**NetLogo Dictionary**

**NetLogo 6.0.4 User Manual**

Alphabetical: [**A**](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#A)[**B**](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#B)[**C**](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#C)[**D**](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#D)[**E**](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#E)[**F**](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#F)[**G**](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#G)[**H**](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#H)[**I**](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#I)[**J**](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#J)[**L**](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#L)[**M**](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#M)[**N**](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#N)[**O**](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#O)[**P**](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#P)[**R**](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#R)[**S**](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#S)[**T**](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#T)[**U**](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#U)[**V**](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#V)[**W**](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#W)[**X**](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#X)[**Y**](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#Y)[**->**](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#ops)

Categories: [Turtle](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#turtlegroup) - [Patch](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patchgroup) - [Links](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#linkgroup) - [Agentset](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#agentsetgroup) - [Color](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#colorgroup) - [Anonymous Procedures](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#anonproceduresgroup) - [Control/Logic](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#controlgroup) - [World](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#worldgroup)   
[Perspective](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#perspectivegroup) - [Input/Output](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#iogroup) - [File](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#fileiogroup) - [List](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#listsgroup) - [String](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#stringgroup) - [Math](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#mathematicalgroup) - [Plotting](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#plottinggroup) - [System](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#systemgroup) - [HubNet](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnetgroup)

Special: [Variables](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#builtinvariables) - [Keywords](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#Keywords) - [Constants](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#Constants)

**Categories**

This is an approximate grouping. Remember that a turtle-related primitive might still be used by patches or the observer, and vice versa. To see which agents (turtles, patches, links, observer) can actually run a primitive, consult its dictionary entry.

**Turtle-related**

[back](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#back) ([bk](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#back)) [*<breeds>*-at](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#turtles-at) [*<breeds>*-here](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#turtles-here) [*<breeds>*-on](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#turtles-on) [can-move?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#can-move) [clear-turtles](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#clear-turtles) ([ct](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#clear-turtles)) [create-*<breeds>*](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#create-turtles) [create-ordered-*<breeds>*](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#create-ordered-turtles) [create-ordered-turtles](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#create-ordered-turtles) ([cro](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#create-ordered-turtles)) [create-turtles](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#create-turtles) ([crt](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#create-turtles)) [die](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#die) [distance](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#distance) [distancexy](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#distancexy) [downhill](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#downhill) [downhill4](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#downhill) [dx](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#dxy) [dy](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#dxy) [face](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#face)[facexy](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#facexy) [forward](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#forward) ([fd](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#forward)) [hatch](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hatch) [hatch-*<breeds>*](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hatch) [hide-turtle](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hide-turtle) ([ht](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hide-turtle)) [home](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#home) [inspect](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#inspect) [is-*<breed>*?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#is-of-type) [is-turtle?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#is-of-type) [jump](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#jump) [layout-circle](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#layout-circle) [left](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#left) ([lt](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#left)) [move-to](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#move-to) [myself](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#myself) [nobody](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#nobody) [no-turtles](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#no-turtles) [of](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#of) [other](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#other) [patch-ahead](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patch-ahead) [patch-at](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patch-at) [patch-at-heading-and-distance](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patch-at-heading-and-distance) [patch-here](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patch-here)[patch-left-and-ahead](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patch-lr-and-ahead) [patch-right-and-ahead](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patch-lr-and-ahead) [pen-down](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#pen-switch-status) ([pd](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#pen-switch-status)) [pen-erase](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#pen-switch-status) ([pe](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#pen-switch-status)) [pen-up](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#pen-switch-status) ([pu](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#pen-switch-status)) [random-xcor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#random-cor) [random-ycor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#random-cor) [right](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#right) ([rt](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#right)) [self](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#self) [set-default-shape](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#set-default-shape) [\_\_set-line-thickness](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#set-line-thickness) [setxy](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#setxy) [shapes](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#shapes) [show-turtle](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#show-turtle) ([st](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#show-turtle)) [sprout](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#sprout) [sprout-*<breeds>*](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#sprout)[stamp](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#stamp) [stamp-erase](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#stamp-erase) [stop-inspecting](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#stop-inspecting) [subject](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#subject) [subtract-headings](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#subtract-headings) [tie](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#tie) [towards](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#towards) [towardsxy](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#towardsxy) [turtle](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#turtle) [turtle-set](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#turtle-set) [turtles](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#turtles) [turtles-at](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#turtles-at) [turtles-here](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#turtles-here) [turtles-on](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#turtles-on) [turtles-own](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#turtles-own) [untie](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#untie) [uphill](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#uphill) [uphill4](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#uphill)

**Patch-related**

[clear-patches](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#clear-patches) ([cp](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#clear-patches)) [diffuse](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#diffuse) [diffuse4](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#diffuse4) [distance](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#distance) [distancexy](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#distancexy) [import-pcolors](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#import-pcolors) [import-pcolors-rgb](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#import-pcolors-rgb) [inspect](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#inspect) [is-patch?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#is-of-type) [myself](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#myself) [neighbors](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#neighbors) [neighbors4](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#neighbors) [nobody](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#nobody) [no-patches](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#no-patches) [of](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#of) [other](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#other) [patch](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patch) [patch-at](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patch-at) [patch-ahead](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patch-ahead) [patch-at-heading-and-distance](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patch-at-heading-and-distance)[patch-here](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patch-here) [patch-left-and-ahead](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patch-lr-and-ahead) [patch-right-and-ahead](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patch-lr-and-ahead) [patch-set](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patch-set) [patches](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patches) [patches-own](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patches-own) [random-pxcor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#random-pcor) [random-pycor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#random-pcor) [self](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#self) [sprout](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#sprout) [sprout-*<breeds>*](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#sprout) [stop-inspecting](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#stop-inspecting) [subject](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#subject) [turtles-here](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#turtles-here)

**Link-related**

[both-ends](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#both-ends) [clear-links](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#clear-links) [create-<breed>-from](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#create-link) [create-<breeds>-from](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#create-link) [create-<breed>-to](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#create-link) [create-<breeds>-to](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#create-link) [create-<breed>-with](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#create-link) [create-<breeds>-with](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#create-link) [create-link-from](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#create-link) [create-links-from](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#create-link) [create-link-to](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#create-link) [create-links-to](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#create-link) [create-link-withcreate-links-with](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#create-link) [die](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#die) [hide-link](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hide-link) [in-<breed>-neighbor?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#in-link-neighbor) [in-<breed>-neighbors](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#in-link-neighbors) [in-<breed>-from](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#in-link-from) [in-link-neighbor?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#in-link-neighbor) [in-link-neighbors](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#in-link-neighbors) [in-link-from](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#in-link-from) [is-directed-link?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#is-of-type) [is-link?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#is-of-type) [is-link-set?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#is-of-type) [is-*<link-breed>*?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#is-of-type) [is-undirected-link?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#is-of-type) [layout-radial](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#layout-radial)[layout-spring](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#layout-spring) [layout-tutte](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#layout-tutte) [<breed>-neighbor?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#link-neighbor) [<breed>-neighbors](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#link-neighbors) [<breed>-with](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#link-with) [link-heading](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#link-heading) [link-length](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#link-length) [link-neighbor?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#link-neighbor) [link](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#link) [links](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#links) [links-own](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#links-own) [<link-breeds>-own](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#links-own) [link-neighbors](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#link-neighbors) [link-with](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#link-with) [my-<breeds>](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#my-links) [my-in-<breeds>](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#my-in-links) [my-in-links](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#my-in-links)[my-links](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#my-links) [my-out-<breeds>](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#my-out-links) [my-out-links](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#my-out-links) [no-links](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#no-links) [other-end](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#other-end) [out-<breed>-neighbor?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#out-link-neighbor) [out-<breed>-neighbors](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#out-link-neighbors) [out-<breed>-to](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#out-link-to) [out-link-neighbor?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#out-link-neighbor) [out-link-neighbors](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#out-link-neighbors) [out-link-to](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#out-link-to) [show-link](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#show-link) [tie](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#tie) [untie](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#untie)

**Agentset**

[all?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#all) [any?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#any) [ask](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#ask) [ask-concurrent](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#ask-concurrent) [at-points](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#at-points) [*<breeds>*-at](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#turtles-at) [*<breeds>*-here](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#turtles-here) [*<breeds>*-on](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#turtles-on) [count](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#count) [in-cone](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#in-cone) [in-radius](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#in-radius) [is-agent?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#is-of-type) [is-agentset?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#is-of-type) [is-patch-set?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#is-of-type) [is-turtle-set?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#is-of-type) [link-set](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#link-set) [max-n-of](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#max-n-of) [max-one-of](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#max-one-of) [member?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#member) [min-n-of](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#min-n-of) [min-one-of](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#min-one-of) [n-of](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#n-of)[neighbors](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#neighbors) [neighbors4](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#neighbors) [no-links](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#no-links) [no-patches](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#no-patches) [no-turtles](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#no-turtles) [of](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#of) [one-of](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#one-of) [other](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#other) [patch-set](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patch-set) [patches](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patches) [sort](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#sort) [sort-by](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#sort-by) [sort-on](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#sort-on) [turtle-set](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#turtle-set) [turtles](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#turtles) [turtles-at](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#turtles-at) [turtles-here](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#turtles-here) [turtles-on](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#turtles-on) [with](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#with) [with-max](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#with-max) [with-min](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#with-min)

**Color**

[approximate-hsb](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#approximate-hsb) [approximate-rgb](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#approximate-rgb) [base-colors](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#base-colors) [color](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#color) [extract-hsb](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#extract-hsb) [extract-rgb](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#extract-rgb) [hsb](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hsb) [import-pcolors](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#import-pcolors) [import-pcolors-rgb](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#import-pcolors-rgb) [pcolor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#pcolor) [rgb](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#rgb) [scale-color](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#scale-color) [shade-of?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#shade-of) [wrap-color](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#wrap-color)

**Control flow and logic**

[and](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#and) [ask](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#ask) [ask-concurrent](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#ask-concurrent) [carefully](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#carefully) [end](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#end) [error](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#error) [error-message](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#error-message) [every](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#every) [if](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#if) [ifelse](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#ifelse) [ifelse-value](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#ifelse-value) [let](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#let) [loop](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#loop) [not](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#not) [or](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#or) [repeat](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#repeat) [report](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#report) [run](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#run) [runresult](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#run) [; (semicolon)](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#semicolon) [set](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#set) [stop](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#stop) [startup](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#startup) [to](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#to) [to-report](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#to-report) [wait](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#wait) [while](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#while) [with-local-randomness](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#with-local-randomness) [without-interruption](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#without-interruption) [xor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#xor)

**Anonymous Procedures**

[-> (anonymous procedure)](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#arrow) [filter](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#filter) [foreach](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#foreach) [is-anonymous-command?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#is-of-type) [is-anonymous-reporter?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#is-of-type) [map](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#map) [n-values](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#n-values) [reduce](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#reduce) [run](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#run) [runresult](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#run) [sort-by](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#sort-by)

**World**

[clear-all](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#clear-all) ([ca](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#clear-all)) [clear-drawing](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#clear-drawing) ([cd](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#clear-drawing)) [clear-globals](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#clear-globals) [clear-patches](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#clear-patches) ([cp](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#clear-patches)) [clear-ticks](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#clear-ticks) [clear-turtles](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#clear-turtles) ([ct](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#clear-turtles)) [display](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#display) [import-drawing](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#import-drawing) [import-pcolors](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#import-pcolors) [import-pcolors-rgb](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#import-pcolors-rgb) [no-display](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#no-display) [max-pxcor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#max-pcor) [max-pycor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#max-pcor) [min-pxcor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#min-pcor) [min-pycor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#min-pcor) [patch-size](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patch-size) [reset-ticks](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#reset-ticks)[resize-world](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#resize-world) [set-patch-size](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#set-patch-size) [stop-inspecting-dead-agents](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#stop-inspecting-dead-agents) [tick](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#tick) [tick-advance](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#tick-advance) [ticks](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#ticks) [world-width](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#world-dim) [world-height](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#world-dim)

**Perspective**

[follow](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#follow) [follow-me](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#follow-me) [reset-perspective](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#reset-perspective) ([rp](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#reset-perspective)) [ride](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#ride) [ride-me](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#ride-me) [subject](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#subject) [watch](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#watch) [watch-me](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#watch-me)

**HubNet**

[hubnet-broadcast](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-broadcast) [hubnet-broadcast-clear-output](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-broadcast-clear-output) [hubnet-broadcast-message](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-broadcast-message) [hubnet-clear-override](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-clear-override) [hubnet-clear-overrides](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-clear-override) [hubnet-clients-list](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-clients-list) [hubnet-enter-message?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-enter-message) [hubnet-exit-message?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-exit-message) [hubnet-kick-all-clients](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-kick-all-clients) [hubnet-kick-client](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-kick-client) [hubnet-fetch-message](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-fetch-message) [hubnet-message](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-message) [hubnet-message-source](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-message-source) [hubnet-message-tag](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-message-tag) [hubnet-message-waiting?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-message-waiting) [hubnet-reset](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-reset) [hubnet-reset-perspective](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-reset-perspective) [hubnet-send](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-send) [hubnet-send-clear-output](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-send-clear-output) [hubnet-send-follow](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-send-follow)[hubnet-send-message](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-send-message) [hubnet-send-override](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-send-override) [hubnet-send-watch](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-send-watch)

**Input/output**

[beep](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#beep) [clear-output](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#clear-output) [date-and-time](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#date-and-time) [export-view](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#export-cmds) [export-interface](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#export-cmds) [export-output](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#export-cmds) [export-plot](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#export-cmds) [export-all-plots](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#export-cmds) [export-world](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#export-cmds) [import-drawing](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#import-drawing) [import-pcolors](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#import-pcolors) [import-pcolors-rgb](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#import-pcolors-rgb) [import-world](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#import-world) [mouse-down?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#mouse-down) [mouse-inside?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#mouse-inside) [mouse-xcormouse-ycor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#mouse-cor) [output-print](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#output-cmds) [output-show](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#output-cmds) [output-type](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#output-cmds) [output-write](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#output-cmds) [print](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#print) [read-from-string](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#read-from-string) [reset-timer](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#reset-timer) [set-current-directory](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#set-current-directory) [show](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#show) [timer](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#timer) [type](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#type) [user-directory](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#user-directory) [user-file](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#user-file) [user-new-file](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#user-new-file) [user-input](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#user-input) [user-message](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#user-message) [user-one-of](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#user-one-of) [user-yes-or-no?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#user-yes-or-no)[write](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#write)

**File**

[file-at-end?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-at-end) [file-close](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-close) [file-close-all](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-close-all) [file-delete](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-delete) [file-exists?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-exists) [file-flush](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-flush) [file-open](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-open) [file-print](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-print) [file-read](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-read) [file-read-characters](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-read-characters) [file-read-line](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-read-line) [file-show](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-show) [file-type](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-type) [file-write](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-write) [user-directory](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#user-directory) [user-file](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#user-file) [user-new-file](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#user-new-file)

