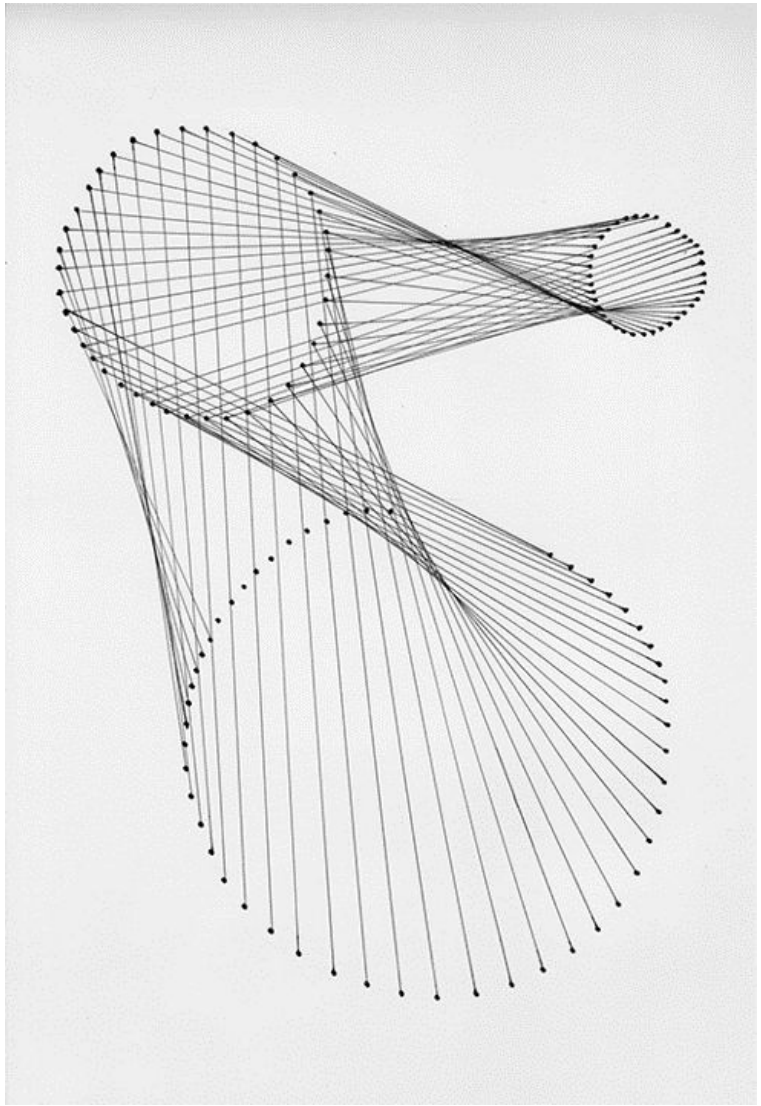


DEFINE | SKETCH | CODE | PLOT

WARM-UP

What does this drawing make you think or feel?

Why?



