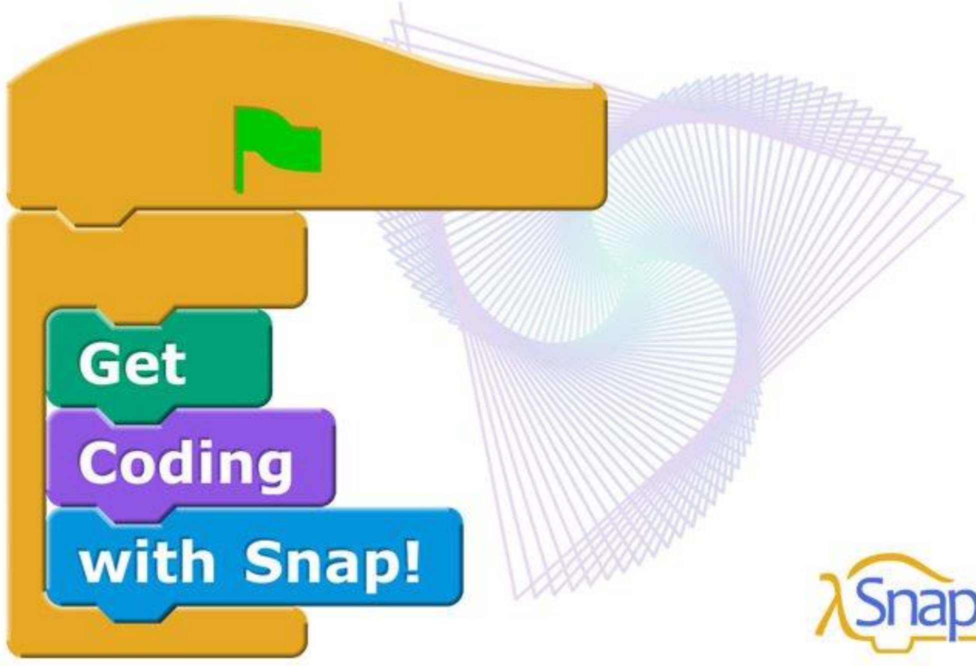
openSAP Get Coding with Snap!

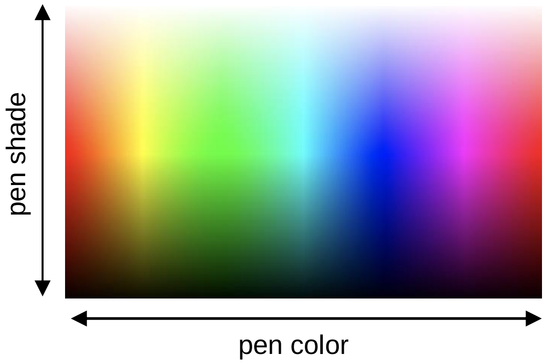
Exercises Week 1 Unit 2

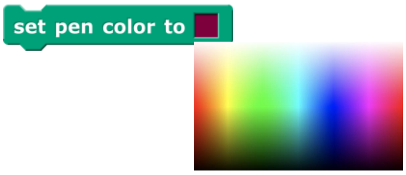


PUBLIC

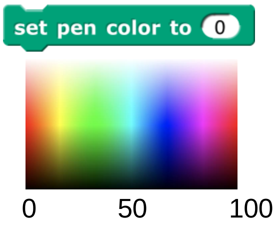
WHAT YOU HAVE LEARNED THIS WEEK

In this unit, you have learned how to draw the movement of your sprite on the stage using the blocks of the “Pen” category. If you put the “pen down”, all of the sprite’s motions are drawn onto the stage until you put the “pen up” again. You can delete everything that is drawn on the stage with the “clear” block. You can set and change the pen’s size. Additionally, you can choose a pen color with different blocks. Pen colors in Snap! consist of 2 values, the pen shade (brightness) and the pen color (hue).

