






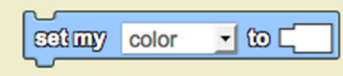
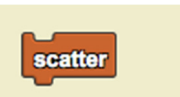

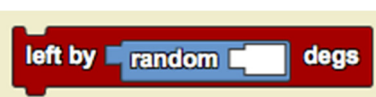
## StarLogo Nova Blocks introduced in Module 1 Lesson 1

### Event Handling


	<p>When “setup” (on the interface) is clicked, code blocks placed inside this block will run once.</p>
	<p>When “forever” (on the interface) is clicked it runs code blocks inside this block until “forever” is clicked again.</p>

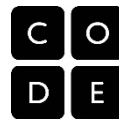
## StarLogo Nova Blocks introduced in Module 1 Lesson 2



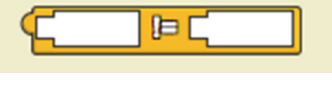
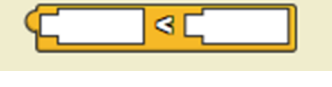


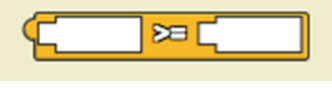
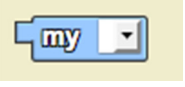
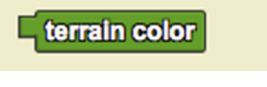

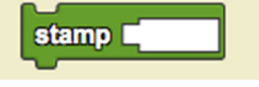
	<p>Executes commands when the push button widget is pressed. Widgets can be created using the <i>Edit Widgets</i> feature.</p>
	<p>Executes commands when the toggle button widget is turned on. When the toggle button widget is turned off, nothing occurs. Widgets can be created using the <i>Edit Widgets</i> feature.</p>
	<p>Removes all stamping or drawing modifications done to the terrain.</p>
	<p>Adjusts the pen tool. If the pen is down, the agent stamps the terrain at each position as it moves along the terrain. If the pen is up, the agent does not stamp the terrain.</p>
	<p>Deletes all agents.</p>
	<p>Creates a specific number of agents of the specific breed. Attach other blocks to specify traits and properties of these agents. Additional breeds are created in the interface.</p>

	<p>Moves an agent forward by a certain number of steps.</p>
	<p>Moves an agent backwards by a certain number of steps.</p>
	<p>Rotates an agent left by a certain number of degrees.</p>
	<p>Rotates an agent right by a certain number of degrees.</p>
	<p>Sets the color of an agent by picking from a list of standard colors or by choosing a color at random.</p>
	<p>Sets an agent's trait to a particular value.</p>
	<p>Places agents in random positions on the terrain.</p>
	<p>Returns a random integer from 0 to the input but NOT including the number entered (non-inclusive).</p>
	<p>Random can be used in place of a number anywhere a number would fit. In this example, instead of having a constant turn angle, an agent following this command would turn different amounts each time this command was executed.</p>

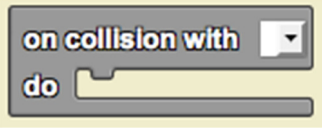

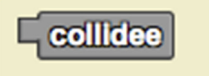

### StarLogo Nova Blocks introduced in Module 1 Lesson 3

	<p>Carries out the specific procedures if the condition is true. This conditional block <b>ONLY</b> runs the main block <b>IF</b> the condition after the “if” is met.</p>
---	--





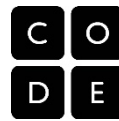
	<p>Carries out the specific procedures if the condition is true. Otherwise (if the condition is false) carries out procedures in the second block.</p>
	<p>Determines whether the two inputs are equivalent and returns true or false. Usually used after an “if” in a conditional block.</p>
	<p>Determines whether the two inputs are unequal and returns true or false. Usually used after an “if” in a conditional block.</p>
	<p>Determines if the first input is less than the second input and returns true or false. Usually used after an “if” in a conditional block.</p>
	<p>Determines if the first input is greater than the second input and returns true or false. Usually used after an “if” in a conditional block.</p>
	<p>Determines if the first input is less than or equal to the second input and returns true or false. Usually used after an “if” in a conditional block.</p>
	<p>Determines if the first input is greater than or equal to the second input and returns true or false. Usually used after an “if” in a conditional block.</p>
	<p>Returns an agent’s trait.</p>
	<p>Returns the color of the terrain the agent is currently on.</p>
	<p>Colors a grid square on the terrain in the agent’s current position.</p>
	<p>Colors a circle on the terrain in the agent’s current position.</p>