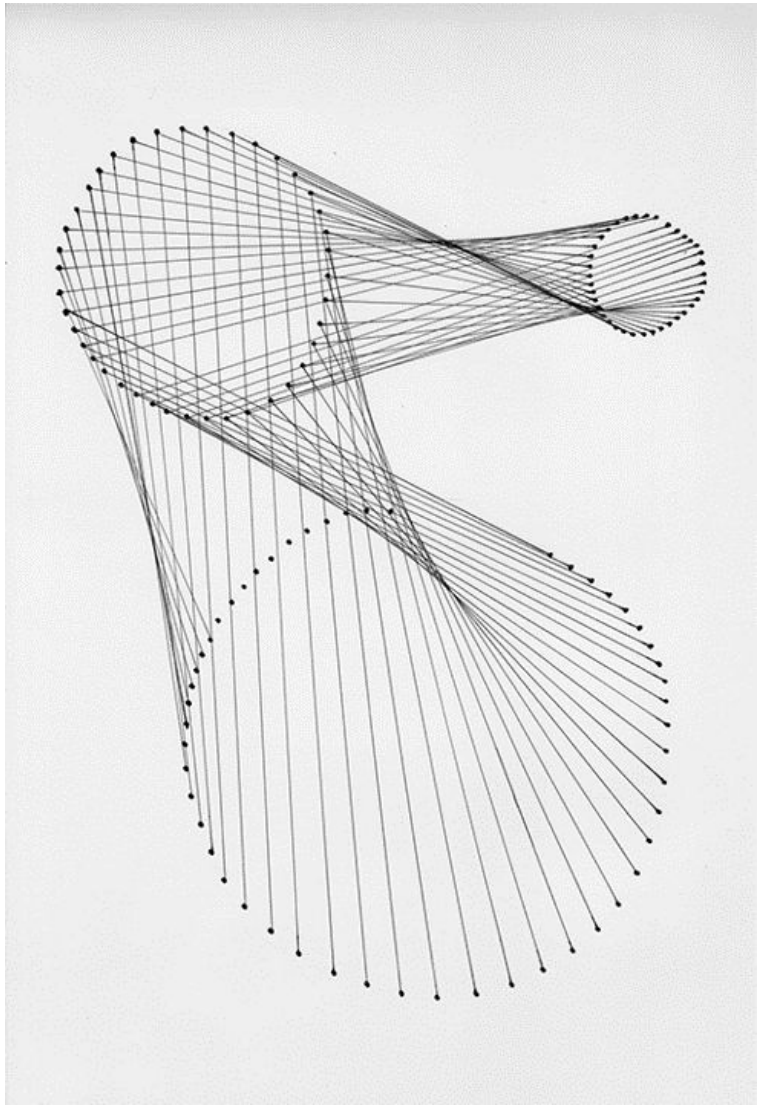
WARM-UP

Lines are a powerful tool that artists use to communicate emotions.

With only lines, perhaps this drawing made you think of:

- A vortex
- Suction
- The universe
- Spiderwebs
- Connectedness

There's no right or wrong answer. What words did you come up with?



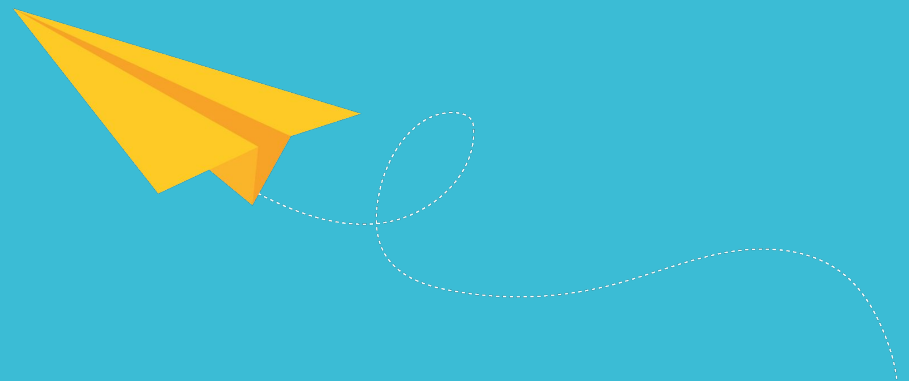
OBJECTIVES

Students will be able to:

- Draw visual representations of poetic vocabulary words using lines
- Import and use the “turtle” library in Python
- Use object-oriented programming, functions, and variables to navigate a Cartesian coordinate system
- Develop custom Python code to draw line-based images of vocabulary words

Unplugged Activity

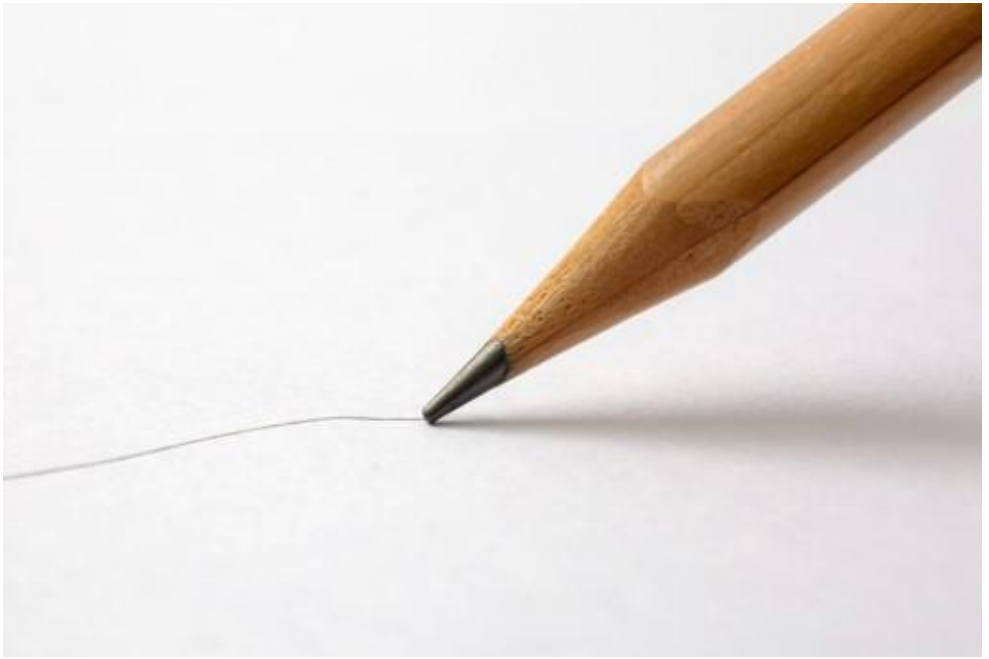
DRAWING WORDS



COMMUNICATING WITH LINES

- Oftentimes, the main goal of artists, poets, and designers is to **communicate** something (information, an emotion, etc.).