**List**

[but-first](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#but-first-and-last) [but-last](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#but-first-and-last) [empty?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#empty) [filter](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#filter) [first](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#first) [foreach](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#foreach) [fput](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#fput) [histogram](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#histogram) [insert-item](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#insert-item) [is-list?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#is-of-type) [item](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#item) [last](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#last) [length](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#length) [list](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#list) [lput](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#lput) [map](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#map) [max](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#max) [member?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#member) [min](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#min) [modes](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#modes) [n-of](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#n-of) [n-values](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#n-values) [of](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#of) [position](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#position) [one-of](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#one-of) [range](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#range) [reduce](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#reduce) [remove](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#remove) [remove-duplicates](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#remove-duplicates) [remove-item](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#remove-item) [replace-item](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#replace-item) [reverse](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#reverse) [sentence](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#sentence) [shuffle](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#shuffle) [sort](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#sort) [sort-by](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#sort-by) [sort-on](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#sort-on) [sublist](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#subliststring)

**String**

[Operators (<, >, =, !=, <=, >=)](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#Symbols) [but-first](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#but-first-and-last) [but-last](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#but-first-and-last) [empty?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#empty) [first](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#first) [insert-item](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#insert-item) [is-string?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#is-of-type) [item](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#item) [last](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#last) [length](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#length) [member?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#member) [position](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#position) [remove](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#remove) [remove-item](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#remove-item) [read-from-string](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#read-from-string) [replace-item](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#replace-item) [reverse](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#reverse) [substring](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#subliststring) [word](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#word)

**Mathematical**

[Arithmetic Operators (+, \*, -, /, ^, <, >, =, !=, <=, >=)](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#Symbols) [abs](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#abs) [acos](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#acos) [asin](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#asin) [atan](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#atan) [ceiling](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#ceiling) [cos](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#cos) [e](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#num-e) [exp](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#exp) [floor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#floor) [int](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#int) [is-number?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#is-of-type) [ln](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#ln) [log](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#log) [max](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#max) [mean](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#mean) [median](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#median) [min](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#min) [mod](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#mod) [modes](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#modes) [new-seed](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#new-seed) [pi](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#pi) [precision](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#precision) [random](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#random) [random-exponential](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#random-reporters) [random-float](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#random-float) [random-gamma](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#random-reporters) [random-normal](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#random-reporters) [random-poisson](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#random-reporters) [random-seed](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#random-seed) [remainder](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#remainder) [round](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#round) [sin](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#sin) [sqrt](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#sqrt) [standard-deviation](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#standard-deviation) [subtract-headings](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#subtract-headings) [sum](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#sum) [tan](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#tan) [variance](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#variance)

**Plotting**

[autoplot?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#autoplot) [auto-plot-off](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#auto-plot-status) [auto-plot-on](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#auto-plot-status) [clear-all-plots](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#clear-all-plots) [clear-plot](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#clear-plot) [create-temporary-plot-pen](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#create-temporary-plot-pen) [export-plot](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#export-cmds) [export-all-plots](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#export-cmds) [histogram](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#histogram) [plot](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#plot) [plot-name](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#plot-name) [plot-pen-exists?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#plot-pen-exists) [plot-pen-down](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#plot-pen-switch-status) [plot-pen-reset](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#plot-pen-reset) [plot-pen-up](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#plot-pen-switch-status) [plot-x-max](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#plot-cor-max-or-min) [plot-x-min](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#plot-cor-max-or-min) [plot-y-max](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#plot-cor-max-or-min) [plot-y-min](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#plot-cor-max-or-min) [plotxy](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#plotxy) [set-current-plot](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#set-current-plot) [set-current-plot-pen](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#set-current-plot-pen) [set-histogram-num-bars](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#set-histogram-num-bars) [set-plot-background-color](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#set-plot-background-color) [set-plot-pen-color](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#set-plot-pen-color) [set-plot-pen-interval](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#set-plot-pen-interval) [set-plot-pen-mode](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#set-plot-pen-mode) [set-plot-x-range](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#set-plot--range) [set-plot-y-range](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#set-plot--range) [setup-plots](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#setup-plots) [update-plots](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#update-plots)

**BehaviorSpace**

[behaviorspace-experiment-name](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#behaviorspace-experiment-name) [behaviorspace-run-number](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#behaviorspace-run-number)

**System**

[netlogo-version](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#netlogo-version) [netlogo-web?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#netlogo-web)

**Built-In Variables**

**Turtles**

[breed](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#breedvar) [color](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#color) [heading](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#heading) [hidden?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hidden) [label](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#label) [label-color](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#label-color) [pen-mode](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#pen-mode) [pen-size](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#pen-size) [shape](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#shape) [size](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#size) [who](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#who) [xcor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#xcor) [ycor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#ycor)

**Patches**

[pcolor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#pcolor) [plabel](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#plabel) [plabel-color](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#plabel-color) [pxcor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#pcor) [pycor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#pcor)

**Links**

[breed](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#breed) [color](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#color) [end1](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#end1) [end2](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#end2) [hidden?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hidden) [label](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#label) [label-color](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#label-color) [shape](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#shape) [thickness](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#thickness) [tie-mode](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#tie-mode)

**Other**

[->](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#arrow)

**Keywords**

[breed](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#breed) [directed-link-breed](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#directed-link-breed) [end](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#end) [extensions](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#extensions) [globals](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#globals) [\_\_includes](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#includes) [links-own](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#links-own) [patches-own](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patches-own) [to](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#to) [to-report](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#to-report) [turtles-own](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#turtles-own) [undirected-link-breed](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#undirected-link-breed)

**Constants**

**Mathematical Constants**

**e** = 2.718281828459045   
**pi** = 3.141592653589793

**Boolean Constants**

**false**   
**true**

**Color Constants**

**black** = 0   
**gray** = 5   
**white** = 9.9   
**red** = 15   
**orange** = 25   
**brown** = 35   
**yellow** = 45   
**green** = 55   
**lime** = 65   
**turquoise** = 75   
**cyan** = 85   
**sky** = 95   
**blue** = 105   
**violet** = 115   
**magenta** = 125   
**pink** = 135

See the [Colors](http://ccl.northwestern.edu/netlogo/docs/programming.html#colors) section of the Programming Guide for more details.

**A**

**abs***1.0*

**abs *number***

Reports the absolute value of *number*.

show abs -7

=> 7

show abs 5

=> 5

**acos***1.3*

**acos *number***

Reports the arc cosine (inverse cosine) of the given number. The input must be in the range -1 to 1. The result is in degrees, and lies in the range 0 to 180.

**all?***4.0*

**all? *agentset* [*reporter*]**

Reports true if all of the agents in the agentset report true for the given reporter. Otherwise reports false as soon as a counterexample is found.

If the agentset is empty, reports true.

The reporter must report a boolean value for every agent (either true or false), otherwise an error occurs.

if all? turtles [color = red]

[ show "every turtle is red!" ]

See also [any?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#any).

**and***1.0*

***condition1* and *condition2***

Reports true if both *condition1* and *condition2* are true.

Note that if *condition1* is false, then *condition2* will not be run (since it can't affect the result).

if (pxcor > 0) and (pycor > 0)

[ set pcolor blue ] ;; the upper-right quadrant of

;; patches turn blue

**any?***2.0*

**any? *agentset***

Reports true if the given agentset is non-empty, false otherwise.

Equivalent to "count *agentset* > 0", but more efficient (and arguably more readable).

if any? turtles with [color = red]

[ show "at least one turtle is red!" ]

Note: nobody is not an agentset. You only get nobody back in situations where you were expecting a single agent, not a whole agentset. If any? gets nobody as input, an error results.

See also [all?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#all), [nobody](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#nobody).

**approximate-hsb***4.0*

**approximate-hsb *hue saturation brightness***

Reports a number in the range 0 to 140, not including 140 itself, that represents the given color, specified in the HSB spectrum, in NetLogo's color space.

The first value (hue) should be in the range of 0 to 360, the second and third (saturation and brightness) in the range between 0 and 100.

The color reported may be only an approximation, since the NetLogo color space does not include all possible colors.

show approximate-hsb 0 0 0

=> 0 ;; (black)

show approximate-hsb 180 57.143 76.863

=> 85 ;; (cyan)

See also [extract-hsb](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#extract-hsb), [approximate-rgb](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#approximate-rgb), [extract-rgb](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#extract-rgb).

**approximate-rgb***4.0*

**approximate-rgb *red green blue***

Reports a number in the range 0 to 140, not including 140 itself, that represents the given color, specified in the RGB spectrum, in NetLogo's color space.

All three inputs should be in the range 0 to 255.

The color reported may be only an approximation, since the NetLogo color space does not include all possible colors. (See [approximate-hsb](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#approximate-hsb) for a description of what parts of the HSB color space NetLogo colors cover; this is difficult to characterize in RGB terms.)

show approximate-rgb 0 0 0

=> 0 ;; black

show approximate-rgb 0 255 255

=> 85.2 ;; cyan

See also [extract-rgb](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#extract-rgb), [approximate-hsb](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#approximate-hsb), and [extract-hsb](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#extract-hsb).

**Arithmetic Operators+***1.0***\****1.0***-***1.0***/***1.0***^***1.0***<***1.0***>***1.0***=***1.0***!=***1.0***<=***1.0***>=***1.0*

All of these operators take two inputs, and all act as "infix operators" (going between the two inputs, as in standard mathematical use). NetLogo correctly supports order of operations for infix operators.

The operators work as follows: + is addition, \* is multiplication, - is subtraction, / is division, ^ is exponentiation, < is less than, > is greater than, = is equal to, != is not equal to, <= is less than or equal, >= is greater than or equal.

Note that the subtraction operator (-) always takes two inputs unless you put parentheses around it, in which case it can take one input. For example, to take the negative of x, write (- x), with the parentheses.

All of the comparison operators also work on strings.

All of the comparison operators work on agents. Turtles are compared by who number. Patches are compared top to bottom left to right, so patch 0 10 is less than patch 0 9 and patch 9 0 is less than patch 10 0. Links are ordered by end points and in case of a tie by breed. So link 0 9 is before link 1 10 as the end1 is smaller, and link 0 8 is less than link 0 9. If there are multiple breeds of links unbreeded links will come before breeded links of the same end points and breeded links will be sorted in the order they are declared in the Code tab.

Agentsets can be tested for equality or inequality. Two agentsets are equal if they are the same type (turtle or patch) and contain the same agents.

If you are not sure how NetLogo will interpret your code, you should add parentheses.

show 5 \* 6 + 6 / 3

=> 32

show 5 \* (6 + 6) / 3

=> 20

Many extension objects may be tested for equality and inequality using = and !=. For instance, the array, matrix, and table objects returned by their respective extensions may be compared for equality / inequality. Extension objects may not be tested using <, >, <=, or >=.

**asin***1.3*

**asin *number***

Reports the arc sine (inverse sine) of the given number. The input must be in the range -1 to 1. The result is in degrees, and lies in the range -90 to 90.

**ask***1.0*

**ask *agentset* [*commands*]ask *agent* [*commands*]**

The specified agent or agentset runs the given commands.

ask turtles [ fd 1 ]

;; all turtles move forward one step

ask patches [ set pcolor red ]

;; all patches turn red

ask turtle 4 [ rt 90 ]

;; only the turtle with id 4 turns right

Note: only the observer can ask all turtles or all patches. This prevents you from inadvertently having all turtles ask all turtles or all patches ask all patches, which is a common mistake to make if you're not careful about which agents will run the code you are writing.

Note: Only the agents that are in the agentset *at the time the ask begins* run the commands.

**ask-concurrent***4.0*

**ask-concurrent *agentset* [*commands*]**

This primitive exists only for backwards compatibility. We don't recommend using it new models.

The agents in the given agentset run the given commands, using a turn-taking mechanism to produce simulated concurrency. See the [Ask-Concurrent](http://ccl.northwestern.edu/netlogo/docs/programming.html#ask-concurrent) section of the Programming Guide for details on how this works.

Note: Only the agents that are in the agentset *at the time the ask begins* run the commands.

See also [without-interruption](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#without-interruption).

**at-points***1.0*

***agentset* at-points [[*x1 y1*] [*x2 y2*] ...]**

Reports a subset of the given agentset that includes only the agents on the patches at the given coordinates (relative to this agent). The coordinates are specified as a list of two-item lists, where the two items are the x and y offsets.

If the caller is the observer, then the points are measured relative to the origin, in other words, the points are taken as absolute patch coordinates.

If the caller is a turtle, the points are measured relative to the turtle's exact location, and not from the center of the patch under the turtle.

ask turtles at-points [[2 4] [1 2] [10 15]]

[ fd 1 ] ;; only the turtles on the patches at the

;; coordinates (2,4), (1,2) and (10,15),

;; relative to the caller, move

**atan***1.0*

**atan *x y***

Converts x and y offsets to a turtle heading in degrees (from 0 to 360).

Note that this version of atan is designed to conform to the geometry of the NetLogo world, where a heading of 0 is straight up, 90 is to the right, and so on clockwise around the circle. (Normally in geometry an angle of 0 is right, 90 is up, and so on, counterclockwise around the circle, and atan would be defined accordingly.)

When y is 0: if x is positive, it reports 90; if x is negative, it reports 270; if x is zero, you get an error.

show atan 1 -1

=> 135

show atan -1 1

=> 315

crt 1 [ set heading 30 fd 1 print atan xcor ycor ]

=> 30

In the final example, note that the result of atan equals the turtle's heading.

If you ever need to convert a turtle heading (obtained with atan or otherwise) to a normal mathematical angle, the following should be helpful:

to-report heading-to-angle [ h ]

report (90 - h) mod 360

end

**autoplot?***1.0*

**autoplot?**

Reports true if auto-plotting is on for the current plot, false otherwise.

**auto-plot-off***1.0***auto-plot-on***1.0*

**auto-plot-offauto-plot-on**

This pair of commands is used to control the NetLogo feature of auto-plotting in the current plot. Auto-plotting will automatically update the x and y axes of the plot whenever the current pen exceeds these boundaries. It is useful when wanting to show all plotted values in the current plot, regardless of the current plot ranges.

**B**

**back***1.0***bk***1.0*

**back *number*Turtle Command**

The turtle moves backward by *number* steps. (If *number* is negative, the turtle moves forward.)

Turtles using this primitive can move a maximum of one unit per time increment. So bk 0.5 and bk 1 both take one unit of time, but bk 3 takes three.

If the turtle cannot move backward *number* steps because it is not permitted by the current topology the turtle will complete as many steps of 1 as it can and stop.

See also [forward](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#forward), [jump](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#jump), [can-move?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#can-move).