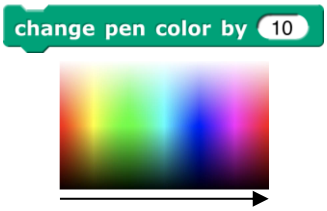




The “set pen color to (color)” block lets you pick any pen color and pen shade from the color picker. You can access the color picker by clicking on the colored square on the block



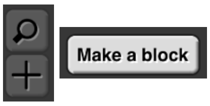
The “set pen color to (number)” block takes numbers from 0 to 100 as input. The pen shade (brightness) of the color depends on the current brightness and can be adjusted with the color picker or the “set pen shade” block



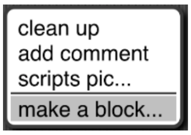
The “change pen color by (number)” block takes numbers as input and changes the pen color accordingly. It does not affect the pen shade (brightness) of the color

Additionally, you learned how to create custom blocks with Snap!’s “Make a block” feature:
If you want to use a script multiple times, you can put it in a custom block. This is especially useful when you want to use the script with different inputs. Also, it makes your script more readable.

In Snap!, there are three ways to open the Block Editor to create custom blocks. You can either click the plus sign at the top of the palette in each category or the “Make a block” button at the bottom, or you can right-click the scripting area and select “make a block” from the context menu.

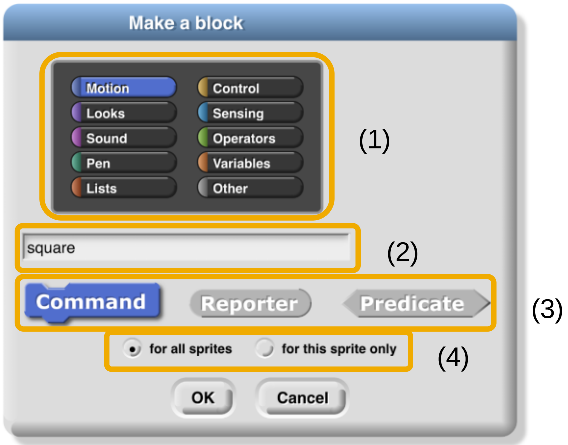


in the palette

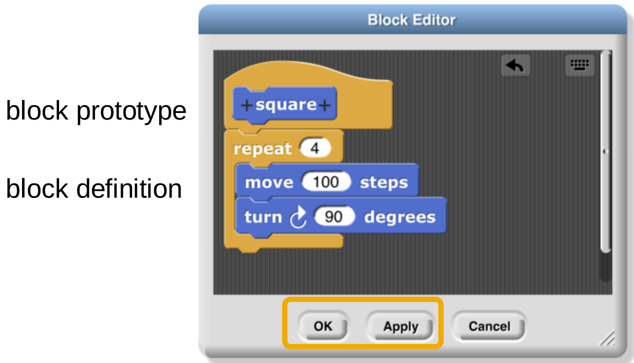


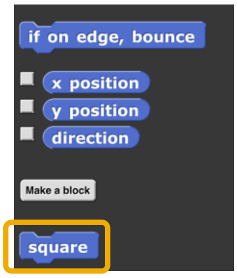
in the scripting area

In the newly opened dialogue box you first need to specify the category (1), name (2), block type (3), and availability (4) for your block. Click “OK” when you specified everything to enter the Block Editor.

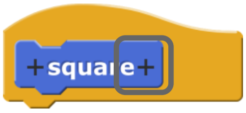


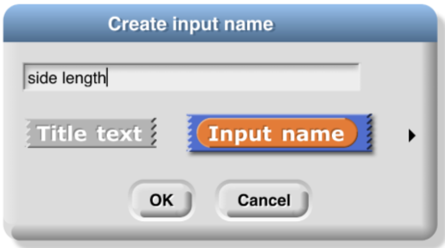
In the Block Editor, you can already see a hat block with the block name on it, the so-called block prototype. You can attach your block definition, the script that should be executed whenever that block is clicked, to the prototype. Press “OK” to save your block and close the Block Editor or “Apply” to just save your block and keep the Block Editor opened. You can find custom blocks at the bottom of the category that they are assigned to.

