

# CS108L Computer Science for All

## Module 6 NetLogo Code Cheat Sheet

Command / Variable	Description
<b>to-report</b> <i>procedure-name</i> □ <b>to-report</b> <i>procedure-name</i> [ <i>input1 input2 ...</i> ]	<p>Used to begin a reporter procedure (<i>procedure-name</i>). Input parameters for the procedure can be specified by using square brackets after procedure-name. The body of the procedure must use report to report a value for the procedure. The procedure must end with the <b>end</b> command.</p> <p><b>Example:</b>  <b>to-report</b> average [a b] ;; a procedure that reports the average of two numbers              <b>report</b> (a + b) / 2  <b>end</b></p> <p><b>to go</b>              <b>let</b> averageAge 0 ;; creates and initializes the variable averageAge              <b>set</b> averageAge average 6 8 ;; calls the <b>average</b> procedure to update the  <b>end</b> ;; variable averageAge with the value 7</p>
<b>report</b> <i>value</i>	<p>Immediately exits from the current to-report procedure and reports <i>value</i> as the result of that procedure. <b>report</b> and <b>to-report</b> are always used in conjunction with each other. See example above.</p>
<b>face</b> <i>agent</i>	<p>Set the caller turtle's heading towards <i>agent</i>. If wrapping is allowed by the topology and the wrapped distance (around the edges of the world) is shorter, face will use the wrapped path. If the caller turtle and the other agent are at the exact same position, the caller's heading won't change.</p> <p><b>Example:</b>  <b>face</b> turtle 1 ;; set the caller turtle's heading to face turtle 1</p>
<b>towards</b> <i>agent</i>	<p>Reports the heading from this agent to the given agent. If wrapping is allowed and the wrapped distance (around the edges of the world) is shorter, towards will use the wrapped path. Note: asking for the heading from an agent to itself, or an agent on the same location, will cause a runtime error.</p> <p><b>EXAMPLE:</b></p>

	<b>set heading towards turtle 1</b> ;; same as <b>face turtle 1</b>
<b>let listname</b> [ value1 value2 ...]	Creates a new local list variable and initializes it to contain the literal values in the list (strings or numbers). If you want to change the value afterwards, use set. <b>Example:</b> <b>let mylist</b> [ 10 2 5 7 ] ;; creates a local list called mylist which contains the ;; values 10, 2, 5, and 7 (in that order).
<b>set listname</b> [ value1 value2 ...]	Sets variable <i>listname</i> to the given list of values. <b>Example:</b> <b>set mylist</b> [ "hello" 2 5 7 ] ;; updates values in the existing variable mylist to ;; "hello", 2, 5, and 7 (in that order).
<b>one-of listname</b> <b>n-of # listname</b>	Randomly selects and reports one item from a list, or # unique items from a list. Note that <b>n-of #</b> will not select the same item twice. <b>Example:</b> <b>show one-of mylist</b> ;; shows either "hello", 2, 5 or 7 in the Command Center <b>show n-of 4 mylist</b> ;; shows ["hello" 2 5 7] in the Command Center
<b>let agentsetname agentset-specification</b>	Creates a new local agentset variable, containing the specified agentset. A local variable is one that exists only within the enclosing block of commands. If you want to change the value afterwards, use set. <b>Example:</b> ;; creates an agentset, redturtles, and initializes it to contain red turtles which ;; are on the same patch as the caller turtle <b>let redturtles turtles-here with [color = red]</b>
<b>set agentsetname agentset-specification</b>	Sets variable <i>agentsetname</i> to the given list of values. <b>Example:</b> <b>set redturtles turtles with [color = red]</b> ;; updates the agentset redturtles to ;; contain all red turtles
<b>ask agentset [commands]</b>	Executes a command block iteratively for each one of the agents in an agentset, after randomizing the order. Only the agents that are in the agentset <i>at the time the ask begins</i> run the commands. <b>Example:</b> <b>ask redturtles</b> [ <b>set shape "airplane"</b>

<b>agentset with [reporter]</b>	Creates and reports a new agentset by filtering an existing agentset according to a specific condition. <b>Example:</b> ;; shows number of red turtles in agentset redturtles in the Command Center <b>show count redturtles with [color = red]</b>
<b>[reporter] of agentset</b>	Reports a list of values (in a random order) created by iteratively evaluating a reporter for each member of an agentset <b>Example:</b> ;; shows list of who numbers of all turtles in redturtles in the Command Center <b>show [who] of redturtles</b> ;; shows x-coordinate values for all turtles in redturtles in the Command Center <b>show [xcor] of redturtles</b>
<b>one-of agentset</b> <b>n-of # agentset</b>	Reporters that return randomly selected agents from the agentset <b>Example:</b> ;; asks 3 random turtles in the agentset redturtles to set their color green <b>ask n-of 3 redturtles [ set color green ]</b> Note: These 3 green turtles are still in the agentset <b>redturtles!</b> To update that: <b>set redturtles turtles with [color = red]</b>