



# Hour of Code: Teacher Guide

## Before the Hour of Code:

- Make sure student computers have an up-to-date browser (Chrome, Safari, or Firefox).
- Read through teacher notes in this document. Download notes to have exercise solutions ready.

## During the Hour of Code:

1. Direct students to [codehs.com/hoc\\_triangles](https://codehs.com/hoc_triangles)
2. Allow students to work through Hour of Code at their own pace, providing encouragement and support when needed. See tips below for handling student questions.
3. Tweet pictures or stories at @CodeHS #ReadWriteCode #HourOfCode!
4. If time allows at the end of the period, facilitate a discussion around the Hour of Code using the following guiding questions:
  - Before today, what did you think about programming or coding?
  - Did any of these ideas change during the Hour of Code?
  - What was your favorite part of the Hour of Code?
  - Did any parts of the Hour of Code challenge you? How?

## Hour of Code Tips:

If students get stuck or have questions, it is okay if you don't have the answer! Ask questions to activate their problem-solving skills such as:

- What can we try differently?
- What do you want the program to do?

Thank you for your dedication to Computer Science Education!

**Interested in going beyond the Hour of Code?  
Reach out to us at [hello@codehs.com](mailto:hello@codehs.com).**

# Hour of Code: Teacher Guide

In this hour of code, students will explore how to create a simulator that will calculate the area of a triangle with dimensions chosen by a user. Students will program with Tracy the Turtle in Python to make this happen. Students should have prior knowledge of basic geometry concepts before beginning this activity.

## Objective

Students will be able to ...

- Utilize functions, parameters, and variables inside a computer program
- Use variables to calculate the area of a triangle using a known equation

**Link to Activity:** [codehs.com/hoc\\_triangles](https://codehs.com/hoc_triangles)

## Discussion Questions

- How do computer simulations of real life events help scientists?
- What are the benefits of using computer simulations?
- In what ways do computer generated values make scientists lives easier?

## Exercise Solutions

Underlined 'Hi!'	
<b>Description</b>	Use the <code>write_text()</code> function and some <code>forward()</code> commands to have Tracy write 'Hi!' to the screen with a line underneath. Tracy should move forward 50 pixels, write 'Hi!', and then move 50 more pixels, drawing a line the whole time.
<b>Motivation</b>	This exercise allows students to practice using commands and calling predefined functions.
<b>Solution</b>	<pre># This function writes text to the screen where font size can be indicated def write_text():     __turtle.write("Hi!", align="center", font=("Arial", 40, "normal"))  forward(50) write_text() forward(50)</pre>
<b>Common Questions</b>	<b>Misspelled commands or wrong letter case</b> In order for the program to run, every command must be spelled correctly and must be in correct letter case. The command <code>write_text</code> must be separated by an underscore, not a hyphen, and must be in all lowercase letters.



## Hour of Code: Teacher Guide

Double 'Hi!'	
<b>Description</b>	Have Tracy write 'Hi!' on the screen twice- one should be 100 pixels behind her starting position and one should be 100 pixels in front of her starting position. She should not leave any lines as she moves to these positions. (Note: You'll need to find a way to move Tracy backward 100 pixels. There are two ways to do this! Check out the DOCS tab for some ideas!)
<b>Motivation</b>	This exercise allows students to problem solve and encourages them to use documentation to find the answer rather than a teacher or classmate.
<b>Solution</b>	<pre># This function writes text to the screen where font size can be indicated def write_text():     __turtle.write("Hi!", align="center", font=("Arial", 40, "normal"))  penup() forward(-100) write_text() forward(200) write_text()  OR  # This function writes text to the screen where font size can be indicated def write_text():     __turtle.write("Hi!", align="center", font=("Arial", 40, "normal"))  penup() backward(100) write_text() forward(200) write_text()</pre>
<b>Common Questions</b>	<b>How do I move backward?</b> There are two main ways to accomplish this task. Students can either enter negative numbers into the <code>forward</code> command, or there is a <code>backward</code> command that they can call, using 100 as the length.

Print Your Name	
<b>Description</b>	Tell Tracy to write your name to the screen with a size 30 font. Once she's done, have her move off the screen to the right so she is no longer visible. Don't leave any additional lines on the screen.
<b>Motivation</b>	This exercise has students practice calling a function using parameters.



## Hour of Code: Teacher Guide

<b>Solution</b>	<pre># This function writes text to the screen where font size can be indicated def write_text(text, size):     __turtle.write(text, align="center", font=("Arial", size, "normal"))  write_text("Student Name", 30) penup() forward(210)</pre>
<b>Common Questions</b>	<p><b>Using text as a parameter without quotation marks</b> Students need to remember that anytime text is used in their program, it needs to be surrounded by quotation marks. It is also important to note that if they put the font value inside quotation marks, that will cause an error as well so they must pay careful attention.</p> <p><b>How far do I need to move to get Tracy off the screen?</b> The center of the screen is (0, 0) and the far right-hand side is (200, 0). If Tracy moves any amount past 200, she will start to move off the screen. Students can use any value larger than about 205 to make her no longer visible.</p>

<b>Print the Date</b>	
<b>Description</b>	Create a variable called <code>date</code> that saves today's date (Note: You can write the date however you wish, but make sure it is inside quotation marks!). Print the date to the screen with size 25 font, with this font size saved inside a variable called <code>font_size</code> .
<b>Motivation</b>	This exercise has students use variables as parameters when calling functions. They may not realize, but this is putting all concepts we've learned together, and is using some pretty high level topics!
<b>Solution</b>	<pre># This function writes text to the screen where font size can be indicated def write_text(text, size):     __turtle.write(text, align="center", font=("Arial", size, "normal"))  date = "Dec. 12th 2019" font_size = 25 write_text(date, font_size)</pre>
<b>Common Questions</b>	<p><b>Parameters aren't calling the correct values</b> Students need to be sure that their variable names are not enclosed in quotation marks when they are being used inside the function call.</p>



## Hour of Code: Teacher Guide

Calculate the Area	
<b>Description</b>	<p>Fill in the <code>calculate_area</code> function to print the area of the triangle to the screen.</p> <p>Hint: The base of your triangle is set to 200 pixels. Where is the height of your triangle saved?</p> <p>You'll need to use the <code>write_text</code> function to print the value to the screen!</p>
<b>Motivation</b>	<p>Students complete the final task here by using a variable inside a mathematical equation and printing that value to the screen using parameters.</p>
<b>Solution</b>	<p>Though this is a very extensive program, the function below is the only function students should be making any alterations to:</p> <pre>def calculate_area(y):     setposition(0, -80)     write_text("Area (in sq. pixels):", 20)     setposition(0, -110)     area = .5 * 200 * y     write_text(area, 20)</pre>
<b>Common Questions</b>	<p><b>How do I write my mathematical equation?</b></p> <p>There are a few things that must be noted when entering the mathematical equation into the program. First, we must use (.5) rather than (1/2) to make sure our calculation is done correctly. Second, all items that are being multiplied together must be separated by an asterisk. This may confuse students who are very used to being able to place two values next to each other and imply they are being multiplied (ex: 2x). Third, the entire equation should be saved to a variable (here we are naming it <code>area</code>). Fourth, <code>y</code> should be included in the equation in place of the height of the triangle.</p>