- One of the simplest, most effective tools for visual communication is the **line**.



EXAMPLE 1

Consider: How is the word **tension** used in the poem below? What images do you think of?

Mom was stressed,
Tired,
Spread thin.

Her loving smile
Stretched like a face on a balloon:
***Tension** pulling all directions.*

EXAMPLE 1

How could you draw the word **tension** using only straight lines?

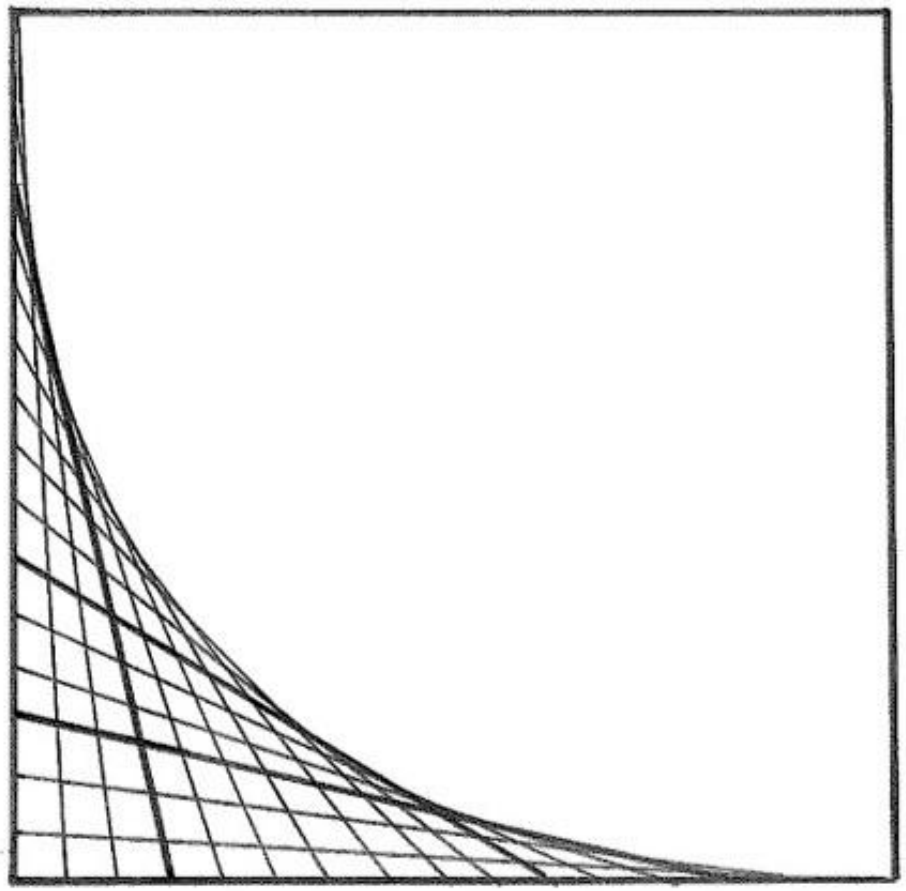
Mom was stressed,
Tired,
Spread thin.

Her loving smile
Stretched like a face on a balloon:
***Tension** pulling all directions.*

EXAMPLE 1

How could you draw the word **tension** using only straight lines?

Tension: the state of being stretched tight



EXAMPLE 2

Consider: How is the word **boldness** used in the poem below? What images do you think of?

The warrior showed strength,
confidence,
and bravery.

*She led her soldiers with **boldness**,*
unafraid of standing up
against the evil ahead.

EXAMPLE 2

How could you draw the word **boldness** using only straight lines?