**base-colors***4.0*

**base-colors**

Reports a list of the 14 basic NetLogo hues.

print base-colors

=> [5 15 25 35 45 55 65 75 85 95 105 115 125 135]

ask turtles [ set color one-of base-colors ]

;; each turtle turns a random base color

ask turtles [ set color one-of remove gray base-colors ]

;; each turtle turns a random base color except for gray

**beep***2.1*

**beep**

Emits a beep. Note that the beep sounds immediately, so several beep commands in close succession may produce only one audible sound.

Example:

beep ;; emits one beep

repeat 3 [ beep ] ;; emits 3 beeps at once,

;; so you only hear one sound

repeat 3 [ beep wait 0.1 ] ;; produces 3 beeps in succession,

;; separated by 1/10th of a second

When running headless, this command has no effect.

**behaviorspace-experiment-name***5.2*

**behaviorspace-experiment-name**

Reports the current experiment name in the current experiment.

If no BehaviorSpace experiment is running, reports "".

**behaviorspace-run-number***4.1.1*

**behaviorspace-run-number**

Reports the current run number in the current BehaviorSpace experiment, starting at 1.

If no BehaviorSpace experiment is running, reports 0.

**both-ends***4.0*

**both-endsLink Command**

Reports the agentset of the 2 nodes connected by this link.

crt 2

ask turtle 0 [ create-link-with turtle 1 ]

ask link 0 1 [

ask both-ends [ set color red ] ;; turtles 0 and 1 both turn red

]

**breed**

**breedTurtle Command Link Command**

This is a built-in turtle and link variable. It holds the agentset of all turtles or links of the same breed as this turtle or link. (For turtles or links that do not have any particular breed, this is the [turtles](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#turtles) agentset of all turtles or the [links](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#links) agentset of all links respectively.)

You can set this variable to change a turtle or link's breed. (When a turtle changes breeds, its shape is reset to the default shape for that breed. See [set-default-shape](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#set-default-shape).)

See also [breed](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#breed), [directed-link-breed](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#directed-link-breed), [undirected-link-breed](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#undirected-link-breed)

Example:

breed [cats cat]

breed [dogs dog]

;; turtle code:

if breed = cats [ show "meow!" ]

set breed dogs

show "woof!"

directed-link-breed [ roads road ]

;; link code

if breed = roads [ set color gray ]

**breed**

**breed [*<breeds>* *<breed>*]**

This keyword, like the globals, turtles-own, and patches-own keywords, can only be used at the beginning of the Code tab, before any procedure definitions. It defines a breed. The first input defines the name of the agentset associated with the breed. The second input defines the name of a single member of the breed.

Any turtle of the given breed:

* is part of the agentset named by the breed name
* has its breed built-in variable set to that agentset

Most often, the agentset is used in conjunction with ask to give commands to only the turtles of a particular breed.

breed [mice mouse]

breed [frogs frog]

to setup

clear-all

create-mice 50

ask mice [ set color white ]

create-frogs 50

ask frogs [ set color green ]

show [breed] of one-of mice ;; prints mice

show [breed] of one-of frogs ;; prints frogs

end

show mouse 1

;; prints (mouse 1)

show frog 51

;; prints (frog 51)

show turtle 51

;; prints (frog 51)

See also [globals](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#globals), [patches-own](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patches-own), [turtles-own](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#turtles-own), [*<breeds>*-own](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#turtles-own), [create-*<breeds>*](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#create-turtles), [*<breeds>*-at](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#turtles-at), [*<breeds>*-here](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#turtles-here).

**but-first***1.0***butfirst***1.0***bf***1.0***but-last***1.0***butlast***1.0***bl***1.0*

**but-first *list*but-first *string*but-last *list*but-last *string***

When used on a list, but-first reports all of the list items of *list* except the first, and but-last reports all of the list items of *list* except the last.

On strings, but-first and but-last report a shorter string omitting the first or last character of the original string.

;; mylist is [2 4 6 5 8 12]

set mylist but-first mylist

;; mylist is now [4 6 5 8 12]

set mylist but-last mylist

;; mylist is now [4 6 5 8]

show but-first "string"

;; prints "tring"

show but-last "string"

;; prints "strin"

**C**

**can-move?***3.1*

**can-move? *distance*Turtle Command**

Reports true if this turtle can move *distance* in the direction it is facing without violating the topology; reports false otherwise.

It is equivalent to:

patch-ahead *distance* != nobody

**carefully***2.1*

**carefully [ *commands1* ] [ *commands2* ]**

Runs *commands1*. If a runtime error occurs inside *commands1*, NetLogo won't stop and alert the user that an error occurred. It will suppress the error and run *commands2* instead.

The error-message reporter can be used in *commands2* to find out what error was suppressed in *commands1*. See [error-message](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#error-message).

carefully [ print one-of [1 2 3] ] [ print error-message ]

=> 3

observer> carefully [ print one-of [] ] [ print error-message ]

=> ONE-OF got an empty list as input.

**ceiling***1.0*

**ceiling *number***

Reports the smallest integer greater than or equal to *number*.

show ceiling 4.5

=> 5

show ceiling -4.5

=> -4

See also [floor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#floor), [round](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#round), [precision](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#precision).

**clear-all***1.0***ca***1.0*

**clear-allObserver Command**

Combines the effects of clear-globals, clear-ticks, clear-turtles, clear-patches, clear-drawing, clear-all-plots, and clear-output.

**clear-all-plots***1.0*

**clear-all-plotsObserver Command**

Clears every plot in the model. See [clear-plot](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#clear-plot) for more information.

**clear-drawing***3.0***cd***3.0*

**clear-drawingObserver Command**

Clears all lines and stamps drawn by turtles.

**clear-globals***5.2*

**clear-globalsObserver Command**

Sets all global variables to 0.

**clear-links***4.0*

**clear-linksObserver Command**

Kills all links.

See also [die](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#die).

**clear-output***1.0*

**clear-outputObserver Command**

Clears all text from the model's output area, if it has one. Otherwise does nothing.

**clear-patches***1.0***cp***1.0*

**clear-patchesObserver Command**

Clears the patches by resetting all patch variables to their default initial values, including setting their color to black.

**clear-plot**

**clear-plot**

In the current plot only, resets all plot pens, deletes all temporary plot pens, resets the plot to its default values (for x range, y range, etc.), and resets all permanent plot pens to their default values. The default values for the plot and for the permanent plot pens are set in the plot Edit dialog, which is displayed when you edit the plot. If there are no plot pens after deleting all temporary pens, that is to say if there are no permanent plot pens, a default plot pen will be created with the following initial settings:

* Pen: down
* Color: black
* Mode: 0 (line mode)
* Name: "default"
* Interval: 1

See also [clear-all-plots](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#clear-all-plots).

**clear-ticks***5.0*

**clear-ticksObserver Command**

Clears the tick counter.

Does not set the counter to zero. After this command runs, the tick counter has no value. Attempting to access or update it is an error until [reset-ticks](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#reset-ticks) is called.

See also [reset-ticks](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#reset-ticks).

**clear-turtles***1.0***ct***1.0*

**clear-turtlesObserver Command**

Kills all turtles.

Also resets the who numbering, so the next turtle created will be turtle 0.

See also [die](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#die).

**color**

**colorTurtle Command Link Command**

This is a built-in turtle or link variable. It holds the color of the turtle or link. You can set this variable to make the turtle or link change color. Color can be represented either as a NetLogo color (a single number), or an RGB color (a list of 3 numbers). See details in the [Colors section](http://ccl.northwestern.edu/netlogo/docs/programming.html#colors) of the Programming Guide.

See also [pcolor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#pcolor).

**cos***1.0*

**cos *number***

Reports the cosine of the given angle. Assumes the angle is given in degrees.

show cos 180

=> -1

**count***1.0*

**count *agentset***

Reports the number of agents in the given agentset.

show count turtles

;; prints the total number of turtles

show count patches with [pcolor = red]

;; prints the total number of red patches

**create-ordered-turtles***4.0***cro***4.0*

**create-ordered-turtles *number*create-ordered-turtles *number* [ *commands* ]create-ordered*<breeds>* *number*create-ordered*<breeds>* *number* [ *commands* ]Observer Command**

Creates *number* new turtles. New turtles start at position (0, 0), are created with the 14 primary colors, and have headings from 0 to 360, evenly spaced.

If the create-ordered-*<breeds>* form is used, the new turtles are created as members of the given breed.

If *commands* are supplied, the new turtles immediately run them. This is useful for giving the new turtles a different color, heading, or whatever. (The new turtles are created all at once then run one at a time, in random order.)

cro 100 [ fd 10 ] ;; makes an evenly spaced circle

**create-<breed>-tocreate-<breeds>-tocreate-<breed>-fromcreate-<breeds>-fromcreate-<breed>-withcreate-<breeds>-withcreate-link-to***4.0***create-links-to***4.0***create-link-from***4.0***create-links-from***4.0***create-link-with***4.0***create-links-with***4.0*

**create-<breed>-to *turtle*create-<breed>-to *turtle* [ *commands* ]create-<breed>-from *turtle*create-<breed>-from *turtle* [ *commands* ]create-<breed>-with *turtle*create-<breed>-with *turtle* [ *commands* ]create-<breeds>-to *turtleset*create-<breeds>-to *turtleset* [ *commands* ]create-<breeds>-from *turtleset*create-<breeds>-from *turtleset* [ *commands* ]create-<breeds>-with *turtleset*create-<breeds>-with *turtleset* [ *commands* ]create-link-to *turtle*create-link-to *turtle* [ *commands* ]create-link-from *turtle*create-link-from *turtle* [ *commands* ]create-link-with *turtle*create-link-with *turtle* [ *commands* ]create-links-to *turtleset*create-links-to *turtleset* [ *commands* ]create-links-from *turtleset*create-links-from *turtleset* [ *commands* ]create-links-with *turtleset*create-links-with *turtleset* [ *commands* ]Turtle Command**

Used for creating breeded and unbreeded links between turtles.

create-link-with creates an undirected link between the caller and *agent*. create-link-to creates a directed link from the caller to *agent*. create-link-from creates a directed link from *agent* to the caller.

When the plural form of the breed name is used, an *agentset* is expected instead of an agent and links are created between the caller and all agents in the agentset.

The optional command block is the set of commands each newly formed link runs. (The links are created all at once then run one at a time, in random order.)

A node cannot be linked to itself. Also, you cannot have more than one undirected link of the same breed between the same two nodes, nor can you have more than one directed link of the same breed going in the same direction between two nodes.

If you try to create a link where one (of the same breed) already exists, nothing happens. If you try to create a link from a turtle to itself you get a runtime error.

to setup

clear-all

create-turtles 5

;; turtle 1 creates links with all other turtles

;; the link between the turtle and itself is ignored

ask turtle 0 [ create-links-with other turtles ]

show count links ;; shows 4

;; this does nothing since the link already exists

ask turtle 0 [ create-link-with turtle 1 ]

show count links ;; shows 4 since the previous link already existed

ask turtle 2 [ create-link-with turtle 1 ]

show count links ;; shows 5

end

directed-link-breed [red-links red-link]

undirected-link-breed [blue-links blue-link]

to setup

clear-all

create-turtles 5

;; create links in both directions between turtle 0

;; and all other turtles

ask turtle 0 [ create-red-links-to other turtles ]

ask turtle 0 [ create-red-links-from other turtles ]

show count links ;; shows 8

;; now create undirected links between turtle 0 and other turtles

ask turtle 0 [ create-blue-links-with other turtles ]

show count links ;; shows 12

end

**create-turtles***1.0***crt***1.0*

**create-turtles *number*create-turtles *number* [ *commands* ]create-*<breeds>* *number*create-*<breeds>* *number* [ *commands* ]Observer Command**

Creates *number* new turtles at the origin. New turtles have random integer headings and the color is randomly selected from the 14 primary colors.

If the create-*<breeds>* form is used, the new turtles are created as members of the given breed.

If *commands* are supplied, the new turtles immediately run them. This is useful for giving the new turtles a different color, heading, or whatever. (The new turtles are created all at once then run one at a time, in random order.)

crt 100 [ fd 10 ] ;; makes a randomly spaced circle

breed [canaries canary]

breed [snakes snake]

to setup

clear-all

create-canaries 50 [ set color yellow ]

create-snakes 50 [ set color green ]

end

See also [hatch](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hatch), [sprout](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#sprout).

**create-temporary-plot-pen***1.1*

**create-temporary-plot-pen *string***

A new temporary plot pen with the given name is created in the current plot and set to be the current pen.

Few models will want to use this primitive, because all temporary pens disappear when clear-plot or clear-all-plots are called. The normal way to make a pen is to make a permanent pen in the plot's Edit dialog.

If a pen with that name already exists in the current plot, no new pen is created, and the existing pen is set to the current pen.

The new temporary plot pen has the following initial settings:

* Pen: down
* Color: black
* Mode: 0 (line mode)
* Interval: 1

See: [clear-plot](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#clear-plot), [clear-all-plots](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#clear-all-plots), and [set-current-plot-pen](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#set-current-plot-pen).

**D**

**date-and-time***3.0*

**date-and-time**

Reports a string containing the current date and time. The format is shown below. All fields are fixed width, so they are always at the same locations in the string. The potential resolution of the clock is milliseconds. (Whether you get resolution that high in practice may vary from system to system, depending on the capabilities of the underlying Java Virtual Machine.)

show date-and-time

=> "01:19:36.685 PM 19-Sep-2002"

**die***1.0*

**dieTurtle Command Link Command**

The turtle or link dies.

if xcor > 20 [ die ]

;; all turtles with xcor greater than 20 die

ask links with [color = blue] [ die ]

;; all the blue links will die

A dead agent ceases to exist. The effects of this include:

* The agent will not execute any further code. So if you write ask turtles [ die print "last words?" ], no last words will be printed, because the turtles are already dead before they have a chance to print anything.
* The agent will disappear from any agentsets it was in, reducing the size of those agentsets by one.
* Any variable that was storing the agent will now instead have nobody in it. So for example let x one-of turtles ask x [ die ] print x prints nobody.
* If the dead agent was a turtle, every link connected to it also dies.
* If the observer was watching or following the agent, the observer's perspective resets, as if reset-perspective had been run.

See also: [clear-turtles](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#clear-turtles) [clear-links](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#clear-links)

**diffuse***1.0*

**diffuse *patch-variable* *number*Observer Command**

Tells each patch to give equal shares of (*number* \* 100) percent of the value of *patch-variable* to its eight neighboring patches. *number* should be between 0 and 1. Regardless of topology the sum of *patch-variable* will be conserved across the world. (If a patch has fewer than eight neighbors, each neighbor still gets an eighth share; the patch keeps any leftover shares.)

