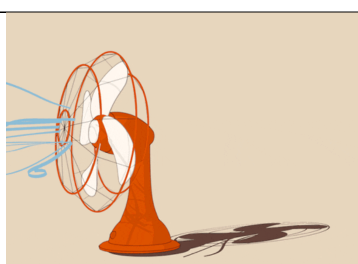
East Williamsburg Scholars Academy
850 Grand Street, Brooklyn NY 11211
Tel: 718-387-2800, Fax: 718-387-3281
Rosemary Vega, Principal

Topic: Aim:	**How can I draw with trigonometric functions in p5?**	**Materials/ Technology/Resources:**	Computer, pens, notebooks, peblio, p5.js curriculum, google classroom

Common Core Standards: *What CC standards is this lesson plan aligned to?* *CC Standards (Math)* *NextGeen Standards (Science)*	**NYS Standards** **9-12.CT.4** Implement a program using a combination of student-defined and third-party functions to organize the computation. **9-12.CT.5** Modify a function or procedure in a program to perform its computation in a different way over the same inputs, while preserving the result of the overall program. **9-12.DL.1** Type proficiently on a keyboard. **9-12.DL.2** Communicate and work collaboratively with others using digital tools to support individual learning and contribute to the learning of others.

Objective:	Students will be able to: • Call sine to create values in p5 • Use sine to create cyclical motion with map() or by multiplying values.
Essential Questions:	
How can I apply math and computation expressively to create motion graphics?	

Do Now/ Motivation:	 The fan shown on the left is oscillating. Based on what you see, what do you think it means for something to oscillate? Day #2

<table>
<tr><td></td><td>Debug Activity. Students are given a code segment and must debug the error in the code. Students must comment each line of the code to demonstrate understanding of the use the different variables used to create an oscillating shape.</td></tr>
<tr><td>Lesson Procedure:
Please explain how you will introduce the content, practice with students, and what students will complete independently.</td><td>

Time Frame – 90 Minutes

What is Sine?

Some of you may remember sine or get flashbacks of learning SOH CAH TOA in a geometry course, but for those of you who haven’t, that’s okay! What we need to know is that sine is a trigonometric function - and anytime we see tri, we should think of triangles.

Way back when the ancient Greeks (shout out Pythagoras) noticed some very interesting relationships with right triangles. Among other things, they learned that their sides have specific ratios in relation to other parts of the triangle, and these ratios occur every. Single. Time.

Sine is one of those ratios that compare the opposite side from an angle to the length of the hypotenuse to create a ratio. This ratio, when converted into a decimal value, will always be a number between -1 and 1, depending on the size of the angle.

This relationship is shown in the diagram below. Note that as the angle increases in size, so does the value of sine, but once sine hits 1, it starts heading back down to -1, and then back up towards 1.

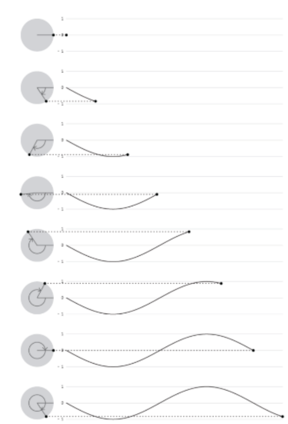


Figure 8-2. A sine wave is created by tracing the sine values of an angle that moves around a circle

Sine in P5.js

Code Along

</td></tr>
</table>

This is why we start by not having our angle update automatically - to demystify things! Once the angle starts counting up, it may be useful to use a console.log that will print the angle and result so students can see what's happening.

Because of this relationship of oscillation between 1 and -1, we can use the sin() function in p5 to create oscillating motion, but first we need to do a little more research into what kind of values sine is giving us.

The sin() function calculates the sine of an angle. This function takes into account the current angleMode(). *If you are not confident with radians and a unit circle, it's advised that you change your angleMode() to DEGREES whenever working with trig ratios!* Values are returned in the range -1 to 1.

Let's create an example and peek in the console to verify:

```
var angle = 15; //try changing this to a view different values each time you hit play.
function draw() {
  var result = sin(angle);
  console.log(result);
}
```

Right now, the result is giving you the **sine ratio of the given angle** (we can imagine that if this angle existed in a right triangle, this is the ratio the sides would be in). At the moment, this sine is pretty boring - it's static and unchanging.

If we adjust the value of the angle, we might get different numbers, but because this is a function, if you change it back to 15, you will always get the same value. The values you are getting, as expected, you get numbers between -1 and 1, like 0.9999914952150669, 0.543768153489388, -0.4123931208391132, and -0.9894020617164505.

So, how can we make the sine change? Just like in the graph, we need to update our angle to get our sine to start changing. So let's try this sample code, now:

```
var angle = 0; //start at 0 just because it's a great place to start!
function draw() {
  var result = sin(angle);
  console.log(result);
 angle += 1
}
```

We are still getting numbers between -1 and 1, like 0.9999914952150669, 0.543768153489388, -0.4123931208391132 and -0.9894020617164505. But now, they are changing each time the draw function runs, because each time we have a new angle to work with!

As we've seen in the last two lessons, consistent change is how we create movement, so this is a big step forward for us.

Sine for Movement
This new changing sine value is really exciting, and you might be thinking that it's time to dive right in and try to use this result variable in a shape! So let's try it.

```
var angle = 0; //start at 0 just because it's a great place to start!

function setup(){
 createCanvas(400,400)
 angleMode(DEGREES)
}

function draw() {
  var result = sin(angle);
  console.log(result);
 angle += 1

 ellipse(width/2,result,50);
}
```

So, what happened? You may have noticed that even though the result is changing, your ellipse is staying quite contentedly in one place, probably up near the top of your screen. Or, almost.