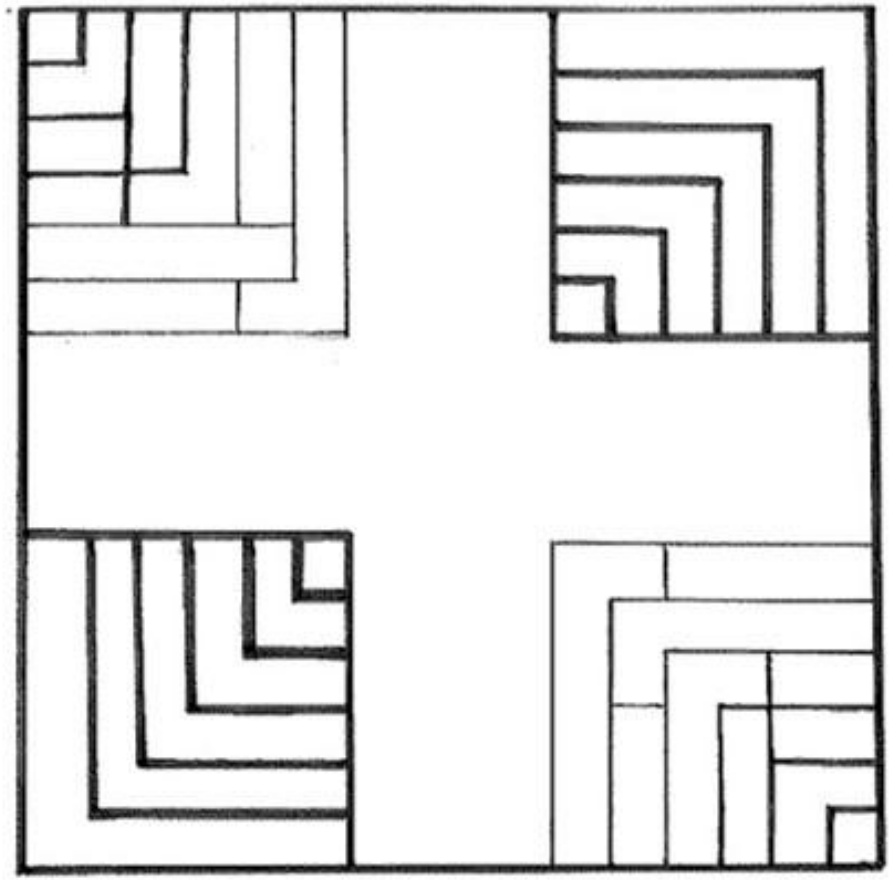
The warrior showed strength,
confidence,
and bravery.

*She led her soldiers with **boldness**,*
unafraid of standing up
against the evil ahead.

EXAMPLE 2

How could you draw the word **boldness** using only straight lines?

Boldness: willingness to take risks and act innovatively; confidence or courage



YOU TRY (5 MINUTES)

On your own, use only straight lines to sketch the definition of **graceful** with a pencil.

Do not write any words on the page.

Graceful: having or showing grace or elegance

- Calm
- Smooth
- Relaxed

YOU TRY - 2ND ROUND (5 MINUTES)

On your own, use only straight lines to sketch the definition of **aggressive** with a pencil.

Do not write any words on the page.

Aggressive: ready to attack or confront

- Angry
- Intense
- High energy

SHARE WITH A PARTNER (5 MINUTES)

Share both drawings with a partner.

Do not tell them which drawing represents which word.

Can your partner guess which drawing is **graceful** and which is **aggressive**?

How can they tell?

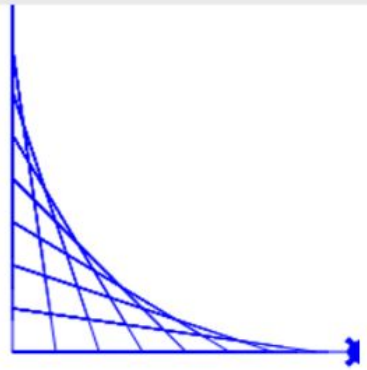
ART, ENGLISH, AND CODING

Another great tool for visual line art is **coding**.

Let's learn how to code Python scripts that draw for us!

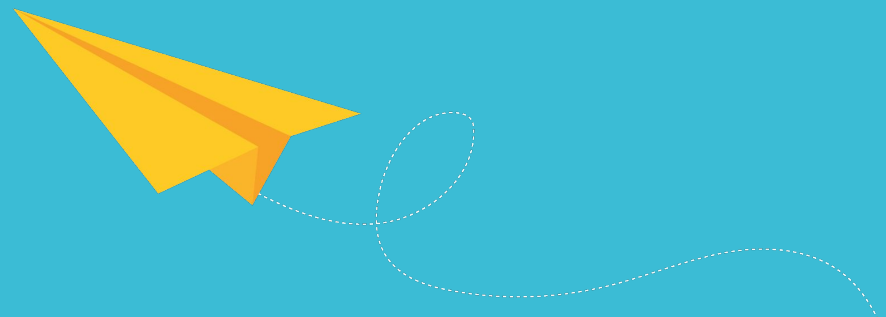
EDITOR
main.py

```
1  import turtle
2  pen = turtle.Turtle()
3  pen.color("blue")
4  pen.shape("turtle")
5  pen.speed(10)
6  pen.pensize(1)
7  x=0
8  y=200
9
10 while x<225:
11     pen.penup()
12     pen.goto(0,y)
13     pen.pendown()
14     pen.goto(x,0)
15     x=x+25
16     y=y-25
17
```



Guided Activity

PYTHON'S TURTLE LIBRARY



CODING ART WITH TURTLE

EDITOR
main.py

```
1  import turtle
2  sally = turtle.Turtle()
3  sally.color("blue")
4  sally.speed(10)
5  sally.pensize(1)
6
7  sally.forward(100)
8  sally.left(90)
9  sally.forward(100)
10
```

CODING ART WITH TURTLE

This line imports a **library** called "turtle."