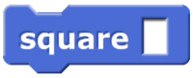


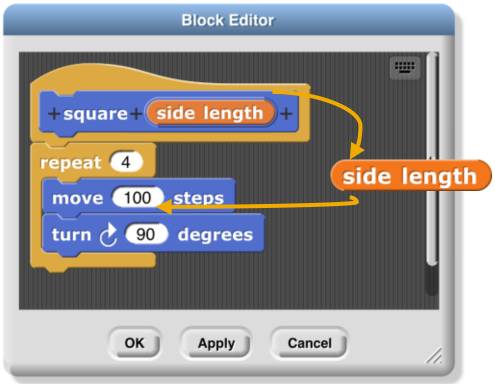


As primitives, custom blocks can also have input slots. To add an input to a block, click on the + sign on the block prototype. This opens a dialogue box, in which you can specify whether you want to expand the title text or create an input, and which name it is going to have. When you press “OK” and “Apply”, your block will get an input slot, which is represented by an orange input reporter inside the Block Editor. Whichever value you add as input to the block is stored in the orange reporter and can be used as an input for other blocks inside the block definition.



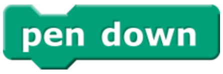
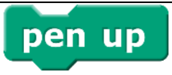
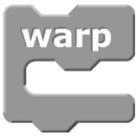
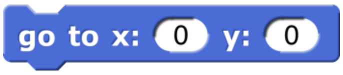
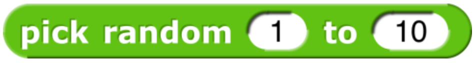






You can edit custom blocks by right-clicking the block and selecting “edit…” from the context menu.

Useful Blocks

This block puts the pen down. All movements of a sprite leave a trace on the stage	 pen down
This block puts the pen up. Sprites don't leave a trace of their movement on the stage	 pen up
The warp-block runs the script inside it without yielding to other threads or allowing redisplay, apart from breaks once every few seconds to allow sensing the stop sign being clicked and held.	 warp
This block relocates a sprite to the center of the stage	 go to x: 0 y: 0
This block reports a random number between the two boundaries indicated by the input slots. Upper and lower limit are included in the pool of numbers to pick from	 pick random 1 to 10

YOUR TURN

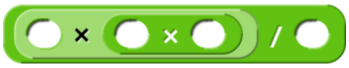
- Create a block which lets you draw circles with a specific radius. A circle is basically a polygon with an infinite number of edges. You can approximate a circle by considering it a polygon with 360 edges and a turning angle of 1°. To get a circle with a specific radius, you need to calculate the input of the move block depending on the desired radius.

  You can approximate the number of steps that you move each time with the formula of the arc length:

  $$b = \frac{\alpha}{360°} * 2 * \pi * r$$

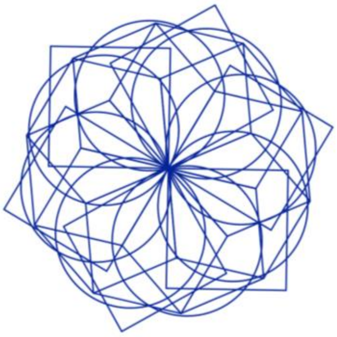
  with:
  b = arc length, number of steps you want to move each time
  α = number of degrees you turn each time (in this case 1°)
  r = your desired radius

  You can easily rebuild this formula in Snap! by nesting arithmetic operators into each other


(( ) × (( ) × ( ))) / ( )

- Combine your custom polygon block that you made during the video with the new circle block to generate mandalas like this:

  If you want, you can also change the colors of the individual shapes to make it more interesting.



Coding Samples

Any software coding or code lines/strings (“Code”) provided in this documentation are only examples and are not intended for use in a production system environment. The Code is only intended to better explain and visualize the syntax and phrasing rules for certain SAP coding. SAP does not warrant the correctness or completeness of the Code provided herein and SAP shall not be liable for errors or damages cause by use of the Code, except where such damages were caused by SAP with intent or with gross negligence.

www.sap.com/contactsap

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE’s or its affiliated companies’ strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

THE BEST RUN SAP