Note that this is an observer command only, even though you might expect it to be a patch command. (The reason is that it acts on all the patches at once -- patch commands act on individual patches.)

diffuse chemical 0.5

;; each patch diffuses 50% of its variable

;; chemical to its neighboring 8 patches. Thus,

;; each patch gets 1/8 of 50% of the chemical

;; from each neighboring patch.)

**diffuse4***1.0*

**diffuse4 *patch-variable* *number*Observer Command**

Like diffuse, but only diffuses to the four neighboring patches (to the north, south, east, and west), not to the diagonal neighbors.

diffuse4 chemical 0.5

;; each patch diffuses 50% of its variable

;; chemical to its neighboring 4 patches. Thus,

;; each patch gets 1/4 of 50% of the chemical

;; from each neighboring patch.)

**directed-link-breed**

**directed-link-breed [*<link-breeds>* *<link-breed>*]**

This keyword, like the globals and breeds keywords, can only be used at the beginning of the Code tab, before any procedure definitions. It defines a directed link breed. Links of a particular breed are always all directed or all undirected The first input defines the name of the agentset associated with the link breed. The second input defines the name of a single member of the breed. Directed links can be created using [create-link(s)-to](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#create-link), and [create-link(s)-from](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#create-link), but not create-link(s)-with

Any link of the given link breed:

* is part of the agentset named by the link breed name
* has its built-in variable breed set to that agentset
* is directed or undirected as declared by the keyword

Most often, the agentset is used in conjunction with ask to give commands to only the links of a particular breed.

directed-link-breed [streets street]

directed-link-breed [highways highway]

to setup

clear-all

crt 2

;; create a link from turtle 0 to turtle 1

ask turtle 0 [ create-street-to turtle 1 ]

;; create a link from turtle 1 to turtle 0

ask turtle 0 [ create-highway-from turtle 1 ]

end

ask turtle 0 [ show one-of my-in-links ]

;; prints (street 0 1)

ask turtle 0 [ show one-of my-out-links ]

;; prints (highway 1 0)

See also [breed](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#breed), [undirected-link-breed](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#undirected-link-breed)

**display***1.0*

**display**

Causes the view to be updated immediately. (Exception: if the user is using the speed slider to fast-forward the model, then the update may be skipped.)

Also undoes the effect of the no-display command, so that if view updates were suspended by that command, they will resume.

no-display

ask turtles [ jump 10 set color blue set size 5 ]

display

;; turtles move, change color, and grow, with none of

;; their intermediate states visible to the user, only

;; their final state

Even if no-display was not used, "display" can still be useful, because ordinarily NetLogo is free to skip some view updates, so that fewer total updates take place, so that models run faster. This command lets you force a view update, so whatever changes have taken place in the world are visible to the user.

ask turtles [ set color red ]

display

ask turtles [ set color blue]

;; turtles turn red, then blue; use of "display" forces

;; red turtles to appear briefly

Note that display and no-display operate independently of the switch in the view control strip that freezes the view.

See also [no-display](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#no-display).

**distance***1.0*

**distance *agent*Turtle Command Patch Command**

Reports the distance from this agent to the given turtle or patch.

The distance to or a from a patch is measured from the center of the patch. Turtles and patches use the wrapped distance (around the edges of the world) if wrapping is allowed by the topology and the wrapped distance is shorter.

ask turtles [ show max-one-of turtles [distance myself] ]

;; each turtle prints the turtle farthest from itself

**distancexy***1.0*

**distancexy *x* *y*Turtle Command Patch Command**

Reports the distance from this agent to the point (*x*, *y*).

The distance from a patch is measured from the center of the patch. Turtles and patches use the wrapped distance (around the edges of the world) if wrapping is allowed by the topology and the wrapped distance is shorter.

if (distancexy 0 0) > 10

[ set color green ]

;; all turtles more than 10 units from

;; the center of the world turn green.

**downhill***1.0***downhill4***1.0*

**downhill *patch-variable*downhill4 *patch-variable*Turtle Command**

Moves the turtle to the neighboring patch with the lowest value for *patch-variable*. If no neighboring patch has a smaller value than the current patch, the turtle stays put. If there are multiple patches with the same lowest value, the turtle picks one randomly. Non-numeric values are ignored.

downhill considers the eight neighboring patches; downhill4 only considers the four neighbors.

Equivalent to the following code (assumes variable values are numeric):

move-to patch-here ;; go to patch center

let p min-one-of neighbors [*patch-variable*] ;; or neighbors4

if [*patch-variable*] of p < *patch-variable* [

face p

move-to p

]

Note that the turtle always ends up on a patch center and has a heading that is a multiple of 45 (downhill) or 90 (downhill4).

See also [uphill](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#uphill), [uphill4](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#uphill).

**dx***1.0***dy***1.0*

**dxdyTurtle Command**

Reports the x-increment or y-increment (the amount by which the turtle's xcor or ycor would change) if the turtle were to take one step forward in its current heading.

Note: dx is simply the sine of the turtle's heading, and dy is simply the cosine. (If this is the reverse of what you expected, it's because in NetLogo a heading of 0 is north and 90 is east, which is the reverse of how angles are usually defined in geometry.)

Note: In earlier versions of NetLogo, these primitives were used in many situations where the new patch-ahead primitive is now more appropriate.

**E**

**empty?***1.0*

**empty? *list*empty? *string***

Reports true if the given list or string is empty, false otherwise.

Note: the empty list is written []. The empty string is written "".

**end**

**end**

Used to conclude a procedure. See [to](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#to) and [to-report](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#to-report).

**end1***4.0*

**end1Link Command**

This is a built-in link variable. It indicates the first endpoint (turtle) of a link. For directed links this will always be the source for undirected links it will always be the turtle with the lower who number. You cannot set end1.

crt 2

ask turtle 0

[ create-link-to turtle 1 ]

ask links

[ show end1 ] ;; shows turtle 0

**end2***4.0*

**end2Link Command**

This is a built-in link variable. It indicates the second endpoint (turtle) of a link. For directed links this will always be the destination for undirected links it will always be the turtle with the higher who number. You cannot set end2.

crt 2

ask turtle 1

[ create-link-with turtle 0 ]

ask links

[ show end2 ] ;; shows turtle 1

**error***5.0*

**error *value***

Causes a runtime error to occur.

The given value is converted to a string (if it isn't one already) and used as the error message.

See also [error-message](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#error-message), [carefully](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#carefully).

**error-message***2.1*

**error-message**

Reports a string describing the error that was suppressed by carefully.

This reporter can only be used in the second block of a carefully command.

See also [error](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#error), [carefully](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#carefully).

**every***1.0*

**every *number* [ *commands* ]**

Runs the given commands only if it's been more than *number* seconds since the last time this agent ran them in this context. Otherwise, the commands are skipped.

By itself, every doesn't make commands run over and over again. You need to use every inside a loop, or inside a forever button, if you want the commands run over and over again. every only limits how often the commands run.

Above, "in this context" means during the same ask (or button press or command typed in the Command Center). So it doesn't make sense to write ask turtles [ every 0.5 [ ... ] ], because when the ask finishes the turtles will all discard their timers for the "every". The correct usage is shown below.

every 0.5 [ ask turtles [ fd 1 ] ]

;; twice a second the turtles will move forward 1

every 2 [ set index index + 1 ]

;; every 2 seconds index is incremented

See also [wait](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#wait).

**exp***1.0*

**exp *number***

Reports the value of e raised to the *number* power.

Note: This is the same as e ^ *number*.

**export-view***3.0***export-interface***2.0***export-output***1.0***export-plot***1.0***export-all-plots***1.2.1***export-world***1.0*

**export-view *filename*export-interface *filename*export-output *filename*export-plot *plotname* *filename*export-all-plots *filename*export-world *filename***

export-view writes the current contents of the current view to an external file given by the string *filename*. The file is saved in PNG (Portable Network Graphics) format, so it is recommended to supply a filename ending in ".png".

export-interface is similar, but for the whole interface tab.

Note that export-view still works when running NetLogo in headless mode, but export-interface doesn't.

export-output writes the contents of the model's output area to an external file given by the string *filename*. (If the model does not have a separate output area, the output portion of the Command Center is used.)

export-plot writes the x and y values of all points plotted by all the plot pens in the plot given by the string *plotname* to an external file given by the string *filename*. If a pen is in bar mode (mode 0) and the y value of the point plotted is greater than 0, the upper-left corner point of the bar will be exported. If the y value is less than 0, then the lower-left corner point of the bar will be exported.

export-all-plots writes every plot in the current model to an external file given by the string *filename*. Each plot is identical in format to the output of export-plot.

export-world writes the values of all variables, both built-in and user-defined, including all observer, turtle, and patch variables, the drawing, the contents of the output area if one exists, the contents of any plots and the state of the random number generator, to an external file given by the string *filename*. (The result file can be read back into NetLogo with the [import-world](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#import-world) primitive.) export-world does not save the state of open files.

export-plot, export-all-plots and export-world save files in in plain-text, "comma-separated values" (.csv) format. CSV files can be read by most popular spreadsheet and database programs as well as any text editor.

If you wish to export to a file in a location other than the model's location, you should include the full path to the file you wish to export. (Use the forward-slash "/" as the folder separator.)

Note that the functionality of these primitives is also available directly from NetLogo's File menu.

export-world "fire.csv"

;; exports the state of the model to the file fire.csv

;; located in the NetLogo folder

export-plot "Temperature" "c:/My Documents/plot.csv"

;; exports the plot named

;; "Temperature" to the file plot.csv located in

;; the C:\My Documents folder

export-all-plots "c:/My Documents/plots.csv"

;; exports all plots to the file plots.csv

;; located in the C:\My Documents folder

If the file already exists, it is overwritten. To avoid this you may wish to use some method of generating fresh names. Examples:

export-world user-new-file

export-world (word "results " date-and-time ".csv") ;; Colon characters in the time cause errors on Windows

export-world (word "results " random-float 1.0 ".csv")

**extensions**

**extensions [*name* ...]**

Allows the model to use primitives from the extensions with the given names. See the [Extensions guide](http://ccl.northwestern.edu/netlogo/docs/extensions.html) for more information.

**extract-hsb***1.0*

**extract-hsb *color***

Reports a list of three values, the first (hue) in the range of 0 to 360, the second and third (brightness and saturation) in the range of 0 to 100.

The given *color* can either be a NetLogo color in the range 0 to 140, not including 140 itself, or an RGB list of three values in the range 0 to 255 representing the levels of red, green, and blue.

show extract-hsb cyan

=> [180 57.143 76.863]

show extract-hsb red

=> [3.103 80.93 84.314]

show extract-hsb [255 0 0]

=> [0 100 100]

See also [approximate-hsb](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#approximate-hsb), [approximate-rgb](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#approximate-rgb), [extract-rgb](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#extract-rgb).

**extract-rgb***1.0*

**extract-rgb *color***

Reports a list of three values in the range 0 to 255 representing the levels of red, green, and blue, respectively, of the given NetLogo *color* in the range 0 to 140, not including 140 itself.

show extract-rgb red

=> [215 50 41]

show extract-rgb cyan

=> [84 196 196]

See also [approximate-rgb](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#approximate-rgb), [approximate-hsb](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#approximate-hsb), [extract-hsb](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#extract-hsb).

**F**

**face***3.0*

**face *agent*Turtle Command**

Set the caller'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 and the agent are at the exact same position, the caller's heading won't change.

**facexy***3.0*

**facexy *x* *y*Turtle Command**

Set the caller's heading towards the point (x,y).

If wrapping is allowed by the topology and the wrapped distance (around the edges of the world) is shorter and wrapping is allowed, facexy will use the wrapped path.

If the caller is on the point (x,y), the caller's heading won't change.

**file-at-end?***2.0*

**file-at-end?**

Reports true when there are no more characters left to read in from the current file (that was opened previously with [file-open](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-open)). Otherwise, reports false.

file-open "my-file.txt"

print file-at-end?

=> false ;; Can still read in more characters

print file-read-line

=> This is the last line in file

print file-at-end?

=> true ;; We reached the end of the file

See also [file-open](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-open), [file-close-all](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-close-all).

**file-close***2.0*

**file-close**

Closes a file that has been opened previously with [file-open](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-open).

Note that this and file-close-all are the only ways to restart to the beginning of an opened file or to switch between file modes.

If no file is open, does nothing.

See also [file-close-all](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-close-all), [file-open](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-open).

**file-close-all***2.0*

**file-close-all**

Closes all files (if any) that have been opened previously with [file-open](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-open).

See also [file-close](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-close), [file-open](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-open).

**file-delete***2.0*

**file-delete *string***

Deletes the file specified as *string*

*string* must be an existing file with writable permission by the user. Also, the file cannot be open. Use the command [file-close](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-close) to close an opened file before deletion.

Note that the string can either be a file name or an absolute file path. If it is a file name, it looks in whatever the current directory is. This can be changed using the command [set-current-directory](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#set-current-directory). It is defaulted to the model's directory.

**file-exists?***2.0*

**file-exists? *string***

Reports true if *string* is the name of an existing file on the system. Otherwise it reports false.

Note that the string can either be a file name or an absolute file path. If it is a file name, it looks in whatever the current directory is. This can be changed using the command [set-current-directory](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#set-current-directory). It defaults to to the model's directory.

**file-flush***4.0*

**file-flush**

Forces file updates to be written to disk. When you use file-write or other output commands, the values may not be immediately written to disk. This improves the performance of the file output commands. Closing a file ensures that all output is written to disk.

Sometimes you need to ensure that data is written to disk without closing the file. For example, you could be using a file to communicate with another program on your machine and want the other program to be able to see the output immediately.

**file-open***2.0*

**file-open *string***

This command will interpret *string* as a path name to a file and open the file. You may then use the reporters [file-read](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-read), [file-read-line](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-read-line), and [file-read-characters](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-read-characters) to read in from the file, or [file-write](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-write), [file-print](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-print), [file-type](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-type), or [file-show](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-show) to write out to the file.

Note that you can only open a file for reading or writing but not both. The next file i/o primitive you use after this command dictates which mode the file is opened in. To switch modes, you need to close the file using [file-close](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-close).

Also, the file must already exist if opening a file in reading mode.

When opening a file in writing mode, all new data will be appended to the end of the original file. If there is no original file, a new blank file will be created in its place. (You must have write permission in the file's directory.) (If you don't want to append, but want to replace the file's existing contents, use [file-delete](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-delete) to delete it first, perhaps inside a [carefully](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#carefully) if you're not sure whether it already exists.)

Note that the string can either be a file name or an absolute file path. If it is a file name, it looks in whatever the current directory is. This can be changed using the command [set-current-directory](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#set-current-directory). It is defaulted to the model's directory.

file-open "my-file-in.txt"

print file-read-line

=> First line in file ;; File is in reading mode

file-open "C:\\NetLogo\\my-file-out.txt"

;; assuming Windows machine

file-print "Hello World" ;; File is in writing mode

Opening a file does not close previously opened files. You can use file-open to switch back and forth between multiple open files.