A library is a special set of pre-written commands.

The turtle library lets you code a virtual pen!

EDITOR
main.py

```
1  import turtle
2  sally = turtle.Turtle()
3  sally.color("blue")
4  sally.speed(10)
5  sally.pensize(1)
6
7  sally.forward(100)
8  sally.left(90)
9  sally.forward(100)
10
```

CODING ART WITH TURTLE

Next, we can name our pen whatever we want!

Let's name her "sally."

EDITOR
main.py

```
1  import turtle
2  sally = turtle.Turtle()
3  sally.color("blue")
4  sally.speed(10)
5  sally.pensize(1)
6
7  sally.forward(100)
8  sally.left(90)
9  sally.forward(100)
10
```

CODING ART WITH TURTLE

These commands control the line **color**, drawing **speed**, and **thickness** of the pen.

EDITOR
main.py

```
1  import turtle
2  sally = turtle.Turtle()
3  sally.color("blue")
4  sally.speed(10)
5  sally.pensize(1)
6
7  sally.forward(100)
8  sally.left(90)
9  sally.forward(100)
10
```

CODING ART WITH TURTLE

This command moves the pen **forward** 100 pixels.

EDITOR
main.py

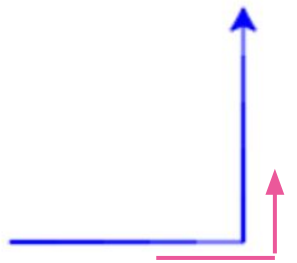
```
1  import turtle
2  sally = turtle.Turtle()
3  sally.color("blue")
4  sally.speed(10)
5  sally.pensize(1)
6
7  sally.forward(100)
8  sally.left(90)
9  sally.forward(100)
10
```

CODING ART WITH TURTLE

This command makes the pen **turn 90 degrees to the left.**

EDITOR
main.py

```
1  import turtle
2  sally = turtle.Turtle()
3  sally.color("blue")
4  sally.speed(10)
5  sally.pensize(1)
6
7  sally.forward(100)
8  sally.left(90)
9  sally.forward(100)
10
```



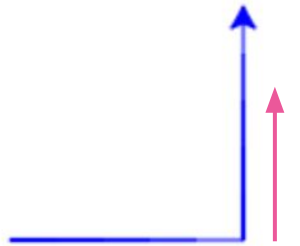
Sally turns 90 degrees to the left

CODING ART WITH TURTLE

Sally moves **forward** another 100 pixels.

EDITOR
main.py

```
1  import turtle
2  sally = turtle.Turtle()
3  sally.color("blue")
4  sally.speed(10)
5  sally.pensize(1)
6
7  sally.forward(100)
8  sally.left(90)
9  sally.forward(100)
10
```



Sally moves 100 pixels forward

BASIC COMMANDS

You know these commands:

- import turtle
- sally=turtle.Turtle()
- sally.color(“red”)
- sally.speed(3)
- sally.pensize(1)
- sally.forward(50)
- sally.left(70)
- sally.right(100)

BASIC COMMANDS

You know these commands:

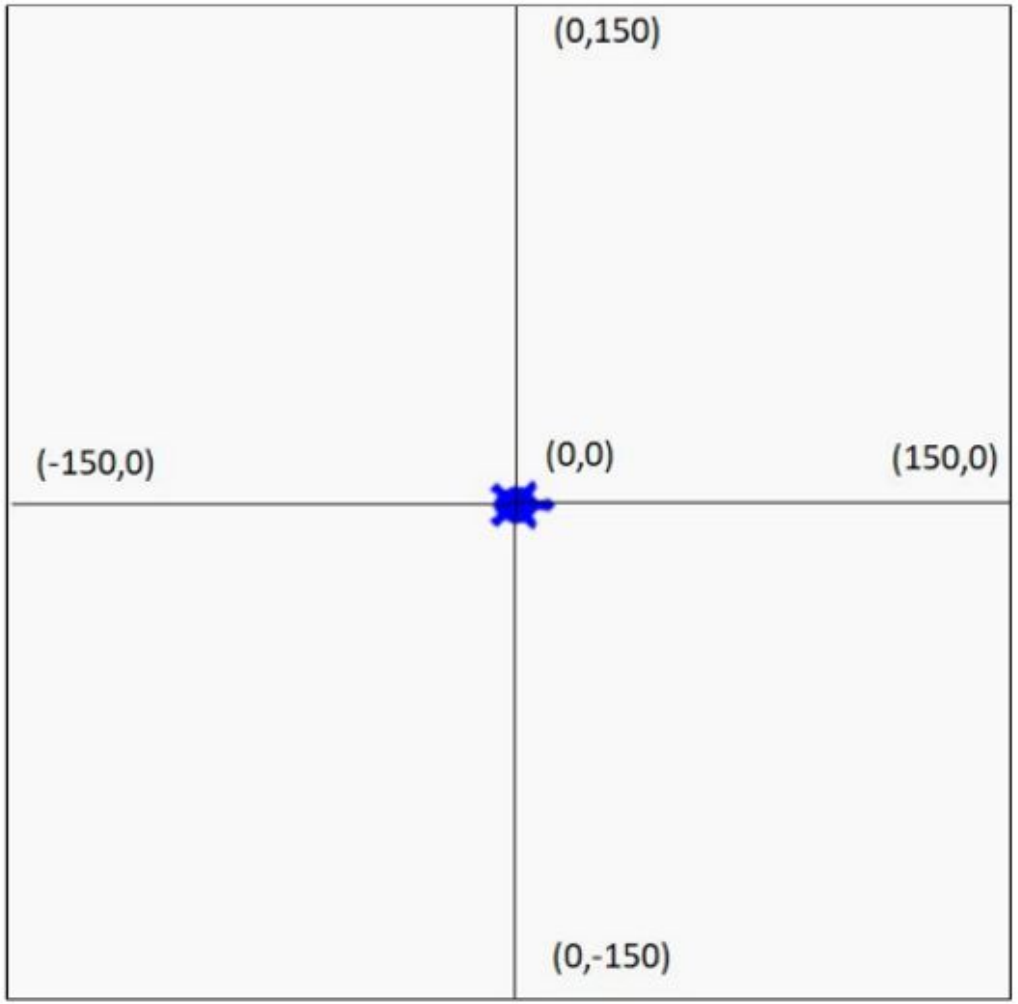
- import turtle
- sally=turtle.Turtle()
- sally.color(“red”)
- sally.speed(3)
- sally.pensize(1)
- sally.forward(50)
- sally.left(70)
- sally.right(100)

What do you think these do?

- sally.goto(x,y)
- sally.penup()
- sally.pendown()
- sally.write(“hello world”)

IT'S LIKE GRAPHING POINTS!

- sally starts at (0,0)
- **sally.goto(100,100)** would send sally to the point (100,100)
- **sally.penup()** moves sally *without* drawing
- **sally.pendown()** moves sally *while* drawing
- **sally.write("hi there")** would print "hi there" to the screen!



IT'S LIKE GRAPHING POINTS!

EDITOR
main.py

```
1  import turtle
2  sally = turtle.Turtle()
3  sally.color("red")
4  sally.speed(10)
5  sally.pensize(4)
6
7  sally.goto(100,100)
8  sally.penup()
9  sally.goto(0,100)
10 sally.pendown()
11 sally.goto(-50,-50)
12 sally.write("hi there")
```

WATCH AND TRY: USING TURTLE

Follow along with the video to:

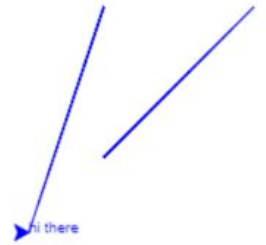
- Go to Tynker.com
- Create a new Python project
- Import the turtle library
- Draw your first shape

WATCH AND TRY: USING TURTLE

Which image would the code below draw?

EDITOR
main.py

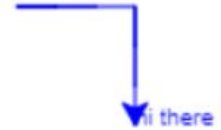
```
1  import turtle
2  sally = turtle.Turtle()
3  sally.color("blue")
4  sally.speed(10)
5  sally.pensize(1)
6
7  sally.forward(50)
8  sally.right(90)
9  sally.penup()
10 sally.forward(50)
11 sally.pendown()
12 sally.goto(0,0)
13 sally.write("hi there")
```



A

B

C



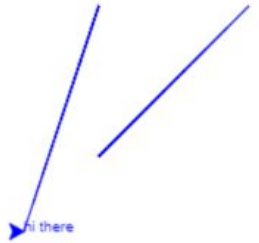
D

WATCH AND TRY: USING TURTLE

Which image would the code below draw?

