

Lesson 3 - Loops and Conditionals

Part I :

Before we decide to draw a regular pentagon, we must define what a regular pentagon is.

Drawing a Regular Pentagon:

To draw a regular pentagon, keep in mind:

- ❖ The angle is _____ degrees
- ❖ A regular pentagon has _____ sides, so the steps should be repeated _____ times.

Write an algorithm that describes the step-by-step process as done in week 1.

Create a flowchart for drawing a regular pentagon. Refer to the flowcharts in week 1 as your guide. Imagine you telling a robot or computer to do these steps what would they do?

Now write a script for a regular pentagon.

Checkpoint 1

Compare your code to what you see on the screen:

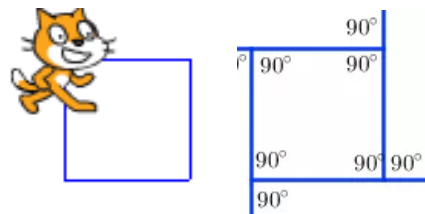
Q1: What patterns do you notice in the script? Look back at square and triangle exercises from week 1.

Part II : The Power of Loops

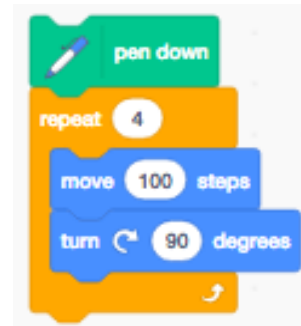
The programs are relatively simple so far, but as you start writing longer scripts, you'll often need to duplicate the same blocks several times in a row which makes the program longer and harder to understand. If you need to change one number, you'll have to make the same change in each copy of the blocks. Loops can help you avoid this problem.

For example, let's say that you want to draw a square. You could command the sprite to follow these repetitive instructions:

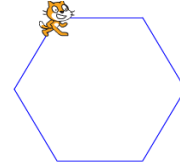
1. Move some distance and turn 90° clockwise.
2. Move the same distance and turn 90° clockwise.
3. Move the same distance and turn 90° clockwise.
4. Move the same distance and turn 90° clockwise.



In contrast, instead of using the same two over and over a **repeat** block can be used, which runs the commands inside it as many times as you tell it to, as shown in the figure.



Drawing a Regular Hexagon:



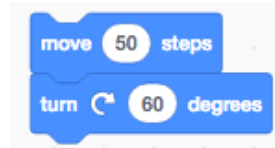
Next you're going to draw a regular hexagon.

Q2: How many degrees will the character turn in total when it draws the regular hexagon?

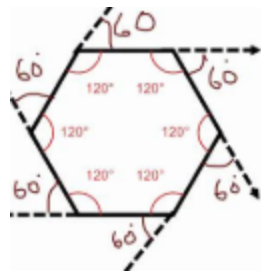
Write an algorithm that describes the step-by-step process.

Create a flowchart for drawing a regular hexagon. Remember to use a loop!

Now take a look at your algorithm and flowchart, and identify the two key steps that are repeated.



Using your knowledge of loops, code your hexagon.








Checkpoint 2

Part III: Conditionals

To find the degree of an angle in any regular polygon, you use the equation: $(n - 2) \times 180^\circ / n$

Let's use this knowledge to code for any sided shape.

- To include a user input, use the  sprite located in the  blocks. Use the  placeholder to repeat however many times the user inputs.
- The distance the sprite moves is arbitrary.
- Use the  blocks to form the equation. Remember, n is equal to .

Checkpoint 3

Test your code out with triangles, squares, etc.

What happens when you input a two sided shape?

Checkpoint 4



This work is licensed under the Creative Commons Attribution 4.0 International License.
To view a copy of the license, visit <https://creativecommons.org/licenses/by/4.0/>.