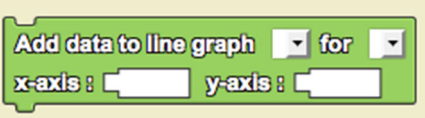


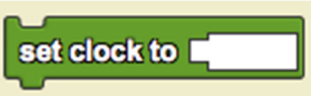
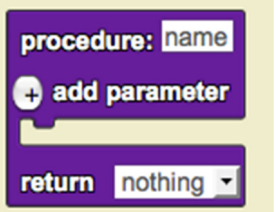
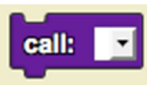
### StarLogo Nova Blocks introduced in Module 1 Lesson 4

	<p>Specifies a series of actions to be executed after a collision between two breeds. If the two breeds are the same, each agent in the collision must perform the same action after the collision. If the two breeds are different, each agent in the collision can be given different commands.</p>
	<p>'Trait of' block is used to specify which trait of the agent it should check for (X, Y, Z, color, size, heading, shape or a trait that you have added)</p>
	<p>Specifies the other agent in the collision.</p>
	<p>Counts the number of agents of a specific breed in a given radius with a specific trait. This can be used to detect other agents that are nearby and have specific traits.</p>

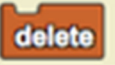
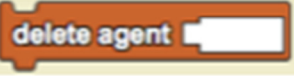
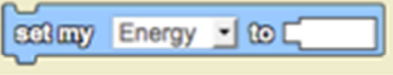
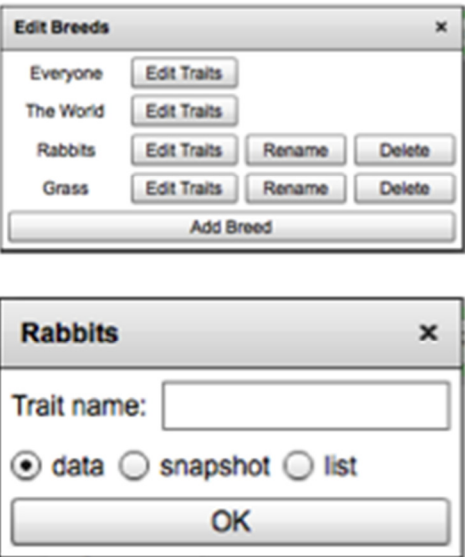
### StarLogo Nova Blocks introduced in Module 1 Lesson 5

	<p>Push this button to get access to the widgets. Widgets are user-input and output elements like buttons, data output, tables, and graphs.</p>
	<p>This button appears AFTER clicking "Edit Widgets." Select the type of widget you'd like to add. Give the widget a name.</p> <ul style="list-style-type: none"> <li>"Push button" – a button that runs once or pops up after being pushed</li> <li>"Toggle button" – a button that stays on until turned off</li> <li>"Data Box" – a data display box</li> <li>"Label" – a place for the user to enter descriptive text</li> <li>"Horizontal Slider" – a slider to control the value of a variable</li> <li>"Table" – a configurable data table with rows and columns</li> <li>"Line Graph" – a configurable line graph with one or more lines</li> <li>"Bar Graph" – a configurable bar graph with one or more bars</li> </ul>

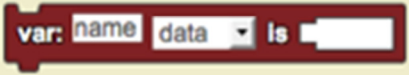
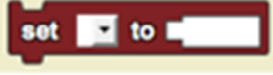



