CS151L Fall 2013
Week 7: NetLogo Command Cheat Sheet

Command	Command Description
breed [plural_name singular_name]	This keyword defines a breed. It is used at the beginning of the Code tab, before any procedure definitions. The first input defines the name of the agentset associated with the breed (the group of all member of that breed)– it is given the plural name of the breed. The second input defines the name of a single member of the breed. Example: **breed [frogs frog]** ;; defines the breed frogs, with the individual member of the breed being called frog
create-<breeds> number **create-<breeds> number [commands]**	Creates *number* new breed_agents at the origin. New breed_agents have random integer headings and the color is randomly selected from the 14 primary colors. If *commands* are supplied, the new breed_agents immediately run them. This is useful for giving the agents a different color, heading, or whatever. (The new agents are created all at once then run one at a time, in random order.) Example: **create-snakes 10** ;; creates 10 snakes **create-frogs 15 [set color green]** ;; creates 15 green frogs
set-default-shape turtles "*string*" **set-default-shape <breeds> "*string*"**	Specifies a default initial shape for all turtles or, or for a particular breed of turtles. When a turtle or a breed is created, or it changes breeds, it shape is set to the given shape. This command doesn't affect existing agents, only agents you create afterwards. The given breed must be already defined breed. The given string must be the name of a currently defined shape, which can be created in the Turtle Shapes Editor or chosen from the Turtles Shapes Editor library. In new models, the default shape for all turtles is "default". Note that specifying a default shape does not prevent you from changing an agent's shape later. Agents don't have to be stuck with their breed's default shape. Example: **set-default-shape cars "car" ;;**sets the default shape of the breed cars to car shape available in the Turtle Shape Editor Library
set shape "ShapeName"	Used in the ask turtles or ask <breeds> command brackets to set the shape of a turtle or a breed. Example: **ask turtles [set shape "wolf"]** ;;sets turtle shape to wolf
ask <breed> # [set *attribute* #] **ask <breeds> [set *attribute* #]**	Tells the given agent or group of agents (agentset) to set a specific attribute (NEtLogo specified agent variable) to a particular value Example: **ask frog 4 [set color red]** ;; sets frog 4's color to red **ask frogs [set size 5]**

ask <*breeds*> [*commands…*] **ask <*breed*> # [*commands…*]** **ask <*breeds*> with [** ***condition***][*commands*]	Tells the given agent or group of agents (agentset) to set a run the given commands Examples: **ask frogs [forward 10]** ;; asks all the frogs to move forward 10 steps **ask frog 2 [right 90]** ;; has frog 2 turn right 90 degrees **ask frogs with [color = red] [hatch 1]** ;; tells all RED frogs to hatch 1 baby
<*breeds*>-own [*var1* ...]	The <*breeds*>-own keyword defines the variables belonging to each that breed and can only be used at the beginning of a program, before any procedures. It you specify turtles-own variables then all breeds of turtles will have that variable. Example: **frogs-own [energy]** ;; defines a variable energy for the breed frogs
hatch # **hatch # [commands]** **hatch-<*breeds*> #** **hatch-<*breeds*> # [commands]**	The **hatch #** is used to create *number* new turtles from EXISITING turtles. Each new turtle inherits of all its variables, including its location, from its parent. (Exceptions: each new turtle will have a new who number (agent ID number), and it may be of a different breed than its parent if the hatch-<*breeds*> form is used.) If the hatch-<*breeds*> form is used, the new turtles are created as members of the given breed. Otherwise, the new turtles are the same breed as their parent The new turtles then run *commands*. You can use the commands to give the new turtles different colors, headings, locations, or whatever. (The new turtles are created all at once, then run one at a time, in random order.) Examples: **hatch 1 [left 45 forward 1]** ;; creates one new turtle and the child turns and moves away **hatch-sheep 1 [set color black]** ;;this turtle creates a new turtle of the sheep breed and sets its color to black
sprout # **sprout # [commands]** **sprout-<*breeds*> #** **sprout-<*breeds*> # [commands]**	Creates *number* new turtles on a patch. The new turtles have random integer headings and the color is randomly selected from the 14 primary colors. The turtles immediately run *commands*. This is useful for giving the new turtles different colors, headings, or whatever. (The new turtles are created all at once then run one at a time, in random order.) If the sprout-<*breeds*> form is used, the new turtles are created as members of the given breed. Examples: **sprout 5** ;; sprouts 5 turtles **sprout-wolves 10** ;; sprouts 10 wolves **sprout 1 [set color red]** ;; sprouts 1 turtles colored red **sprout-sheep 1 [set color black]** ;; sprouts one black sheep