CS151L Fall 2013
Week 10: NetLogo Command Cheat Sheet

Command	Command Description
to-report *procedure-name* **to-report *procedure-name* [*input1* ...]**	Used to begin a reporter procedure (*procedure-name*). 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 **end** command. **Example:** **to-report average [a b]** **report (a + b) / 2** **end** ;; a reporter procedure that reports out the average of two numbers
report *value*	Immediately exits from the current to-report procedure and reports *value* as the result of that procedure. report and to-report are always used in conjunction with each other. See example above.
face *agent*	Set the caller turtle's heading towards *agent*. 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. **Example:** **face turtle 1** ;; set the caller turtle heading facing turtle 1
towards *agent*	Reports the heading from this agent to the given agent. If wrapping is allowed by the topology 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. **EXAMPLE:** **set heading towards turtle 1** ;; same as "face turtle 1

LIST PRIMITIVES	
let *listname* [value1 value2 …]	Creates a new local list variable and initializes it to contain the literal values in the list. A local variable is one that exists only within the enclosing block of commands. If you want to change the value afterwards, use set. **Example:** **let mylist [10 2 5 7]** ;; creates a local list called rlist which contains the values 10, 2, 5, and 7, in that order.
set *listname* [value1 value2 …]	Sets variable *listname,* to the given list of literal values. **Example:** **set mylist [10 2 5 7]** ;; sets the values in the existing variable mylist to 10, 2, 5, 7.
(list value1 value2 ...)	Reports a list of values, which may include literals as well as more complicated reporter expressions. (Parentheses are not required if exactly two values are specified; otherwise, they are required, with opening parenthesis before **list**, and closing parenthesis after the final value to be included in the list.) **Example:** **set rlist (list random 10 random 20 random 30)** ;; sets the values in the existing variable mylist to three random numbers.
fput *item listname* **lput *item listname***	A reporter (results in or reports a value) that creates and reports a new list from an existing list by adding the specified value to the beginning or end of the existing list, *listname*. **Example:** **set mylist fput 20 mylist** ;; if *mylist* initial contains 10,2,5,7; the new *mylist* contains 20,10, 2, 5, 7
but-first *listname* **but-last *listname***	A reporter that creates and reports a new list containing all but the first or the last items of an existing list. **Example:** **set mylist but-last mylist** ;; if *mylist* initially contained 20, 10, 2, 5, 7, the new *mylist* contains 20, 10, 2, 5
one-of *listname* **n-of # *listname***	Randomly selects and reports items from a list. **Example:** **show one-of mylist** ;; if *mylist* contains 20, 10, 2, 5, then one of the numbers in the list will be printed

AGENTSET PRIMITIVES	
let *agentsetname agentset-specification*	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. **Example:** **let redturtles turtles-here with [color = red]** ;; creates an agentset, redturtles, containing the agents on the same patch as the caller turtle that are red.
set *agentsetname agentset-specification*	Sets variable *agentsetname* to the given list of values. **Example:** **set nturtles turtles-on neighbors4** ;;set the agentset nturtles to the set of turtles standing on my neighboring four patches
ask agentset **[…]**	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 *at the time the ask begins* run the commands. **Example:** **Ask nturtles […]** ;; asks the turtles in the agentset nturtles to do the given commands
***agentset* with [*reporter*]**	Creates and reports a new agentset by filtering an existing agentset according to a specific condition **Example:** **show count myturtles with [color = red]** ;; prints the number of red turtles in the agentset myturtles
[*reporter*] of *agentset*	Reports a list of values (in a random order) created by iteratively evaluating a reporter (results in or reports a value) for each member of an agentset **Example:** **show [who] of myturtles** ;; prints out the list of who numbers of all the turtles in the agentset myturtles
one-of *agentset* **n-of # *agentset***	Reporters that return randomly selected agents from the agentset **Example:** **ask n-of 3 myturtles [set color green]** ;; sets 3 turtles in the agents set myturtles to the color green

turtles-here **<*breeds*>-here**	Reports an agentset containing all the turtles on the caller's patch (including the caller itself if it's a turtle). **Example:** **crt 10** **ask turtle 0 [show count turtles-here]** ;; prints 10
turtles-on *agent* **turtles-on *agentset*** **<*breeds*>-on *agent*** **<*breeds*>-on *agentset***	Reports an agentset containing all the turtles that are on the given patch or patches, or standing on the same patch as the given turtle or turtles. If the name of a breed is substituted for "turtles", then only turtles of that breed are included. **Example:** **ask turtles [** **if not any? turtles-on patch-ahead 1** **[fd 1]** **]** ;; lets the turtles move forward one step if there are no turtles on the patch ahead