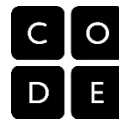
	<p>After creating a data box, use this block to set its value to a number or a color.</p>
	<p>After creating a line graph widget and naming a series (on the interface), this block adds data to the line graph. It will plot the point x, y based on the blocks that you provide. Usually these values are variables that you have defined.</p>
	<p>After setting up a slider widget, you can use the value of the slider in your code using this block.</p>
	<p>Returns the current value of the clock. This can be used when you plot data vs. time.</p>
	<p>Changes the value of the clock. This block is particularly useful for resetting the clock on setup.</p>
	<p>Creates a procedure with a name and a list of commands. Procedures are useful for organizing code into reusable modules. Some procedures just make changes to the agents and the world and return nothing. Other procedures calculate values and return the result.</p>
	<p>Calls the procedure selected.</p>

## StarLogo Nova Blocks introduced in Module 3 Lesson 2

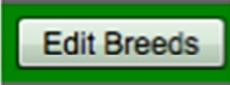
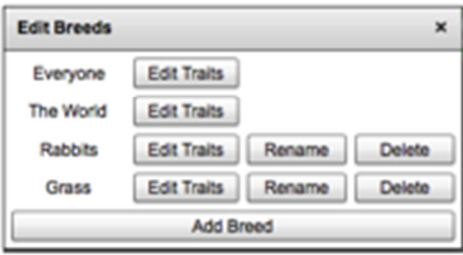

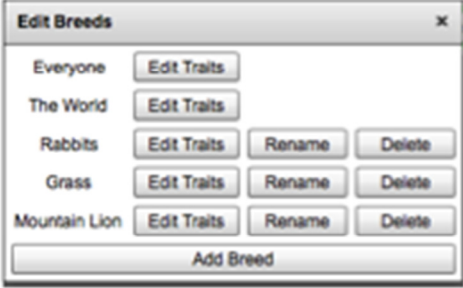
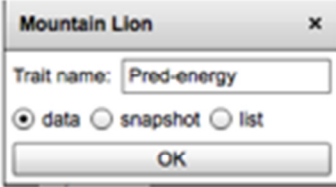

	<p>The delete block deletes the current agent.</p>
	<p>The delete agent block deletes the agent referenced.</p>
	<p>The set my (trait) to (value) block is used to set the current agent's trait to some value. In this case, Energy is a new trait created by the user.</p>
	<p>The user created the Rabbits trait "Energy" using the Edit Breeds panel. Click on "Edit Breeds" in the Spaceland panel. Then click on the "Edit Traits" button next to Rabbits. From here you can give an agent a new trait by clicking on "Add Trait"</p> <p>Then specify a trait name and type. The type can either be "data", "snapshot", or "list".</p> <p>For Energy, we want to store a value or number, so choose "data" and click OK.</p> <p>In computer science, this is called declaring a variable. The variable in this case is a number variable called "Energy".</p>

## Optional StarLogo Nova Blocks to use in Module 3 Lesson 2

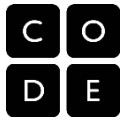
	<p>Creates a variable of the type indicated and sets its value.</p>
	<p>After creating a variable, this block sets the value.</p>
	<p>Returns the value of an existing variable.</p>



StarLogo Nova Blocks introduced in Module 3 Lesson 3

	<p>Use the Edit Breeds button in the <u>Spaceland</u> area to create a new breed.</p>
	<p>Click on "Add Breed" to add a new breed.</p>
	<p>In the "New Breed" dialog box, give the breed a name and press OK.</p>
	<p>You will see the Breed on the next panel. (We added the breed called "Mountain Lion".</p> <p>Next, click Edit Traits next to the new breed label for "Mountain Lion"</p>
	<p>Create a new trait (or variable) for the Mountain Lion breed called "<u>Pred-energy</u>".</p>
	<p>The set my (trait) to (value) block is used to set an agent's trait to some value. In this case, the Mountain Lion's <u>Pred-energy</u> is being set to a starting value. This is called "initializing" the variable.</p>





## CS Concepts introduced in Module 1 Lesson 1

### CS Concept: Event Handling

What is an event?	An event is an action or occurrence detected by a computer program. Events can be user actions, such as clicking a mouse button or pressing a key.
What does it mean to handle an event?	To “handle” an event is to respond to an event. An event handler is typically a piece of computer code that process actions such as keystrokes and mouse movements. In StarLogo Nova, each breed can react to events independently. For example, when the “forever” button is pushed, the rabbits can eat, the grass can grow, and the wolves can hunt.