The result is changing, and so your ellipse is moving, but as we can see from the console, those values are really small. Like, teeny tiny. So we don't really get to see what's happening. Luckily, we can change that - we are going to take our result and change it to a larger, more useful number.

We have two ways to do this: map, which will give us some control over the final values or to scale by multiplication, which will just generally make our numbers bigger.

Oscillating Up & Down with Mapping
In the example below, we use the map() function to get an oscillating movement from sin() that we can see. Steps 1 - 3 should be familiar as we've already done them together.

1. 1.

We create a variable called angle, which starts at zero.

2. 2.

For every frame, we add 1 to it.

3. 3.

Then we calculate the sin for each angle, which will be a value between -1 and 1.

East Williamsburg Scholars Academy
850 Grand Street, Brooklyn NY 11211
Tel: 718-387-2800, Fax: 718-387-3281
Rosemary Vega, Principal

4. 4.

Finally, we map that value from the (-1, 1) range to a range that we can use to set the y position of our ball.

```
var angle = 0.0;

function setup() {
  createCanvas(600, 120);
  angleMode(DEGREES)
}

function draw() {
  background(0);
  var sineValue = sin(angle);
  var y = map(sineValue, -1, 1, 0, 120);
  ellipse(300, y, 40, 40);
  angle += 1;
}
```

It's been a while since we've used map(), so let's take a second to play. Let's make our canvas a bit bigger, and then experiment. What happens if we map sin from 1 to -1 instead of -1 to 1? Put a console.log in to help that says: console.log(sineValue, y)

Now, what if we make the 0,120 into 120,300 instead? What about 120,0, or some other numbers? Think about how we could use map() to our advantage as we start animating.

Multiple Shapes Oscillating

See the example below. You'll notice that we aren't using any type of map function here, but the shapes are still happily bouncing along! Let's figure out how they work. Try to do the following:

1. Play with the values for the different ellipses. What changes can you identify? What do the different values control? a) Change the "60" so they are all different numbers - make one 60 into 0 or take it away entirely. b) Change the "* 40" to different values. What happens? c) Alter what is being added to the angle.

2. Then, try to get all the circles to start at the center of the canvas.

Making the oscillation smaller or larger.

3.

Try using one of the y1/y2/y3 variables to control ellipse size - what happens?

4. 5.

As you work through, add comments to your sketch about what you discover.

```
var angle = 0.0;
var speed = 0.05;
var x = 0;


function setup() {
  createCanvas(600, 600);
}
function draw() {
  background(0);
  var y1 = 60 + sin(angle) * 40;
  var y2 = 60 + sin(angle + 0.4) * 40;
  var y3 = 60 + sin(angle + 0.8) * 40;
  ellipse( x + 80, y1, 40, 40);
  ellipse(x + 120, y2, 40, 40);
  ellipse(x + 160, y3, 40, 40);
  angle += speed;
```

}

Exercise: Play with Oscillation

Once you have the hang of making things oscillate, see if you can complete the following challenges:

1. Make an ellipse oscillate left and right
2. Make another element oscillate on a diagonal
3. Use sin() to control the size of something - see if you can make an emoji face that winks or screams.
4. Make an element oscillate between 50 and 200 on the x or y-axis, but no further.
5. Make a shape oscillate from lower-left corner to upper right and another shape that goes from upper left to lower right.

The given prompts in the lesson are suggestions for interesting ways to use sine that will require some logic and light math. They aren’t super interesting and don’t create a meaningful project, but it’s important for kids to try to apply these ideas in a few different situations.

We’ve tried to be mindful about making some challenges that are best solved with map(), and others that are best solved with multiplication. Leave it to students to figure out which is which or offer it as a hint once they’ve struggled for a while.

Extensions

Students can make a more cohesive project using sine!

Creating a winking emoji, emoji on a rollercoaster (screaming!), or otherwise are all great ways to make students start thinking about application over just use of a skill.

Wrap Up

- Describe what sin is and why it allows for oscillating motion.
- Why does sin need to be mapped or multiplied before it can be used?
- What is something you are struggling with or confused about related to sine?

Differentiation:

What will you differentiate?	*How will students access materials that support learning?*	*Strategies for engaged learning*	*What methods do students have to demonstrate their learning?*
	☐ *Content*	☑ *Process*	☐ *Product/Assessment*

How will you differentiate? Explain			**Formative:** Post-It Note Check-In (students answer one of the following): • Describe what sin is and why it allows for oscillating motion. • Why does sin need to be mapped or multiplied before it can be used? • What is something you are struggling with or confused about related to sine? Students are assessed on code samples for Tasks given **Summative:** Unit 4 Project, Create a Greeting Card

Assessment: *How will students demonstrate that they have met the objective?*	**Formative:** Post-It Note Check-In (students answer one of the following): • Describe what sin is and why it allows for oscillating motion. • Why does sin need to be mapped or multiplied before it can be used? • What is something you are struggling with or confused about related to sine? Students are assessed on code samples for Tasks given **Summative:** Unit 4 Project, Create a Greeting Card

Homework:	N/A

Lesson Plan Part II: Data Collection and Reflection

- Which of the Common Core Instructional Shifts were reflected in this lesson? How?
- How does this lesson reflect academic rigor?
- What evidence of cognitive engagement can you cite?
- Approximately what percentage of students were cognitively engaged?
- Approximately what percentage of students participated voluntarily? were cold-called?

East Williamsburg Scholars Academy
850 Grand Street, Brooklyn NY 11211
Tel: 718-387-2800, Fax: 718-387-3281
Rosemary Vega, Principal

- How could this lesson be improved? (i.e. technology, group activity, etc.)