See also [file-close](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-close) See also [file-close-all](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-close-all).

**file-print***2.0*

**file-print *value***

Prints *value* to an opened file, followed by a carriage return.

This agent is *not* printed before the value, unlike [file-show](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-show).

Note that this command is the file i/o equivalent of [print](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#print), and [file-open](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-open) needs to be called before this command can be used.

See also [file-show](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-show), [file-type](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-type), [file-write](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-write), and [Output (programming guide)](http://ccl.northwestern.edu/netlogo/docs/programming.html#output).

**file-read***2.0*

**file-read**

This reporter will read in the next constant from the opened file and interpret it as if it had been typed in the Command Center. It reports the resulting value. The result may be a number, list, string, boolean, or the special value nobody.

Whitespace separates the constants. Each call to file-read will skip past both leading and trailing whitespace.

Note that strings need to have quotes around them. Use the command [file-write](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-write) to have quotes included.

Also note that the [file-open](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-open) command must be called before this reporter can be used, and there must be data remaining in the file. Use the reporter [file-at-end?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-at-end) to determine if you are at the end of the file.

file-open "my-file.data"

print file-read + 5

;; Next value is the number 1

=> 6

print length file-read

;; Next value is the list [1 2 3 4]

=> 4

See also [file-open](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-open) and [file-write](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-write).

**file-read-characters***2.0*

**file-read-characters *number***

Reports the given *number* of characters from an opened file as a string. If there are fewer than that many characters left, it will report all of the remaining characters.

Note that it will return every character including newlines and spaces.

Also note that the [file-open](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-open) command must be called before this reporter can be used, and there must be data remaining in the file. Use the reporter [file-at-end?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-at-end) to determine if you are at the end of the file.

file-open "my-file.txt"

print file-read-characters 5

;; Current line in file is "Hello World"

=> Hello

See also [file-open](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-open).

**file-read-line***2.0*

**file-read-line**

Reads the next line in the file and reports it as a string. It determines the end of the file by a carriage return, an end of file character or both in a row. It does not return the line terminator characters.

Also note that the [file-open](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-open) command must be called before this reporter can be used, and there must be data remaining in the file. Use the reporter [file-at-end?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-at-end) to determine if you are at the end of the file.

file-open "my-file.txt"

print file-read-line

=> Hello World

See also [file-open](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-open).

**file-show***2.0*

**file-show *value***

Prints *value* to an opened file, preceded by this agent agent, and followed by a carriage return. (This agent is included to help you keep track of what agents are producing which lines of output.) Also, all strings have their quotes included similar to [file-write](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-write).

Note that this command is the file i/o equivalent of [show](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#show), and [file-open](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-open) needs to be called before this command can be used.

See also [file-print](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-print), [file-type](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-type), [file-write](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-write), and [Output (programming guide)](http://ccl.northwestern.edu/netlogo/docs/programming.html#output).

**file-type***2.0*

**file-type *value***

Prints *value* to an opened file, *not* followed by a carriage return (unlike [file-print](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-print) and [file-show](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-show)). The lack of a carriage return allows you to print several values on the same line.

This agent is *not* printed before the value. unlike [file-show](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-show).

Note that this command is the file i/o equivalent of [type](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#type), and [file-open](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-open) needs to be called before this command can be used.

See also [file-print](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-print), [file-show](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-show), [file-write](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-write), and [Output (programming guide)](http://ccl.northwestern.edu/netlogo/docs/programming.html#output).

**file-write***2.0*

**file-write *value***

This command will output *value*, which can be a number, string, list, boolean, or nobody to an opened file, *not* followed by a carriage return (unlike [file-print](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-print) and [file-show](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-show)).

This agent is *not* printed before the value, unlike [file-show](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-show). Its output also includes quotes around strings and is prepended with a space. It will output the value in such a manner that [file-read](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-read) will be able to interpret it.

Note that this command is the file i/o equivalent of [write](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#write), and [file-open](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-open) needs to be called before this command can be used.

file-open "locations.txt"

ask turtles

[ file-write xcor file-write ycor ]

See also [file-print](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-print), [file-show](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-show), [file-type](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-type), and [Output (programming guide)](http://ccl.northwestern.edu/netlogo/docs/programming.html#output).

**filter***1.3*

**filter *reporter* *list***

Reports a list containing only those items of *list* for which the reporter reports true -- in other words, the items satisfying the given condition. *reporter* may be an anonymous reporter or the name of a reporter.

show filter is-number? [1 "2" 3]

=> [1 3]

show filter [ i -> i < 3 ] [1 3 2]

=> [1 2]

show filter [ s -> first s != "t" ] ["hi" "there" "everyone"]

=> ["hi" "everyone"]

See also [map](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#map), [reduce](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#reduce), [-> (anonymous procedure)](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#arrow).

**first***1.0*

**first *list*first *string***

On a list, reports the first (0th) item in the list.

On a string, reports a one-character string containing only the first character of the original string.

**floor***1.0*

**floor *number***

Reports the largest integer less than or equal to *number*.

show floor 4.5

=> 4

show floor -4.5

=> -5

See also [ceiling](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#ceiling), [round](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#round), [precision](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#precision).

**follow***3.0*

**follow *turtle*Observer Command**

Similar to ride, but, in the 3D view, the observer's vantage point is behind and above *turtle*.

The observer may only watch or follow a single subject. Calling follow will alter the highlight created by prior calls to watch and watch-me, highlighting the followed turtle instead.

See also [follow-me](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#follow-me), [ride](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#ride), [reset-perspective](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#reset-perspective), [watch](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#watch), [subject](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#subject).

**follow-me***3.0*

**follow-meTurtle Command**

Asks the observer to follow this turtle.

The observer may only watch or follow a single subject. Calling follow-me will remove the highlight created by prior calls to watch and watch-me, highlighting this turtle instead.

See also [follow](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#follow).

**foreach***1.3*

**foreach *list* *command*(foreach *list1* ... *command*)**

With a single list, runs the command for each item of *list*. *command* may be the name of a command, or an anonymous command created with [->](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#arrow).

foreach [1.1 2.2 2.6] show

=> 1.1

=> 2.2

=> 2.6

foreach [1.1 2.2 2.6] [ x -> show (word x " -> " round x) ]

=> 1.1 -> 1

=> 2.2 -> 2

=> 2.6 -> 3

With multiple lists, runs *command* for each group of items from each list. So, they are run once for the first items, once for the second items, and so on. All the lists must be the same length.

Some examples make this clearer:

(foreach [1 2 3] [2 4 6]

[ [a b] -> show word "the sum is: " (a + b) ])

=> "the sum is: 3"

=> "the sum is: 6"

=> "the sum is: 9"

(foreach list (turtle 1) (turtle 2) [3 4]

[ [the-turtle num-steps] -> ask the-turtle [ fd num-steps ] ])

;; turtle 1 moves forward 3 patches

;; turtle 2 moves forward 4 patches

See also [map](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#map), [-> (anonymous procedure)](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#arrow).

**forward***1.0***fd***1.0*

**forward *number*Turtle Command**

The turtle moves forward by *number* steps, one step at a time. (If *number* is negative, the turtle moves backward.)

fd 10 is equivalent to repeat 10 [ jump 1 ]. fd 10.5 is equivalent to repeat 10 [ jump 1 ] jump 0.5.

If the turtle cannot move forward *number* steps because it is not permitted by the current topology the turtle will complete as many steps of 1 as it can, then stop.

See also [jump](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#jump), [can-move?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#can-move).

**fput***1.0*

**fput *item list***

Adds *item* to the beginning of a list and reports the new list.

;; suppose mylist is [5 7 10]

set mylist fput 2 mylist

;; mylist is now [2 5 7 10]

**G**

**globals**

**globals [*var1* ...]**

This keyword, like the breed, *<breeds>*-own, patches-own, and turtles-own keywords, can only be used at the beginning of a program, before any function definitions. It defines new global variables. Global variables are "global" because they are accessible by all agents and can be used anywhere in a model.

Most often, globals is used to define variables or constants that need to be used in many parts of the program.

**H**

**hatch***1.0*

**hatch *number* [ *commands* ]hatch-*<breeds>* *number* [ *commands* ]Turtle Command**

This turtle creates *number* new 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, and it may be of a different breed than its parent if the hatch-*<breeds>* form is used.)

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.)

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.

hatch 1 [ lt 45 fd 1 ]

;; this turtle 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

See also [create-turtles](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#create-turtles), [sprout](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#sprout).

**heading**

**headingTurtle Command**

This is a built-in turtle variable. It indicates the direction the turtle is facing. This is a number greater than or equal to 0 and less than 360. 0 is north, 90 is east, and so on. You can set this variable to make a turtle turn.

See also [right](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#right), [left](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#left), [dx](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#dxy), [dy](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#dxy).

Example:

set heading 45 ;; turtle is now facing northeast

set heading heading + 10 ;; same effect as "rt 10"

**hidden?**

**hidden?Turtle Command Link Command**

This is a built-in turtle or link variable. It holds a boolean (true or false) value indicating whether the turtle or link is currently hidden (i.e., invisible). You can set this variable to make a turtle or link disappear or reappear.

See also [hide-turtle](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hide-turtle), [show-turtle](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#show-turtle), [hide-link](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hide-link), [show-link](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#show-link)

Example:

set hidden? not hidden?

;; if turtle was showing, it hides, and if it was hiding,

;; it reappears

**hide-link***4.0*

**hide-linkLink Command**

The link makes itself invisible.

Note: This command is equivalent to setting the link variable "hidden?" to true.

See also [show-link](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#show-turtle).

**hide-turtle***1.0***ht***1.0*

**hide-turtleTurtle Command**

The turtle makes itself invisible.

Note: This command is equivalent to setting the turtle variable "hidden?" to true.

See also [show-turtle](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#show-turtle).

**histogram***1.0*

**histogram *list***

Histograms the values in the given list

Draws a histogram showing the frequency distribution of the values in the list. The heights of the bars in the histogram represent the numbers of values in each subrange.

Before the histogram is drawn, first any previous points drawn by the current plot pen are removed.

Any non-numeric values in the list are ignored.

The histogram is drawn on the current plot using the current plot pen and pen color. Auto scaling does not affect a histogram's horizontal range, so set-plot-x-range should be used to control the range, and the pen interval can then be set (either directly with set-plot-pen-interval, or indirectly via set-histogram-num-bars) to control how many bars that range is split up into.

Be sure that if you want the histogram drawn with bars that the current pen is in bar mode (mode 1).

For histogramming purposes the plot's X range is not considered to include the maximum X value. Values equal to the maximum X will fall outside of the histogram's range.

histogram [color] of turtles

;; draws a histogram showing how many turtles there are

;; of each color

**home***1.0*

**homeTurtle Command**

This turtle moves to the origin (0,0). Equivalent to setxy 0 0.

**hsb***1.0*

**hsb *hue saturation brightness***

Reports a RGB list when given three numbers describing an HSB color. Hue, saturation, and brightness are integers in the range 0-360, 0-100, 0-100 respectively. The RGB list contains three integers in the range of 0-255.

See also [rgb](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#rgb)

**hubnet-broadcast***1.1*

**hubnet-broadcast *tag-name value***

This broadcasts *value* from NetLogo to the interface element with the name *tag-name* on the clients.

See the [HubNet Authoring Guide](http://ccl.northwestern.edu/netlogo/docs/hubnet-authoring.html) for details and instructions.

**hubnet-broadcast-clear-output***4.1*

**hubnet-broadcast-clear-output**

This clears all messages printed to the text area on every client.

See also: [hubnet-broadcast-message](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-broadcast-message), [hubnet-send-clear-output](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-send-clear-output)

**hubnet-broadcast-message***4.1*

**hubnet-broadcast-message *value***

This prints the value in the text area on each client. This is the same functionality as the "Broadcast Message" button in the HubNet Control Center.

See also: [hubnet-send-message](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-send-message)

**hubnet-clear-override***4.1***hubnet-clear-overrides***4.1*

**hubnet-clear-override *client* *agent-or-set* *variable-name*hubnet-clear-overrides *client***

Remove overrides from the override list on *client*. hubnet-clear-override removes only the override for the specified variable for the specified agent or agentset. hubnet-clear-overrides removes all overrides from the specified client.

See also: [hubnet-send-override](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-send-override)

**hubnet-clients-list***5.0*

**hubnet-clients-list**

Reports a list containing the names of all the clients currently connected to the HubNet server.

**hubnet-enter-message?***1.2.1*

**hubnet-enter-message?**

Reports true if a new client just entered the simulation. Reports false otherwise. [hubnet-message-source](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-message-source) will contain the user name of the client that just logged on.

See the [HubNet Authoring Guide](http://ccl.northwestern.edu/netlogo/docs/hubnet-authoring.html) for details and instructions.

**hubnet-exit-message?***1.2.1*

**hubnet-exit-message?**

Reports true if a client just exited the simulation. Reports false otherwise. [hubnet-message-source](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-message-source) will contain the user name of the client that just logged off.

See the [HubNet Authoring Guide](http://ccl.northwestern.edu/netlogo/docs/hubnet-authoring.html) for details and instructions.

**hubnet-fetch-message***1.1*

**hubnet-fetch-message**

If there is any new data sent by the clients, this retrieves the next piece of data, so that it can be accessed by [hubnet-message](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-message), [hubnet-message-source](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-message-source), and [hubnet-message-tag](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-message-tag). This will cause an error if there is no new data from the clients.

See the [HubNet Authoring Guide](http://ccl.northwestern.edu/netlogo/docs/hubnet-authoring.html) for details.

**hubnet-kick-client***5.0*

**hubnet-kick-client *client-name***

Kicks the client with the given client-name. This is equivalent to clicking the client name in the HubNet Control Center and pressing the Kick button.

**hubnet-kick-all-clients***5.0*

**hubnet-kick-all-clients**

Kicks out all currently connected HubNet clients. This is equivalent to selecting all clients in the HubNet Control Center and pressing the Kick button.

**hubnet-message***1.1*

**hubnet-message**

Reports the message retrieved by [hubnet-fetch-message](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-fetch-message).

See the [HubNet Authoring Guide](http://ccl.northwestern.edu/netlogo/docs/hubnet-authoring.html) for details.

**hubnet-message-source***1.1*

**hubnet-message-source**

Reports the name of the client that sent the message retrieved by [hubnet-fetch-message](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-fetch-message).

See the [HubNet Authoring Guide](http://ccl.northwestern.edu/netlogo/docs/hubnet-authoring.html) for details.

**hubnet-message-tag***1.1*

**hubnet-message-tag**

Reports the tag that is associated with the data that was retrieved by [hubnet-fetch-message](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-fetch-message). The tag will be one of the Display Names of the interface elements in the client interface.