## CS Concepts introduced in Module 1 Lesson 2

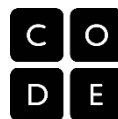
### CS Concept: Instructions / Programs

What is an instruction?	A computer instruction is a simple command given to a computer in a computer language. Computer instructions are called “commands” in StarLogo Nova.
What is a computer program?	A computer program is a sequence of instructions that a computer runs or “executes.”

### CS Concept: Iteration / Looping

What is an iteration?	An iteration is a repetition of a process. Each iteration of a loop is a single time through the loop.
What is a loop?	A loop is a type of instruction that performs a set of enclosed instructions over and over again. There are different types of loop instructions such as conditional loops, iterated loops, and infinite loops.





### CS Concept: Public and Private Repositories / Galleries

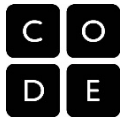
What is a public gallery?	A public gallery is a space where others can view your projects.
What is a private gallery?	A private gallery is a space where you can store your projects and only you can see them.

### CS Concept: Remixing

What does it mean to REMIX a project?	Remixing a project means that you are copying an existing project (either one you have created or one someone else created) and are saving a version of it as your own.
Is it cheating to REMIX a project?	No, if you acknowledge that you have remixed from another source and document what you added, changed, or removed to make it your own, it is not cheating. The important thing is to learn from other projects.
Why remix?	We can learn and build off of each other's projects. Remix culture allows and encourages combining and/or editing existing materials to produce a new product. Remixing is a way to learn from others, improve upon, change, integrate, or otherwise repurpose existing work.

### CS Concept: Random Numbers

What is a random number?	A random number is a number that appears to have been selected randomly from a set.
What are random numbers used for?	We generate and use random numbers any time we want to simulate a chance occurrence or chance behavior. For example, when two agents meet, there might be a chance that one passes a disease to the other. Say there is a 40 percent chance of transmission. Instead of rolling a 10-sided die, we might ask for a random number between 1 and 10. Then if we get a 1, 2, 3, or 4, which should happen 40 percent of the time, the disease would spread from one agent to the other.



## CS Concepts introduced in Module 1 Lesson 3

### CS Concept: Conditional Instruction

What is a conditional instruction?	A conditional instruction is a type of instruction that may, when executed by a computer, cause the computer to branch or begin execution of a different instruction sequence. Conditional instructions contain Boolean expressions. Depending on whether a Boolean expression evaluates to true or false, a different sequence of instructions is executed. IF/THEN and IF/THEN/ELSE are examples of conditional instructions.
------------------------------------	---

### CS Concept: Boolean Expressions

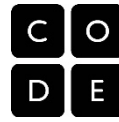
What is a boolean expression?	A Boolean expression is a mathematical expression that evaluates to one of two values (usually true or false).
-------------------------------	--

### CS Concept: Nesting Conditionals

What does it mean to nest conditionals?	When we layer conditionals within other conditionals we call that “nesting.” Nesting of conditionals can be used to implement more complicated logical constructs.
---	--

### CS Concept: Algorithms

What is an algorithm?	An algorithm is a step-by-step set of instructions for performing calculations or behaviors. Algorithms are used for problem solving, calculation, data procession, and automated reasoning,
-----------------------	--



## CS Concepts introduced in Module 1 Lesson 4

### CS Concept: Collision Detection

What is collision detection?	Collision detection is the determination of whether or not virtual objects are touching or overlapping.
------------------------------	---

### CS Concept: Model

What is model?	A model is a representation of a system, process or object.
----------------	---