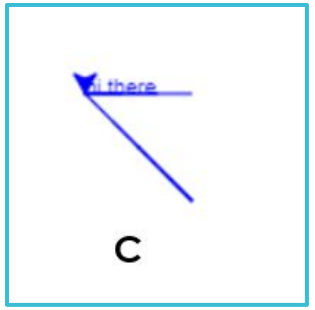
EDITOR
main.py

```
1  import turtle
2  sally = turtle.Turtle()
3  sally.color("blue")
4  sally.speed(10)
5  sally.pensize(1)
6
7  sally.forward(50)
8  sally.right(90)
9  sally.penup()
10 sally.forward(50)
11 sally.pendown()
12 sally.goto(0,0)
13 sally.write("hi there")
```

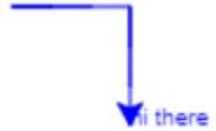


A

B



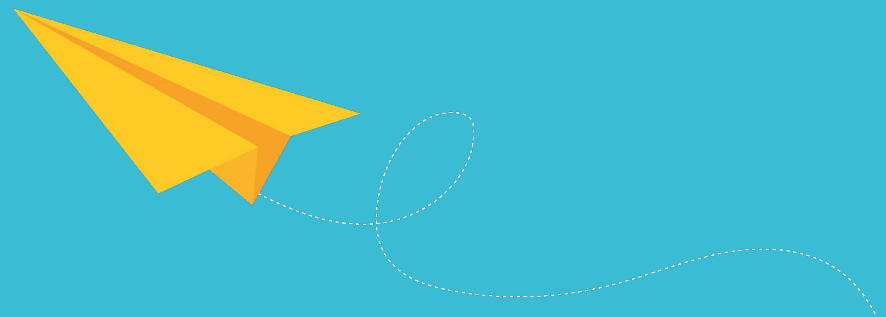
C



D

Independent Activity

CUSTOM PYTHON ART



ACTIVITY INSTRUCTIONS

Choose one word from the list on the right.
Write a simple 3–5 line poem using the word.

Then create a turtle program that draws a visual representation of the word

Your program must include:

- At least 8 different lines
- At least 2 different line colors
- At least 2 different line thicknesses
- Use of the “goto” and “forward” commands
- Use of the “penup” and “pendown” commands

Pick a word:

- United
- Focused
- Stressed
- Relaxed
- Tired
- Adventurous

EXAMPLE

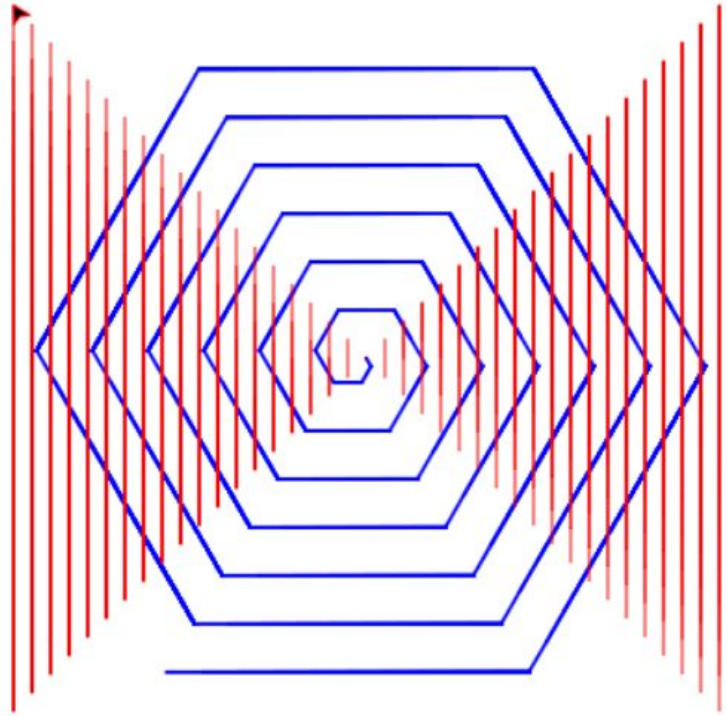
Word: **Echo** (not on list)

Definition: a sound or series of sounds caused by the reflection of sound waves from a surface back to the listener

I yelled
yelled
yelled
into the echoing cave

EDITOR
main.py

```
1  import turtle
2
3  sally = turtle.Turtle()
4  sally.pensize(2)
5  sally.pencolor("blue")
6  sally.speed(10)
7
8  for i in range(40):
9    sally.forward(i * 5)
10   sally.right(60)
11
12 sally.pensize(1)
13 sally.pencolor("red")
14
15 for i in range(20):
16     sally.penup()
17     sally.goto(10*i,10*i)
18     sally.pendown()
19     sally.goto(10*i,-10*i)
20     sally.penup()
21     sally.goto(-10*i,-10*i)
22     sally.pendown()
23     sally.goto(-10*i,10*i)
24
25 turtle.done()
```