See the [HubNet Authoring Guide](http://ccl.northwestern.edu/netlogo/docs/hubnet-authoring.html) for details.

**hubnet-message-waiting?***1.1*

**hubnet-message-waiting?**

This looks for a new message sent by the clients. It reports true if there is one, and false if there is not.

See the [HubNet Authoring Guide](http://ccl.northwestern.edu/netlogo/docs/hubnet-authoring.html) for details.

**hubnet-reset***1.1*

**hubnet-reset**

Starts up the HubNet system. HubNet must be started to use any of the other hubnet primitives.

See the [HubNet Authoring Guide](http://ccl.northwestern.edu/netlogo/docs/hubnet-authoring.html) for details.

**hubnet-reset-perspective***4.1*

**hubnet-reset-perspective *tag-name***

Clears watch or follow sent directly to the client. The view perspective will revert to the server perspective.

See also: [hubnet-send-watch](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-send-watch) [hubnet-send-follow](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-send-follow)

**hubnet-send***1.1*

**hubnet-send *string tag-name value***

**hubnet-send *list-of-strings tag-name value***

For a *string*, this sends *value* from NetLogo to the tag *tag-name* on the client that has *string* for its user name.

For a *list-of-strings*, this sends *value* from NetLogo to the tag *tag-name* on all the clients that have a user name that is in the *list-of-strings*.

Sending a message to a non-existent client, using hubnet-send, generates a hubnet-exit-message.

See the [HubNet Authoring Guide](http://ccl.northwestern.edu/netlogo/docs/hubnet-authoring.html) for details.

**hubnet-send-clear-output***4.1*

**hubnet-send-clear-output *string***

**hubnet-send-clear-output *list-of-strings***

This clears all messages printed to the text area on the given client or clients (specified in the *string* or *list-of-strings*.

See also: [hubnet-send-message](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-send-message), [hubnet-broadcast-clear-output](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-broadcast-clear-output)

**hubnet-send-follow***4.1*

**hubnet-send-follow *client-name agent radius***

Tells the client associated with *client-name* to follow *agent* showing a *radius* sized Moore neighborhood around the agent.

A client may only watch or follow a single subject. Calling hubnet-send-follow will alter the highlight created by prior calls to hubnet-send-watch, highlighting the followed agent instead.

See also: [hubnet-send-watch](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-send-watch), [hubnet-reset-perspective](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-reset-perspective)

**hubnet-send-message***4.1*

**hubnet-send-message *string* *value***

This prints value in the text area on the client specified by string.

See also: [hubnet-broadcast-message](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-broadcast-message)

**hubnet-send-override***4.1*

**hubnet-send-override *client-name agent-or-set variable-name*[ *reporter* ]**

Evaluates *reporter* for the agent or agentset indicated then sends the values to the client to "override" the value of *variable-name* only on *client-name*. This is used to change the appearance of agents in the client view, hence, only built-in variables that affect the appearance of the agent may be selected. For example, you can override the color variable of a turtle:

ask turtles [ hubnet-send-override client-name self "color" [ red ] ]

In this example assume that there is a turtles-own variable client-name which is associated with a logged in client, and all the turtles are blue. This code makes the turtle associated with each client appear red in his or her own view but not on anyone else's or on the server.

See also: [hubnet-clear-overrides](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-clear-override)

**hubnet-send-watch***4.1*

**hubnet-send-watch *client-name agent***

Tells the client associated with *client-name* to watch *agent*.

A client may only watch or follow a single subject. Calling hubnet-send-watch will undo perspective changes caused by prior calls to hubnet-send-follow.

See also: [hubnet-send-follow](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-send-follow), [hubnet-reset-perspective](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hubnet-reset-perspective)

**I**

**if***1.0*

**if *condition* [ *commands* ]**

Reporter must report a boolean (true or false) value.

If *condition* reports true, runs *commands*.

The reporter may report a different value for different agents, so some agents may run *commands* and others don't.

if xcor > 0[ set color blue ]

;; turtles in the right half of the world

;; turn blue

See also [ifelse](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#ifelse), [ifelse-value](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#ifelse-value).

**ifelse***1.0*

**ifelse *reporter* [ *commands1* ] [ *commands2* ]**

Reporter must report a boolean (true or false) value.

If *reporter* reports true, runs *commands1*.

If *reporter* reports false, runs *commands2*.

The reporter may report a different value for different agents, so some agents may run *commands1* while others run *commands2*.

ask patches

[ ifelse pxcor > 0

[ set pcolor blue ]

[ set pcolor red ] ]

;; the left half of the world turns red and

;; the right half turns blue

See also [if](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#if), [ifelse-value](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#ifelse-value).

**ifelse-value***2.0*

**ifelse-value *reporter* [*reporter1*] [*reporter2*]**

Reporter must report a boolean (true or false) value.

If *reporter* reports true, the result is the value of *reporter1*.

If *reporter* reports false, the result is the value of *reporter2*.

This can be used when a conditional is needed in the context of a reporter, where commands (such as [ifelse](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#ifelse)) are not allowed.

ask patches [

set pcolor ifelse-value (pxcor > 0) [blue] [red]

]

;; the left half of the world turns red and

;; the right half turns blue

show n-values 10 [ifelse-value (? < 5) [0] [1]]

=> [0 0 0 0 0 1 1 1 1 1]

show reduce [ [a b] -> ifelse-value (a > b) [a] [b] ]

[1 3 2 5 3 8 3 2 1]

=> 8

See also [if](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#if), [ifelse](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#ifelse).

**import-drawing***3.0*

**import-drawing *filename*Observer Command**

Reads an image file into the drawing, scaling it to the size of the world, while retaining the original aspect ratio of the image. The image is centered in the drawing. The old drawing is not cleared first.

Agents cannot sense the drawing, so they cannot interact with or process images imported by import-drawing. If you need agents to sense an image, use [import-pcolors](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#import-pcolors) or [import-pcolors-rgb](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#import-pcolors-rgb).

The following image file formats are supported: BMP, JPG, GIF, and PNG. If the image format supports transparency (alpha), that information will be imported as well.

**import-pcolors***3.0*

**import-pcolors *filename*Observer Command**

Reads an image file, scales it to the same dimensions as the patch grid while maintaining the original aspect ratio of the image, and transfers the resulting pixel colors to the patches. The image is centered in the patch grid. The resulting patch colors may be distorted, since the NetLogo color space does not include all possible colors. (See the Color section of the Programming Guide.) import-pcolors may be slow for some images, particularly when you have many patches and a large image with many different colors.

Since import-pcolors sets the pcolor of patches, agents can sense the image. This is useful if agents need to analyze, process, or otherwise interact with the image. If you want to simply display a static backdrop, without color distortion, see [import-drawing](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#import-drawing).

The following image file formats are supported: BMP, JPG, GIF, and PNG. If the image format supports transparency (alpha), then all fully transparent pixels will be ignored. (Partially transparent pixels will be treated as opaque.)

**import-pcolors-rgb***4.0*

**import-pcolors-rgb *filename*Observer Command**

Reads an image file, scales it to the same dimensions as the patch grid while maintaining the original aspect ratio of the image, and transfers the resulting pixel colors to the patches. The image is centered in the patch grid. Unlike [import-pcolors](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#import-pcolors) the exact colors in the original image are retained. The pcolor variable of all the patches will be an RGB list rather than an (approximated) NetLogo color.

The following image file formats are supported: BMP, JPG, GIF, and PNG. If the image format supports transparency (alpha), then all fully transparent pixels will be ignored. (Partially transparent pixels will be treated as opaque.)

**import-world***1.0*

**import-world *filename*Observer Command**

Reads the values of all variables for a model, both built-in and user-defined, including all observer, turtle, and patch variables, from an external file named by the given string. The file should be in the format used by the [export-world](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#export-cmds) primitive.

Note that the functionality of this primitive is also directly available from NetLogo's File menu.

When using import-world, to avoid errors, perform these steps in the following order:

1. Open the model from which you created the export file.
2. Press the Setup button, to get the model in a state from which it can be run.
3. Import the file.
4. Re-open any files that the model had opened with the file-open command.
5. If you want, press Go button to continue running the model from the point where it left off.

If you wish to import a file from a location other than the model's location, you may include the full path to the file you wish to import. See [export-world](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#export-cmds) for an example.

**in-cone***3.0*

***agentset* in-cone *distance* *angle*Turtle Command**

This reporter lets you give a turtle a "cone of vision" in front of itself. The cone is defined by the two inputs, the vision distance (radius) and the viewing angle. The viewing angle may range from 0 to 360 and is centered around the turtle's current heading. (If the angle is 360, then in-cone is equivalent to in-radius.)

in-cone reports an agentset that includes only those agents from the original agentset that fall in the cone. (This can include the agent itself.)

The distance to a patch is measured from the center of the patch.

ask turtles

[ ask patches in-cone 3 60

[ set pcolor red ] ]

;; each turtle makes a red "splotch" of patches in a 60 degree

;; cone of radius 3 ahead of itself

**in-<breed>-neighbor?in-link-neighbor?***4.0*

**in-<breed>-neighbor? *agent*in-link-neighbor? *turtle*Turtle Command**

Reports true if there is a directed link going from *turtle* to the caller or an undirected link connecting *turtle* to the caller. You can think of this as "is there a link I can use to get from *turtle* to the caller?"

crt 2

ask turtle 0 [

create-link-to turtle 1

show in-link-neighbor? turtle 1 ;; prints false

show out-link-neighbor? turtle 1 ;; prints true

]

ask turtle 1 [

show in-link-neighbor? turtle 0 ;; prints true

show out-link-neighbor? turtle 0 ;; prints false

]

**in-<breed>-neighborsin-link-neighbors***4.0*

**in-<breed>-neighborsin-link-neighborsTurtle Command**

Reports the agentset of all the turtles that have directed links coming from them to the caller as well as all turtles that have an undirected link connecting them with the caller. You can think of this as "all the turtles that can get to the caller using a link."

crt 4

ask turtle 0 [ create-links-to other turtles ]

ask turtle 1 [ ask in-link-neighbors [ set color blue ] ] ;; turtle 0 turns blue

**in-<breed>-fromin-link-from***4.0*

**in-<breed>-from *turtle*in-link-from *turtle*Turtle Command**

Reports a directed link from *turtle* to the caller or an undirected link connecting the two. If no link exists then it reports nobody. If more than one such link exists, reports a random one. You can think of this as "give me a link that I can use to travel from *turtle* to the caller."

crt 2

ask turtle 0 [ create-link-to turtle 1 ]

ask turtle 1 [ show in-link-from turtle 0 ] ;; shows link 0 1

ask turtle 0 [ show in-link-from turtle 1 ] ;; shows nobody

See also: [out-link-to](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#out-link-to) [link-with](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#link-with)

**\_\_includes***4.0*

**\_\_includes [ *filename* ... ]**

Causes external NetLogo source files (with the .nls suffix) to be included in this model. Included files may contain breed, variable, and procedure definitions. \_\_includes can only be used once per file.

The file names must be strings, for example:

\_\_includes [ "utils.nls" ]

Or, for multiple files:

\_\_includes [ "utils1.nls" "utils2.nls" ]

**in-radius***1.0*

***agentset* in-radius *number*Turtle Command Patch Command**

Reports an agentset that includes only those agents from the original agentset whose distance from the caller is less than or equal to *number*. (This can include the agent itself.)

The distance to or a from a patch is measured from the center of the patch.

ask turtles

[ ask patches in-radius 3

[ set pcolor red ] ]

;; each turtle makes a red "splotch" around itself

**insert-item***6.0.2*

**insert-item *index list value*insert-item *index string1 string2***

On a list, inserts an item in that list. *index* is the index where the item will be inserted. The first item has an index of 0. (The 6th item in a list would have an index of 5.)

Likewise for a string, but all characters in a multiple-character *string2* are inserted at *index*.

show insert-item 2 [2 7 4 5] 15

=> [2 7 15 4 5]

show insert-item 2 "cat" "re"

=> "caret"

**inspect***1.1*

**inspect *agent***

Opens an agent monitor for the given agent (turtle or patch).

inspect patch 2 4

;; an agent monitor opens for that patch

inspect one-of sheep

;; an agent monitor opens for a random turtle from

;; the "sheep" breed

See [stop-inspecting](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#stop-inspecting) and [stop-inspecting-dead-agents](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#stop-inspecting-dead-agents)

**int***1.0*

**int *number***

Reports the integer part of number -- any fractional part is discarded.

show int 4.7

=> 4

show int -3.5

=> -3

**is-agent?***1.2.1***is-agentset?***1.2.1***is-anonymous-command?***6.0***is-anonymous-reporter?***6.0***is-boolean?***1.2.1***is-directed-link?***4.0***is-link?***4.0***is-link-set?***4.0***is-list?***1.0***is-number?***1.2.1***is-patch?***1.2.1***is-patch-set?***4.0***is-string?***1.0***is-turtle?***1.2.1***is-turtle-set?***4.0***is-undirected-link?***4.0*

**is-agent? *value*is-agentset? *value*is-anonymous-command? *value*is-anonymous-reporter? *value*is-boolean? *value*is-*<breed>*? *value*is-*<link-breed>*? *value*is-directed-link? *value*is-link? *value*is-link-set? *value*is-list? *value*is-number? *value*is-patch? *value*is-patch-set? *value*is-string? *value*is-turtle? *value*is-turtle-set? *value*is-undirected-link? *value***

Reports true if *value* is of the given type, false otherwise.

**item***1.0*

**item *index list*item *index string***

On lists, reports the value of the item in the given list with the given index.

On strings, reports the character in the given string at the given index.

Note that the indices begin from 0, not 1. (The first item is item 0, the second item is item 1, and so on.)

;; suppose mylist is [2 4 6 8 10]

show item 2 mylist

=> 6

show item 3 "my-shoe"

=> "s"

**J**

**jump***1.0*

**jump *number*Turtle Command**

The turtle moves forward by *number* units all at once (rather than one step at a time as with the forward command).

If the turtle cannot jump *number* units because it is not permitted by the current topology the turtle does not move at all.

See also [forward](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#forward), [can-move?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#can-move).

**L**

**label**

**labelTurtle Command Link Command**

This is a built-in turtle or link variable. It may hold a value of any type. The turtle or link appears in the view with the given value "attached" to it as text. You can set this variable to add, change, or remove a turtle or link's label.

See also [label-color](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#label-color), [plabel](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#plabel), [plabel-color](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#plabel-color).

Example:

ask turtles [ set label who ]

;; all the turtles now are labeled with their

;; who numbers

ask turtles [ set label "" ]

;; all turtles now are not labeled

**label-color**

**label-colorTurtle Command Link Command**

This is a built-in turtle or link variable. It holds a number greater than or equal to 0 and less than 140. This number determines what color the turtle or link's label appears in (if it has a label). You can set this variable to change the color of a turtle or link's label.