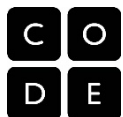
### CS Concept: Simulation

What is simulation?	Simulation is the imitation of the operation of a real-world process or system over time. The act of simulating something first requires that a model be developed; this model represents the key characteristics or behaviors/functions of the selected physical or abstract system or process. The model represents the system itself, whereas the simulation represents the operation of the system over time.
---------------------	---

## CS Concepts introduced in Module 1 Lesson 5

### CS Concept: Widgets

What is a widget?	A widget is a user interface element. Typically, a widget gets input from the user or displays output to the user. In StarLogo Nova, the user interface widgets include buttons, sliders, data boxes, graphs, and charts.
-------------------	---



### CS Concept: Variables

What is a variable?	<p>A variable is a container for a value. The value held in the variable can be initially set and later changed.</p> <p>Agent variables can be created using edit breeds. Global variables are created using add widgets, specifically, adding data boxes and sliders.</p>
---------------------	--

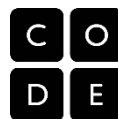
### CS Concepts introduced in other Modules

#### CS Concept: Abstraction and Decomposition

What is abstraction?	<p>Abstraction is the process of making something easier to understand by ignoring some of the details that may be unimportant. This allows you to focus on what is important at this time to you. This doesn't mean that what you ignore is not important. It's just not important to what you're doing at that moment. Computer scientists talk about the levels or layers of abstraction.</p>
When do you use abstraction?	<p>In modeling, we use abstraction when we decide what we want to include in our model and what can be left out.</p>

#### CS Concept: Data

What are data?	<p>Data, in modeling and simulation, usually refers to numbers that are generated by running our model. Data (plural) can be numbers such as the number of rabbits in a population over time.</p>
How do we collect it?	<p>We collect and visualize data using the graphs and tables in StarLogo Nova. We can see the changing values of different variables over time. Later, we will be able to output the data to a file.</p>



### CS Concept: Data Analysis














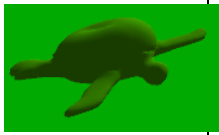
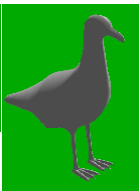







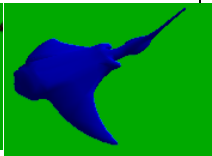
<p>What is data analysis?</p>	<p>Data analysis is the process of examining data to find trends or patterns. We use mathematical descriptions such as mean, median, and mode to describe characteristics of data sets. We can also use terms such as linear, non-linear, exponential, and logistic to describe patterns of change in time series data.</p>
-------------------------------	---



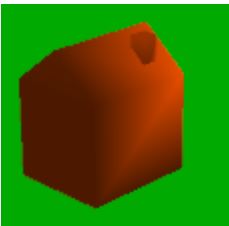
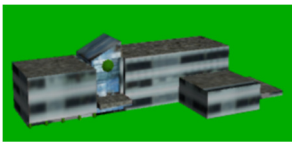
### CS Concept: Breeds, Agents, and Traits


<p>What is a breed?</p>	<p>A breed is a prototype for an agent. All agents inherit their type and traits from a breed. Breeds are defined using the “edit breeds” panel. After naming the breed, new traits can be added. When we create an agent, we must specify which breed or class of agent it is derived from.</p>
<p>What is an agent?</p>	<p>An agent is a virtual individual or object implemented in code. Agents have variables or traits that hold their characteristics or state. Agents also have behaviors, usually implemented as procedures.</p>
<p>What is a trait?</p>	<p>A trait is an agent variable. Agents have traits that they are created with by default such as color, breed, shape, and size. New traits can be added using the “edit breeds” panel.</p>





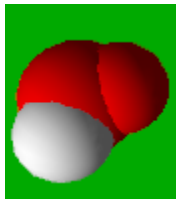
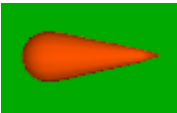



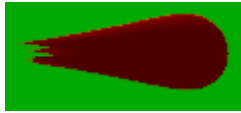


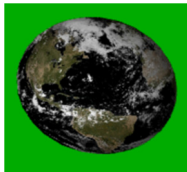
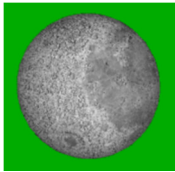

