

An Introduction to the Snap! Coding System

This initial module provides an orientation to the Snap! programming language. Snap! was developed at the University of California, Berkeley, and is used in computer science (CS) for non-CS majors at that university. Because of its widespread use, a user community has developed around this programming language. The capabilities of this language make it well-suited to explorations in art and music.

Topic 0.1 Securing a Snap! Account

A free Snap! account can be obtained from the University of California, Berkeley web site:

<https://Snap.berkeley.edu/Snap>

Secure a Snap! account before continuing. The Snap! reference manual is available here:

<https://Snap.berkeley.edu/Snap/help/SnapManual.pdf>

Section II of the reference manual provides information about securing a Snap! account and saving projects. Review this section before continuing to the explorations that follow below.

Snap! help forums maintained at the University of California, Berkeley, are available here:

<https://forum.Snap.berkeley.edu/>

Assistance with questions that are not addressed in the reference manual can be obtained through the Snap! forum.

Topic 0.2 The Snap! Workspace

In a typical Snap! session, blocks of code are dragged from the *Code Block Palette* on the left-hand side of the screen to a *Script Area* in the middle of the screen. Blocks of code are snapped together to create scripts. Clicking a group of code blocks causes the script to run, performing an action. Often these actions involve movement of sprites on a *Stage* at the right-hand side of the screen.



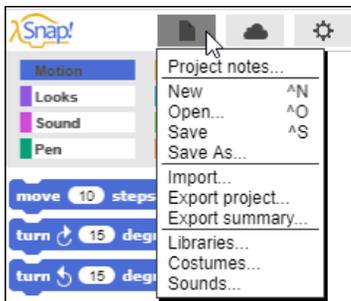
A *sprite corral* beneath the stage indicates which sprite is currently selected. Each sprite has its own separate script space.

Topic 0.3 Snap! Menus

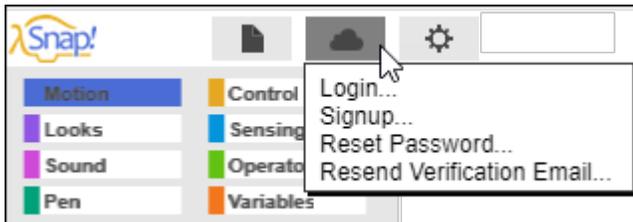
There are several menus that can be accessed in the top left-hand corner of the Snap! screen. The Snap! icon in the top left-hand corner can be used to access the Snap! reference manual.



The *File* menu to the right of the Snap! icon can be used to save projects and open projects that have been previously saved. This menu can also be used to access costumes for sprites, sounds, and libraries of additional Snap! code blocks.



The *Login* menu is to the right of the *File* menu.



A user must be logged into Snap! in order to save projects. Therefore it is a good idea to log in to Snap! at the beginning of every session.

A *Settings* menu is to the right of the *Login* menu. This menu provides a number of options for customizing Snap! For example, the *Zoom Blocks* option can be used to increase the size of the code blocks so that they can be more easily viewed in presentations.

Exploration 0.3

If you have not already, log into Snap! Explore some of the options available under the different menu settings.

Topic 0.4 The Code Block Palette

Scripts are created in Snap! by snapping blocks of code together (such as the **Move 10 Steps** code block in the illustration). The types of *code blocks* available are displayed in a *Code Block Palette* at the top left-hand side of the screen. For example, the *Motion* code blocks are currently highlighted in the palette below. Other categories of code blocks include *Looks*, *Sound*, *Pen*, *Control*, *Sensing*, *Operators*, and *Variables*. Each category is a different color (e.g., *Motion* code blocks are blue). Click on the different categories (Motion, Looks, etc.) to access the code blocks associated with that category.



The *Motion* code blocks direct the movement of sprites (actors that can move about the stage on the right-hand side of the screen.) The *Looks* code blocks control the appearance of sprites. The *Sound* code blocks are used to play sounds. The *Pen* code blocks control the color and thickness of the turtle's pen. The *Control* code blocks provide control structures such as the **Repeat** command. The *Sensing* code blocks are used to sense the status of Snap! objects and monitor external inputs such as the keyboard and the microphone. The *Operators* code blocks provide mathematical and logical functions. The *Variables* palette is used to create and modify variables.

Exploration 0.4 The Code Block Palette

Click on each of the categories in the *Code Block Palette* to get a sense of the types of commands that are found under each category.