See also [label](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#label), [plabel](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#plabel), [plabel-color](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#plabel-color).

Example:

ask turtles [ set label-color red ]

;; all the turtles now have red labels

**last***1.0*

**last *list*last *string***

On a list, reports the last item in the list.

On a string, reports a one-character string containing only the last character of the original string.

**layout-circle***4.0*

**layout-circle *agentset* *radius*layout-circle *list-of-turtles* *radius***

Arranges the given turtles in a circle centered on the patch at the center of the world with the given radius. (If the world has an even size the center of the circle is rounded down to the nearest patch.) The turtles point outwards.

If the first input is an agentset, the turtles are arranged in random order.

If the first input is a list, the turtles are arranged clockwise in the given order, starting at the top of the circle. (Any non-turtles in the list are ignored.)

;; in random order

layout-circle turtles 10

;; in order by who number

layout-circle sort turtles 10

;; in order by size

layout-circle sort-by [ [a b] -> [size] of a < [size] of b ] turtles 10

**layout-radial***4.0*

**layout-radial *turtle-set* *link-set* *root-agent***

Arranges the turtles in *turtle-set* connected by links in *link-set*, in a radial tree layout, centered around the *root-agent* which is moved to the center of the world view.

Only links in the *link-set* will be used to determine the layout. If links connect turtles that are not in *turtle-set* those turtles will remain stationary.

Even if the network does contain cycles, and is not a true tree structure, this layout will still work, although the results will not always be pretty.

to make-a-tree

set-default-shape turtles "circle"

crt 6

ask turtle 0 [

create-link-with turtle 1

create-link-with turtle 2

create-link-with turtle 3

]

ask turtle 1 [

create-link-with turtle 4

create-link-with turtle 5

]

; do a radial tree layout, centered on turtle 0

layout-radial turtles links (turtle 0)

end

**layout-spring***4.0*

**layout-spring *turtle-set* *link-set* *spring-constant* *spring-length* *repulsion-constant***

Arranges the turtles in *turtle-set*, as if the links in *link-set* are springs and the turtles are repelling each other. Turtles that are connected by links in *link-set* but not included in *turtle-set* are treated as anchors and are not moved.

*spring-constant* is a measure of the "tautness" of the spring. It is the "resistance" to change in their length. spring-constant is the force the spring would exert if it's length were changed by 1 unit.

spring-length is the "zero-force" length or the natural length of the springs. This is the length which all springs try to achieve either by pushing out their nodes or pulling them in.

repulsion-constant is a measure of repulsion between the nodes. It is the force that 2 nodes at a distance of 1 unit will exert on each other.

The repulsion effect tries to get the nodes as far as possible from each other, in order to avoid crowding and the spring effect tries to keep them at "about" a certain distance from the nodes they are connected to. The result is the laying out of the whole network in a way which highlights relationships among the nodes and at the same time is crowded less and is visually pleasing.

The layout algorithm is based on the Fruchterman-Reingold layout algorithm. More information about this algorithm can be obtained [here](http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.13.8444).

to make-a-triangle

set-default-shape turtles "circle"

crt 3

ask turtle 0

[

create-links-with other turtles

]

ask turtle 1

[

create-link-with turtle 2

]

repeat 30 [ layout-spring turtles links 0.2 5 1 ] ;; lays the nodes in a triangle

end

**layout-tutte***4.0*

**layout-tutte *turtle-set* *link-set* *radius***

The turtles that are connected by links in *link-set* but not included in *turtle-set* are placed in a circle layout with the given *radius*. There should be at least 3 agents in this agentset.

The turtles in *turtle-set* are then laid out in the following manner: Each turtle is placed at centroid (or barycenter) of the polygon formed by its linked neighbors. (The centroid is like a 2-dimensional average of the coordinates of the neighbors.)

(The purpose of the circle of "anchor agents" is to prevent all the turtles from collapsing down to one point.)

After a few iterations of this, the layout will stabilize.

This layout is named after the mathematician William Thomas Tutte, who proposed it as a method for graph layout.

to make-a-tree

set-default-shape turtles "circle"

crt 6

ask turtle 0 [

create-link-with turtle 1

create-link-with turtle 2

create-link-with turtle 3

]

ask turtle 1 [

create-link-with turtle 4

create-link-with turtle 5

]

; place all the turtles with just one

; neighbor on the perimeter of a circle

; and then place the remaining turtles inside

; this circle, spread between their neighbors.

repeat 10 [ layout-tutte (turtles with [link-neighbors = 1]) links 12 ]

end

**left***1.0***lt***1.0*

**left *number*Turtle Command**

The turtle turns left by *number* degrees. (If *number* is negative, it turns right.)

**length***1.0*

**length *list*length *string***

Reports the number of items in the given list, or the number of characters in the given string.

**let***2.1*

**let *variable* *value***

Creates a new local variable and gives it the given value. A local variable is one that exists only within the enclosing block of commands.

If you want to change the value afterwards, use [set](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#set).

Example:

let prey one-of sheep-here

if prey != nobody

[ ask prey [ die ] ]

**link***4.0*

**link *end1* *end2*<breed> *end1* *end2***

Given the who numbers of the endpoints, reports the link connecting the turtles. If there is no such link reports nobody. To refer to breeded links you must use the singular breed form with the endpoints.

ask link 0 1 [ set color green ]

;; unbreeded link connecting turtle 0 and turtle 1 will turn green

ask directed-link 0 1 [ set color red ]

;; directed link connecting turtle 0 and turtle 1 will turn red

See also [patch-at](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patch-at).

**link-heading***4.0*

**link-headingLink Command**

Reports the heading in degrees (at least 0, less than 360) from end1 to end2 of the link. Throws a runtime error if the endpoints are at the same location.

ask link 0 1 [ print link-heading ]

;; prints [[towards other-end] of end1] of link 0 1

See also [link-length](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#link-length)

**link-length***4.0*

**link-lengthLink Command**

Reports the distance between the endpoints of the link.

ask link 0 1 [ print link-length ]

;; prints [[distance other-end] of end1] of link 0 1

See also [link-heading](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#link-heading)

**link-set***4.0*

**link-set *value*(link-set *value1* *value2* ...)**

Reports an agentset containing all of the links anywhere in any of the inputs. The inputs may be individual links, link agentsets, nobody, or lists (or nested lists) containing any of the above.

link-set self

link-set [my-links] of nodes with [color = red]

See also [turtle-set](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#turtle-set), [patch-set](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patch-set).

**link-shapes***4.0*

**link-shapes**

Reports a list of strings containing all of the link shapes in the model.

New shapes can be created, or imported from other models, in the [Link Shapes Editor](http://ccl.northwestern.edu/netlogo/docs/shapes.html).

show link-shapes

=> ["default"]

**links***4.0*

**links**

Reports the agentset consisting of all links. This is a special agentset that can grow as turtles are added to the world, see [the programming guide for more info](http://ccl.northwestern.edu/netlogo/docs/programming.html#special-agentsets).

show count links

;; prints the number of links

**links-own**

**links-own [*var1* ...]*<link-breeds>*-own [*var1* ...]**

The links-own keyword, like the globals, breed, *<breeds>*-own, turtles-own, and patches-own keywords, can only be used at the beginning of a program, before any function definitions. It defines the variables belonging to each link.

If you specify a breed instead of "links", only links of that breed have the listed variables. (More than one link breed may list the same variable.)

undirected-link-breed [sidewalks sidewalk]

directed-link-breed [streets street]

links-own [traffic] ;; applies to all breeds

sidewalks-own [pedestrians]

streets-own [cars bikes]

**list***1.0*

**list *value1* *value2*(list *value1* ...)**

Reports a list containing the given items. The items can be of any type, produced by any kind of reporter.

show list (random 10) (random 10)

=> [4 9] ;; or similar list

show (list 5)

=> [5]

show (list (random 10) 1 2 3 (random 10))

=> [4 1 2 3 9] ;; or similar list

**ln***1.0*

**ln *number***

Reports the natural logarithm of *number*, that is, the logarithm to the base e (2.71828...).

See also [e](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#num-e), [log](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#log).

**log***1.0*

**log *number* *base***

Reports the logarithm of *number* in base *base*.

show log 64 2

=> 6

See also [ln](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#ln).

**loop***1.0*

**loop [ *commands* ]**

Repeats the commands forever, or until the enclosing procedure exits through use of the [stop](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#stop) or [report](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#report) commands.

to move-to-world-edge ;; turtle procedure

loop [

if not can-move? 1 [ stop ]

fd 1

]

end

In this example, stop exits not just the loop, but the entire procedure.

Note: in many circumstances, it is more appropriate to use a forever button to repeat something indefinitely. See [Buttons](http://ccl.northwestern.edu/netlogo/docs/programming.html#buttons) in the Programming Guide.

**lput***1.0*

**lput *value list***

Adds *value* to the end of a list and reports the new list.

;; suppose mylist is [2 7 10 "Bob"]

set mylist lput 42 mylist

;; mylist now is [2 7 10 "Bob" 42]

**M**

**map***1.3*

**map *reporter* *list*(map *reporter* *list1* ...)**

With a single *list*, the given reporter is run for each item in the list, and a list of the results is collected and reported. *reporter* may be an anonymous reporter or the name of a reporter.

show map round [1.1 2.2 2.7]

=> [1 2 3]

show map [ i -> i \* i ] [1 2 3]

=> [1 4 9]

With multiple lists, the given reporter is run for each group of items from each list. So, it is run once for the first items, once for the second items, and so on. All the lists must be the same length.

Some examples make this clearer:

show (map + [1 2 3] [2 4 6])

=> [3 6 9]

show (map [ [a b c] -> a + b = c ] [1 2 3] [2 4 6] [3 5 9])

=> [true false true]

See also [foreach](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#foreach), [-> (anonymous procedure)](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#arrow).

**max***1.0*

**max *list***

Reports the maximum number value in the list. It ignores other types of items.

show max [xcor] of turtles

;; prints the x coordinate of the turtle which is

;; farthest right in the world

show max list a b

;; prints the larger of the two variables a and b

show max (list a b c)

;; prints the largest of the three variables a, b, and c

**max-n-of***4.0*

**max-n-of *number* *agentset* [*reporter*]**

Reports an agentset containing *number* agents from *agentset* with the highest values of *reporter*. The agentset is built by finding all the agents with the highest value of *reporter*, if there are not *number* agents with that value then agents with the second highest value are found, and so on. At the end, if there is a tie that would make the resulting agentset too large, the tie is broken randomly.

;; assume the world is 11 x 11

show max-n-of 5 patches [pxcor]

;; shows 5 patches with pxcor = max-pxcor

show max-n-of 5 patches with [pycor = 0] [pxcor]

;; shows an agentset containing:

;; (patch 1 0) (patch 2 0) (patch 3 0) (patch 4 0) (patch 5 0)

See also [max-one-of](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#max-one-of), [with-max](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#with-max).

**max-one-of***1.0*

**max-one-of *agentset* [*reporter*]**

Reports the agent in the agentset that has the highest value for the given reporter. If there is a tie this command reports one random agent with the highest value. If you want all such agents, use with-max instead.

show max-one-of patches [count turtles-here]

;; prints the first patch with the most turtles on it

See also [max-n-of](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#max-n-of), [with-max](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#with-max).

**max-pxcor***3.1***max-pycor***3.1*

**max-pxcormax-pycor**

These reporters give the maximum x-coordinate and maximum y-coordinate, (respectively) for patches, which determines the size of the world.

Unlike in older versions of NetLogo the origin does not have to be at the center of the world. However, the maximum x- and y- coordinates must be greater than or equal to zero.

Note: You can set the size of the world only by editing the view -- these are reporters which cannot be set.

crt 100 [ setxy random-float max-pxcor

random-float max-pycor ]

;; distributes 100 turtles randomly in the

;; first quadrant

See also [min-pxcor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#min-pcor), [min-pycor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#min-pcor), [world-width](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#world-dim), and [world-height](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#world-dim)

**mean***1.0*

**mean *list***

Reports the statistical mean of the numeric items in the given list. Errors on non-numeric items. The mean is the average, i.e., the sum of the items divided by the total number of items.

show mean [xcor] of turtles

;; prints the average of all the turtles' x coordinates

**median***1.0*

**median *list***

Reports the statistical median of the numeric items of the given list. Ignores non-numeric items. The median is the item that would be in the middle if all the items were arranged in order. (If two items would be in the middle, the median is the average of the two.)

show median [xcor] of turtles

;; prints the median of all the turtles' x coordinates

**member?***1.0*

**member? *value list*member? *string1 string2*member? *agent agentset***

For a list, reports true if the given value appears in the given list, otherwise reports false.

For a string, reports true or false depending on whether *string1* appears anywhere inside *string2* as a substring.

For an agentset, reports true if the given agent is appears in the given agentset, otherwise reports false.

show member? 2 [1 2 3]

=> true

show member? 4 [1 2 3]

=> false

show member? "bat" "abate"

=> true

show member? turtle 0 turtles

=> true

show member? turtle 0 patches

=> false

See also [position](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#position).

**min***1.0*

**min *list***

Reports the minimum number value in the list. It ignores other types of items.

show min [xcor] of turtles

;; prints the lowest x-coordinate of all the turtles

show min list a b

;; prints the smaller of the two variables a and b

show min (list a b c)

;; prints the smallest of the three variables a, b, and c

**min-n-of***4.0*

**min-n-of *number* *agentset* [*reporter*]**

Reports an agentset containing *number* agents from *agentset* with the lowest values of *reporter*. The agentset is built by finding all the agents with the lowest value of *reporter*, if there are not *number* agents with that value then the agents with the second lowest value are found, and so on. At the end, if there is a tie that would make the resulting agentset too large, the tie is broken randomly.

;; assume the world is 11 x 11

show min-n-of 5 patches [pxcor]

;; shows 5 patches with pxcor = min-pxcor

show min-n-of 5 patches with [pycor = 0] [pxcor]

;; shows an agentset containing:

;; (patch -5 0) (patch -4 0) (patch -3 0) (patch -2 0) (patch -1 0)

See also [min-one-of](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#min-one-of), [with-min](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#with-min).

**min-one-of***1.0*

**min-one-of *agentset* [*reporter*]**

Reports a random agent in the agentset that reports the lowest value for the given reporter. If there is a tie, this command reports one random agent that meets the condition. If you want all such agents use with-min instead.

show min-one-of turtles [xcor + ycor]

;; reports the first turtle with the smallest sum of

;; coordinates

See also [with-min](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#with-min), [min-n-of](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#min-n-of).

**min-pxcor***3.1***min-pycor***3.1*

**min-pxcormin-pycor**

These reporters give the minimum x-coordinate and minimum y-coordinate, (respectively) for patches, which determines the size of the world.

Unlike in older versions of NetLogo the origin does not have to be at the center of the world. However, the minimum x- and y- coordinates must be less than or equal to zero.