### CS Concept: Decomposition and Procedures

<p>What is decomposition?</p>	<p>Decomposition is the breaking down of a big problem into a set of smaller problems or breaking a process into different smaller steps.</p>
<p>What is a procedure?</p>	<p>Procedures are stacks of commands that perform a particular function and can be given a name. Procedures must be “called” or “invoked” to execute or run. When we decompose a problem into smaller problems, sometimes the solutions to the smaller problems are implemented in procedures.</p>



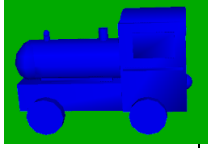




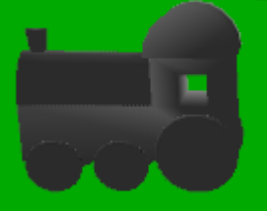
Animals					
Angel-fish 	Bear 	Clownfish 	Dog 	Elephant 	Giraffe 
Killer-whale 	Lion 	Lioness 	Otter-float 	Otter-stand 	Penguin* 
Raccoon 	Sea-turtle 	Seagull 	Seahorse 	Shark 	Simple Turtle 
Tuna 	Turtle2* 	Urchin 	Dragon* 	Mantaray 	

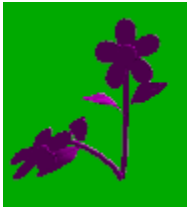
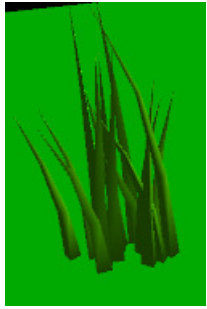



Buildings			
Castle* 	Freecastle* 	House 	Hospital* 

People					
fireman*					

Objects					
Carrot (need to go up by 2 to see it all) 	Enzyme 	Hydrant 	Phospholipid 	Water-molecule 	Fireball1 
Gem 	Coin-silver 	Coin2* 	Fireball2 	Fireball* 	Earth* 
Earth-Clouds* 	Moon* 	Starch 			



Vehicles					
Helicopter*	Lambo	Train	Truck*	Boat*	Spaceship1
					
Spaceship2	Train_wood				
					

Landscape					
Flower	Grass	Kelp	Rock	Tree*	
					

\* set color trait to "white" for original texture in these shapes

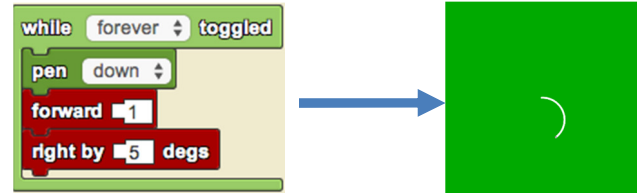
## Left and Right degrees turning: *Making your turtle turn while it is moving*

The movement blocks 'right by' and 'left by' are used to make your turtle agents turn while walking. The amount they turn is *relative* to where they are facing.



### Thinking Challenge:

Look at the code to the right. What will happen if you have your turtles turn by a number larger than 5?

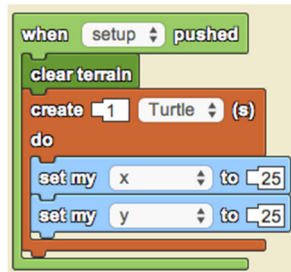


## X & Y coordinates in Spaceland: *Placing your turtles on Spaceland*

This is the standard view of the area called 'Spaceland' and is the terrain where the programmed action will take place. There is 1 turtle that is created and the **default location for the turtle is (0,0)**. The **default color and shape is a gray square**.