SHARE WITH A PARTNER

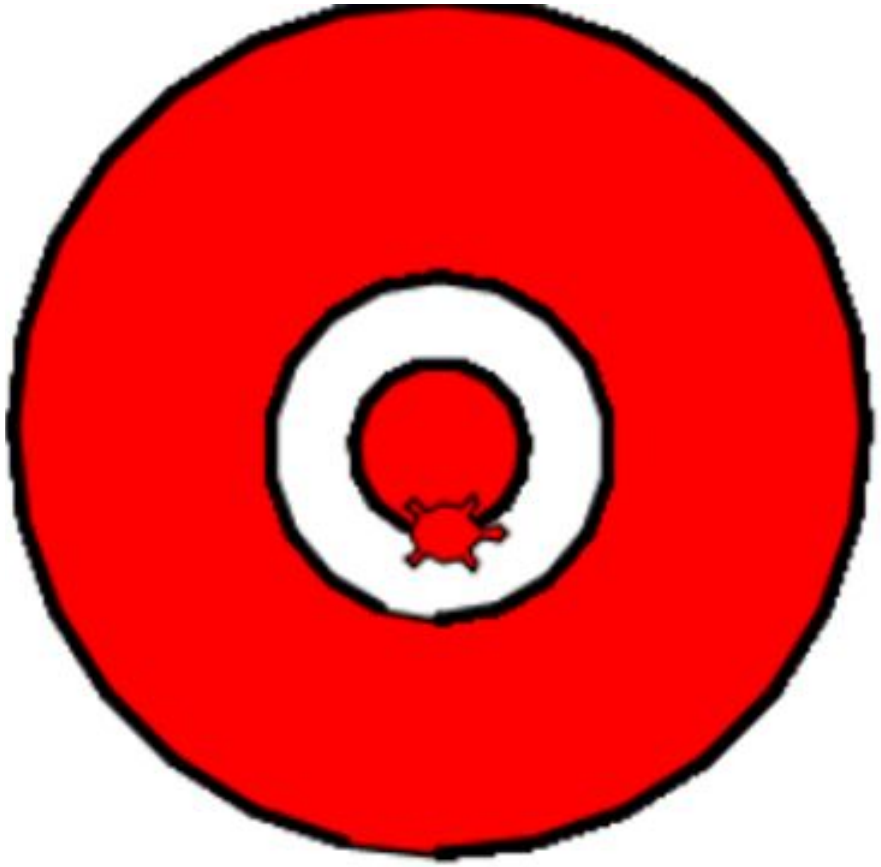
- Share your code with a partner.
- Can they guess which word you chose from the list?
- Choose a new word (not from the list) and code it together!

EXTENSION: ADVANCED COMMANDS

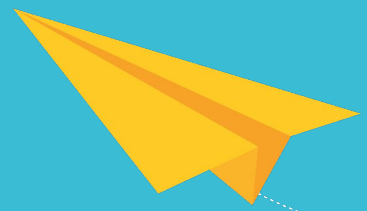
Try out this set of commands (you have to figure out what they do!):

```
sally.fillcolor("red")
sally.begin_fill()
sally.circle(100)
sally.end_fill()
```

Create another visual representation of the word **target** using these new commands.



CLOSING



REFLECTION QUESTIONS

Discuss the following questions with a partner:

1. How can we use line art to represent the emotions and definitions of words?
2. Describe how the turtle library works and its main commands.
3. What is the most challenging part of creating art with Python and the turtle library?

EXTENSION: ADVANCED COMMANDS

Juan wants to draw a **red square** using the turtle library, but he's made **three** mistakes in his code.

Identify and correct all three mistakes.

```
1  import turtle
2  sally = turtle.Turtle()
3  juan.color("blue")
4  juan.speed(10)
5  juan.pensize(3)
6
7  juan.penup()
8  juan.forward(50)
9  juan.right(90)
10 juan.forward(50)
11 juan.right(90)
12 juan.forward(50)
13 juan.right(90)
14 juan.forward(50)
15 juan.right(90)
16
```

EXTENSION: ADVANCED COMMANDS

```
1  import turtle
2  sally = turtle.Turtle()
3  juan.color("blue")
4  juan.speed(10)
5  juan.pensize(3)
6
7  juan.penup()
8  juan.forward(50)
9  juan.right(90)
10 juan.forward(50)
11 juan.right(90)
12 juan.forward(50)
13 juan.right(90)
14 juan.forward(50)
15 juan.right(90)
16
```

Rename turtle to “juan”

Change color to “red”

Change to “pendown()”

```
1  import turtle
2  juan = turtle.Turtle()
3  juan.color("red")
4  juan.speed(10)
5  juan.pensize(3)
6
7  juan.pendown()
8  juan.forward(50)
9  juan.right(90)
10 juan.forward(50)
11 juan.right(90)
12 juan.forward(50)
13 juan.right(90)
14 juan.forward(50)
15 juan.right(90)
16
```

CODE SOMETHING BEAUTIFUL

THE
REIMAGINED
CLASSROOM