



Lesson 8.6: Using Graphics in JavaScript

<https://codehs.com/course/23411/lesson/8.6>

Description	<p>In this lesson, students will learn the basics of creating graphics objects. Graphic creation relies on setting the type, shape, size, position, and color on the artist's canvas before adding to the screen. Using the geometric concepts, and the concept of <code>getWidth()</code> and <code>getHeight()</code>, multiple graphic objects can be created in JavaScript.</p>
Objective	<p>Students will be able to...</p> <ul style="list-style-type: none">• Create graphical JavaScript programs that draw shapes on the canvas• Locate points on the graphics canvas using (x, y) coordinates
Activities	<p>8.6.1 Video: Graphics 8.6.2 Check for Understanding: JavaScript Graphics Quiz 8.6.3 Example: Graphics Hello World 8.6.4 Example: Blue Circle 8.6.5 Example: Red Rectangle 8.6.6 Example: 8 Ball 8.6.7 Exercise: Flag of the Netherlands 8.6.8 Exercise: Snowman 8.6.9 Badge: Graphics Badge</p>
Prior Knowledge	<ul style="list-style-type: none">• Proficient understanding of the Cartesian Coordinate system. Students should be able to locate points in an (x,y) plane• Proficient understanding of the terms, radius, length, width, and height• Proficient understanding of initializing and assigning variables
Planning Notes	<ul style="list-style-type: none">• Review the following concepts to prepare students for drawing circles and rectangles<ul style="list-style-type: none">◦ radius and diameter in a circle◦ width and height in a rectangle◦ plotting points in an xy coordinate graph• Consider creating posters with students to think about the steps involved to draw a graphic on a screen in JavaScript by looking at patterns. Write these steps down on poster paper so that students

can reference this through the Basic JavaScript and Graphics module. An example can be:

- Declare variables
 - Determine which object to create such as new Text, new Circle
 - Determine size and color
 - Set the position of the graphic
 - Add it to the screen!
- Students might experience confusion with plotting points in the artist's canvas in JavaScript because the y-axis does not decrease but increase as it moves down the canvas.
 - Look through handout options and determine if any will be used in class or as a homework assignment.

Standards Addressed

Teaching and Learning Strategies

Lesson Opener:

- Provide students with an image of a piece of art that is based on mathematical principles, such as the [Sierpinski Triangle](#). Ask students to consider if this is art or not, and to provide their reasoning behind that. Then explain to students the basic mathematical concept related to the art that was shared, and how it was used to put this piece of art together. Explain to students that they will begin to create art based on simple mathematical principles as well. [5-10 mins]
 - Consider using the beginning of class discussion questions as well.
- As an alternative exercise, have students complete the *Do Now* handout that has students review radius, diameter and x,y coordinate planes. [5-10 mins]

Activities:

- Watch Graphics video and complete the corresponding quiz. [5- 7 mins]
 - Consider pausing the video to use the whiteboard to draw shapes at certain coordinates. Ask students to draw shapes on the board based on coordinates, sizes, and colors you give them related to how they have been brought up in the video.
- Before students walk through the examples, direct students to pair with a partner and begin experimenting with positioning and sizing of the text, circles, and rectangles. At the end of the activity, students will share their findings with the class. [8 mins]
 - Prompt students to think about the following:
 - In which direction does the text/circle/rectangle move if you change the x-coordinate? What about if you alter the y-coordinate? What is the relationship between increasing the x or y-coordinate and its position?
 - How are variables used to set positions?
- Have students explore *Graphics Hello World*, *Blue Circle*, *Red Rectangle* and *8 ball*. [6 -10 mins]
 - Encourage students to change the `setPosition` for the different examples to see how the different types of shapes

react to the changes. What point on the shape determines the location?

- Have students remove one of the commands on the canvas- what happens to the shape. Explain how important it is for them to include the correct lines of code in order for each graphic to make it onto the console screen.
- Complete *Flag of the Netherlands* and *Snowman*. [10- 15 mins]
 - Students may struggle with placing the different shapes in the correct position. Encourage them to map out their solutions on paper before getting started. These activities may also be best suited for pair programming because students are learning this concept for the first time.
 - Provide students with the *Artists Canvas* handout to give them a visual for the canvas that they are working on.

Lesson Closer:

- Have students complete the end of class discussion questions. [5 - 7 mins]
- Provide any handouts to students as an exit ticket or for homework. [5 - 10 mins]

Discussion Questions

Beginning of Class:

- Many programs generate graphics. What are some programs, apps, or websites that you use that create graphics?
 - *Answers will vary. Students may refer to Snapchat, Instagram.*
- There is a strong correlation between math and art. Can you think of examples of how this is the case?
 - *Symmetry is mathematical. Artists think about the angle of the shapes that they combine with one another.*

End of Class:

- How can we get the width and height of the graphics canvas?
 - *Using the `getWidth()` and `getHeight()` commands.*
- What are the coordinates for the top left corner of the canvas?
 - *(0,0)*
- What are the coordinates for the center of the canvas?
 - *(getWidth()/2, getHeight()/2)*
- Describe how you would make a blue circle with radius 10 and add it to the center of the canvas.
 - *Create a new rectangle, set its size, set its color, then add it to the canvas.*
- What is challenging about creating graphical programs? What is rewarding about creating graphical programs?
 - *There are a lot of steps that need to be taken before the shape can be put on the screen, and if students struggle with math, it may be hard to remember how to get shapes to exist on the canvas in the correct location.*
- How can variables be used with `getWidth()` and `getHeight()` to find the dimensions of the canvas?

- We can set the value of `getWidth()` and `getHeight()` to variables at the beginning of our program, then use those variables in our programs instead of the getter functions.

Resources/Handouts

[The Artist's Canvas](#)

[User Input and Graphics \(student\)](#)

[User Input and Graphics \(teacher\)](#)

[Variables and Graphics \(student\)](#)

[Variables and Graphics \(teacher\)](#)

[Do Now \(student\)](#)

[Do Now \(teacher\)](#)

[CLI vs. GUI \(teacher\)](#)

[CLI vs. GUI \(student\)](#)

Vocabulary

Term	Definition
Constant	A variable in a program that has a value that does not change.
Magic Number	A number in your code that appears arbitrary. These should all be replaced with calculations or constants.
Canvas	The screen in which our graphics programs are drawn.
Coordinate system	A coordinate system uses numbers as coordinates to place objects in a geometric space.
getWidth()	JavaScript function that returns the width of the graphics canvas
getHeight()	JavaScript function that returns the height of the graphics canvas

Radius	The length between the center and edge of a circle
------------------------	--

Modification: Advanced	Modification: Students Needing Additional Support	Modification: English Language Learners
<ul style="list-style-type: none"> • Attempt to draw an emoji in a Sandbox program using JavaScript. • Have students complete the <i>GUI</i> handout to learn more about the different types of ways users can interface with programs. 	<ul style="list-style-type: none"> • Provide extra copies of The Artist's Canvas in JavaScript handout so that students may draw where their rectangles and circles will be positioned on the screen. • Have students first identify the variables and functions that will be used in the exercises before any programming takes place. • Provide students with the <i>Artist's Canvas</i> handout as a visual 	<ul style="list-style-type: none"> • Define the word canvas to students beforehand. • Have student write down observations throughout their experimenting with the example exercises. • Provide students with the <i>Artist's Canvas</i> handout as a visual