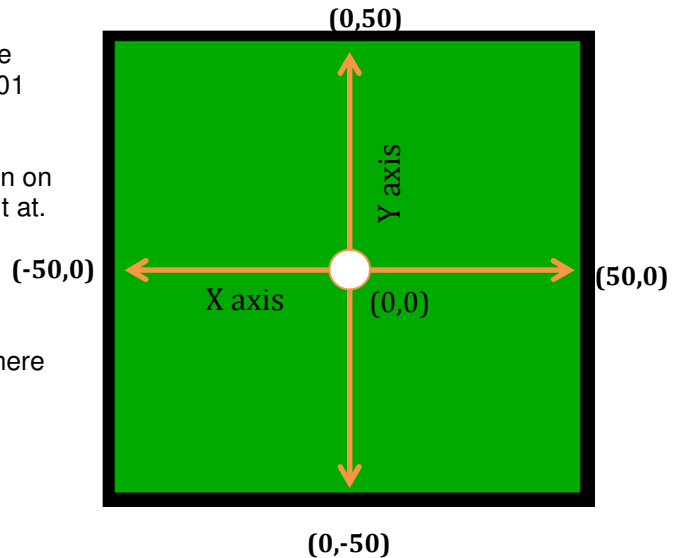
Spaceland is like a grid made up of patches. (0,0) is at the center of the plane. The plane is 101 patches wide and 101 patches tall.

As a programmer you can position an agent at any location on the plane by specifying what X and Y coordinate to place it at.



### Thinking Challenge:

Look at the code to the left. Where on Spaceland will the turtle be created?

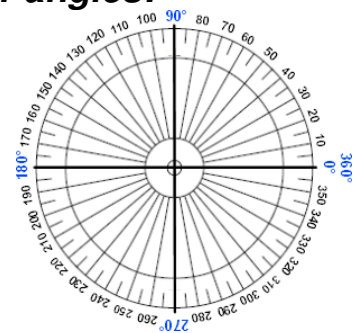


## Getting your turtles to go in a specific direction: *know your angles!*

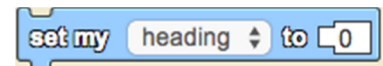
When turtles are created, they are facing out from the center. To have your turtles go in a specific direction you will need to set their heading, which is based on the degrees in a protractor. Notice the start of the circle is at 0 degrees, which also becomes 360 degrees after going around the circle. To go towards the top of Spaceland, the turtle would need to set its heading towards 90 degrees.

### Thinking Challenge:

What will happen when you create 5 turtles that are separated by 2 steps forward (like in your Flower Turtle Project)? What direction does each of them face? How does it change with 10 turtles?



What will happen if you have your turtles set heading to 0 degrees using the blue traits block 'set my...'? In what direction will the turtles head?



### Agents

- create (s)
- create (s) do
- delete
- delete everyone
- delete agent
- scatter
- scatter everyone
- take camera
- me
- my parent

### Traits

- my
- set my to
- of
- color: red green
- color:
- built-in shape:

### Movement

- forward
- backwards
- left by degs
- right by degs
- up
- down
- face towards

### Detection

- on collision with do
- collidee
- count within
- count within with =
- nearest within
- nearest within with =

### Lists

- list of data
- splice into at beginning
- contains data
- Insert data into at beginning
- get data in
- length of

### Variables

- var: name data is
- set to
- value of

### Environment

- clear terrain
- stamp
- stamp grid
- pen down
- terrain color
- clock
- set clock to
- world trait:
- set world to
- The World

### Math

- pi
- random
- + - x /
- remainder of /
- sin
- arcsin
- sqrt
- log
- ln
- power
- larger of and
- round to
- abs

### Interface

- when pushed
- while toggled
- toggle to on for
- hide
- show
- set data box to
- data box
- set label to
- label
- slider value
- Add data to line graph x-axis: y-axis:
- clear line graph
- Update bar graph
- Add entry to table undefined: undefi
- clear table

### Keyboard

- key held?
- key typed?

### Sound

- play sound
- play record delete

### Logic

- if
- if else
- while do
- repeat times
- yield
- =
- not
- !=
- <
- >
- <=
- >=
- and
- or

### Procedures

- procedure: name
- add parameter
- return nothing
- call:
- call:
- parameter
- return-early nothing