Note: You can set the size of the world only by editing the view -- these are reporters which cannot be set.

crt 100 [ setxy random-float min-pxcor

random-float min-pycor ]

;; distributes 100 turtles randomly in the

;; third quadrant

See also [max-pxcor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#max-pcor), [max-pycor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#max-pcor), [world-width](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#world-dim), and [world-height](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#world-dim)

**mod***1.0*

***number1* mod *number2***

Reports *number1* modulo *number2*: that is, the residue of *number1* (mod *number2*). mod is is equivalent to the following NetLogo code:

*number1* - (floor (*number1* / *number2*)) \* *number2*

Note that mod is "infix", that is, it comes between its two inputs.

show 62 mod 5

=> 2

show -8 mod 3

=> 1

See also [remainder](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#remainder). mod and remainder behave the same for positive numbers, but differently for negative numbers.

**modes***2.0*

**modes *list***

Reports a list of the most common item or items in *list*.

The input list may contain any NetLogo values.

If the input is an empty list, reports an empty list.

show modes [1 2 2 3 4]

=> [2]

show modes [1 2 2 3 3 4]

=> [2 3]

show modes [ [1 2 [3]] [1 2 [3]] [2 3 4] ]

=> [[1 2 [3]]]

show modes [pxcor] of turtles

;; shows which columns of patches have the most

;; turtles on them

**mouse-down?***1.0*

**mouse-down?**

Reports true if the mouse button is down, false otherwise.

Note: If the mouse pointer is outside of the current view , mouse-down? will always report false.

**mouse-inside?***3.0*

**mouse-inside?**

Reports true if the mouse pointer is inside the current view, false otherwise.

**mouse-xcor***1.0***mouse-ycor***1.0*

**mouse-xcormouse-ycor**

Reports the x or y coordinate of the mouse in the 2D view. The value is in terms of turtle coordinates, so it might not be an integer. If you want patch coordinates, use round mouse-xcor and round mouse-ycor.

Note: If the mouse is outside of the 2D view, reports the value from the last time it was inside.

;; to make the mouse "draw" in red:

if mouse-down?

[ ask patch mouse-xcor mouse-ycor [ set pcolor red ] ]

**move-to***4.0*

**move-to *agent*Turtle Command**

The turtle sets its x and y coordinates to be the same as the given agent's.

(If that agent is a patch, the effect is to move the turtle to the center of that patch.)

move-to turtle 5

;; turtle moves to same point as turtle 5

move-to one-of patches

;; turtle moves to the center of a random patch

move-to max-one-of turtles [size]

;; turtle moves to same point as biggest turtle

Note that the turtle's heading is unaltered. You may want to use the [face](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#face) command first to orient the turtle in the direction of motion.

See also [setxy](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#setxy).

**my-<breeds>my-links***4.0*

**my-<breeds>my-linksTurtle Command**

Reports an agentset of all links connected to the caller of the corresponding breed, regardless of directedness. Generally, you might consider using [my-out-links](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#my-out-links) instead of this primitive, as it works well for either directed or undirected networks (since it excludes directed, incoming links).

crt 5

ask turtle 0

[

create-links-with other turtles

show my-links ;; prints the agentset containing all links

;; (since all the links we created were with turtle 0 )

]

ask turtle 1

[

show my-links ;; shows an agentset containing the link 0 1

]

end

If you only want the undirected links connected to a node, you can do my-links with [ not is-directed-link? self ].

**my-in-<breeds>my-in-links***4.0*

**my-in-<breeds>my-in-linksTurtle Command**

Reports an agentset of all the directed links coming in from other nodes to the caller as well as all undirected links connected to the caller. You can think of this as "all links that you can use to travel *to* this node".

crt 5

ask turtle 0

[

create-links-to other turtles

show my-in-links ;; shows an empty agentset

]

ask turtle 1

[

show my-in-links ;; shows an agentset containing the link 0 1

]

**my-out-<breeds>my-out-links***4.0*

**my-out-<breeds>my-out-linksTurtle Command**

Reports an agentset of all the directed links going out from the caller to other nodes as well as undirected links connected to the caller. You can think of this as "all links you can use to travel *from* this node".

crt 5

ask turtle 0

[

create-links-to other turtles

show my-out-links ;; shows agentset containing all the links

]

ask turtle 1

[

show my-out-links ;; shows an empty agentset

]

**myself***1.0*

**myselfTurtle Command Patch Command Link Command**

"self" and "myself" are very different. "self" is simple; it means "me". "myself" means "the turtle, patch or link who asked me to do what I'm doing right now."

When an agent has been asked to run some code, using myself in that code reports the agent (turtle, patch or link) that did the asking.

myself is most often used in conjunction with of to read or set variables in the asking agent.

myself can be used within blocks of code not just in the ask command, but also hatch, sprout, of, with, all?, with-min, with-max, min-one-of, max-one-of, min-n-of, max-n-of.

ask turtles

[ ask patches in-radius 3

[ set pcolor [color] of myself ] ]

;; each turtle makes a colored "splotch" around itself

See the "Myself Example" code example for more examples.

See also [self](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#self).

**N**

**n-of***3.1*

**n-of *size* *agentset*n-of *size* *list***

From an agentset, reports an agentset of size *size* randomly chosen from the input set, with no repeats.

From a list, reports a list of size *size* randomly chosen from the input set, with no repeats. The items in the result appear in the same order that they appeared in the input list. (If you want them in random order, use shuffle on the result.)

It is an error for *size* to be greater than the size of the input.

ask n-of 50 patches [ set pcolor green ]

;; 50 randomly chosen patches turn green

See also [one-of](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#one-of).

**n-values***2.0*

**n-values *size* *reporter***

Reports a list of length *size* containing values computed by repeatedly running the reporter. *reporter* may be an anonymous reporter or the name of a reporter.

If the reporter accepts inputs, the input will be the number of the item currently being computed, starting from zero.

show n-values 5 [1]

=> [1 1 1 1 1]

show n-values 5 [ i -> i ]

=> [0 1 2 3 4]

show n-values 3 turtle

=> [(turtle 0) (turtle 1) (turtle 2)]

show n-values 5 [ x -> x \* x ]

=> [0 1 4 9 16]

See also [reduce](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#reduce), [filter](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#filter), [-> (anonymous procedure)](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#arrow), [range](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#range).

**neighbors***1.1***neighbors4***1.1*

**neighborsneighbors4Turtle Command Patch Command**

Reports an agentset containing the 8 surrounding patches (neighbors) or 4 surrounding patches (neighbors4).

show sum [count turtles-here] of neighbors

;; prints the total number of turtles on the eight

;; patches around this turtle or patch

show count turtles-on neighbors

;; a shorter way to say the same thing

ask neighbors4 [ set pcolor red ]

;; turns the four neighboring patches red

**<breed>-neighborslink-neighbors***4.0*

**<breed>-neighborslink-neighborsTurtle Command**

Reports the agentset of all turtles found at the other end of any links (undirected or directed, incoming or outgoing) connected to this turtle.

crt 3

ask turtle 0

[

create-links-with other turtles

ask link-neighbors [ set color red ] ;; turtles 1 and 2 turn red

]

ask turtle 1

[

ask link-neighbors [ set color blue ] ;; turtle 0 turns blue

]

end

**<breed>-neighbor?link-neighbor?***4.0*

**<breed>-neighbor? *turtle*link-neighbor? *turtle*Turtle Command**

Reports true if there is a link (either directed or undirected, incoming or outgoing) between *turtle* and the caller.

crt 2

ask turtle 0

[

create-link-with turtle 1

show link-neighbor? turtle 1 ;; prints true

]

ask turtle 1

[

show link-neighbor? turtle 0 ;; prints true

]

**netlogo-version***3.0*

**netlogo-version**

Reports a string containing the version number of the NetLogo you are running.

show netlogo-version

=> "6.0.4"

**netlogo-web?***5.2*

**netlogo-web?**

Reports true if the model is running in NetLogo Web.

**new-seed***3.0*

**new-seed**

Reports a number suitable for seeding the random number generator.

The numbers reported by new-seed are based on the current date and time in milliseconds and lie in the generator's usable range of seeds, -2147483648 to 2147483647.

new-seed never reports the same number twice in succession, even across parallel BehaviorSpace runs. (This is accomplished by waiting a millisecond if the seed for the current millisecond was already used.)

See also [random-seed](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#random-seed).

**no-display***1.0*

**no-display**

Turns off all updates to the current view until the display command is issued. This has two major uses.

One, you can control when the user sees view updates. You might want to change lots of things on the view behind the user's back, so to speak, then make them visible to the user all at once.

Two, your model will run faster when view updating is off, so if you're in a hurry, this command will let you get results faster. (Note that normally you don't need to use no-display for this, since you can also use the on/off switch in view control strip to freeze the view.)

Note that display and no-display operate independently of the switch in the view control strip that freezes the view.

See also [display](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#display).

**nobody**

**nobody**

This is a special value which some primitives such as turtle, one-of, max-one-of, etc. report to indicate that no agent was found. Also, when a turtle dies, it becomes equal to nobody.

Note: Empty agentsets are not equal to nobody. If you want to test for an empty agentset, use [any?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#any). You only get nobody back in situations where you were expecting a single agent, not a whole agentset.

set target one-of other turtles-here

if target != nobody

[ ask target [ set color red ] ]

**no-links***4.0*

**no-links**

Reports an empty link agentset.

**no-patches***4.0*

**no-patches**

Reports an empty patch agentset.

**not***1.0*

**not *boolean***

Reports true if *boolean* is false, otherwise reports false.

if not any? turtles [ crt 10 ]

**no-turtles***4.0*

**no-turtles**

Reports an empty turtle agentset.

**O**

**of***4.0*

**[*reporter*] of *agent*[*reporter*] of *agentset***

For an agent, reports the value of the reporter for that agent (turtle or patch).

show [pxcor] of patch 3 5

;; prints 3

show [pxcor] of one-of patches

;; prints the value of a random patch's pxcor variable

show [who \* who] of turtle 5

=> 25

show [count turtles in-radius 3] of patch 0 0

;; prints the number of turtles located within a

;; three-patch radius of the origin

For an agentset, reports a list that contains the value of the reporter for each agent in the agentset (in random order).

crt 4

show sort [who] of turtles

=> [0 1 2 3]

show sort [who \* who] of turtles

=> [0 1 4 9]

**one-of***1.0*

**one-of *agentset*one-of *list***

From an agentset, reports a random agent. If the agentset is empty, reports [nobody](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#nobody).

From a list, reports a random list item. It is an error for the list to be empty.

ask one-of patches [ set pcolor green ]

;; a random patch turns green

ask patches with [any? turtles-here]

[ show one-of turtles-here ]

;; for each patch containing turtles, prints one of

;; those turtles

;; suppose mylist is [1 2 3 4 5 6]

show one-of mylist

;; prints a value randomly chosen from the list

See also [n-of](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#n-of).

**or***1.0*

***boolean1* or *boolean2***

Reports true if either *boolean1* or *boolean2*, or both, is true.

Note that if *condition1* is true, then *condition2* will not be run (since it can't affect the result).

if (pxcor > 0) or (pycor > 0) [ set pcolor red ]

;; patches turn red except in lower-left quadrant

**other***4.0*

**other *agentset*Turtle Command Patch Command**

Reports an agentset which is the same as the input agentset but omits this agent.

show count turtles-here

=> 10

show count other turtles-here

=> 9

**other-end***4.0*

**other-endTurtle Command Link Command**

If run by a turtle, reports the turtle at the other end of the asking link.

If run by a link, reports the turtle at the end of the link that isn't the asking turtle.

These definitions are difficult to understand in the abstract, but the following examples should help:

ask turtle 0 [ create-link-with turtle 1 ]

ask turtle 0 [ ask link 0 1 [ show other-end ] ] ;; prints turtle 1

ask turtle 1 [ ask link 0 1 [ show other-end ] ] ;; prints turtle 0

ask link 0 1 [ ask turtle 0 [ show other-end ] ] ;; prints turtle 1

As these examples hopefully make plain, the "other" end is the end that is neither asking nor being asked.

**out-<breed>-neighbor?out-link-neighbor?***4.0*

**out-<breed>-neighbor? *turtle*out-link-neighbor? *turtle*Turtle Command**

Reports true if there is a directed link going from the caller to *turtle* or if there is an undirected link connecting the caller with *turtle*. You can think of this as "can I get from the caller to *turtle* using a link?"

crt 2

ask turtle 0 [

create-link-to turtle 1

show in-link-neighbor? turtle 1 ;; prints false

show out-link-neighbor? turtle 1 ;; prints true

]

ask turtle 1 [

show in-link-neighbor? turtle 0 ;; prints true

show out-link-neighbor? turtle 0 ;; prints false

]

**out-<breed>-neighborsout-link-neighbors***4.0*

**out-<breed>-neighborsout-link-neighborsTurtle Command**

Reports the agentset of all the turtles that have directed links from the caller, or undirected links with the caller. You can think of this as "who can I get to from the caller using a link?"

crt 4

ask turtle 0

[

create-links-to other turtles

ask out-link-neighbors [ set color pink ] ;; turtles 1-3 turn pink

]

ask turtle 1

[

ask out-link-neighbors [ set color orange ] ;; no turtles change colors

;; since turtle 1 only has in-links

]

end

**out-<breed>-toout-link-to***4.0*

**out-<breed>-to *turtle*out-link-to *turtle*Turtle Command**

Reports a directed link from the caller to *turtle* or an undirected link connecting the two. If no link exists then it reports nobody. If more than one such link exists, reports a random one. You can think of this as "give me a link that I can use to travel from the caller to *turtle*."

crt 2

ask turtle 0 [

create-link-to turtle 1

show out-link-to turtle 1 ;; shows link 0 1

]

ask turtle 1

[

show out-link-to turtle 0 ;; shows nobody

]

See also: [in-link-from](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#in-link-from) [link-with](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#link-with)

**output-print***2.1***output-show***2.1***output-type***2.1***output-write***2.1*

**output-print *value*output-show *value*output-type *value*output-write *value***

These commands are the same as the [print](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#print), [show](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#show), [type](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#type), and [write](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#write) commands except that *value* is printed in the model's output area, instead of in the Command Center. (If the model does not have a separate output area, then the Command Center is used.) See also [Output (programming guide)](http://ccl.northwestern.edu/netlogo/docs/programming.html#output).

**P**

**patch***1.0*

**patch *xcor* *ycor***

Given the x and y coordinates of a point, reports the patch containing that point. (The coordinates are absolute coordinates; they are not computed relative to this agent, as with patch-at.)

If x and y are integers, the point is the center of a patch. If x or y is not an integer, rounding to the nearest integer is used to determine which patch contains the point.

If wrapping is allowed by the topology, the given coordinates will be wrapped to be within the world. If wrapping is not allowed and the given coordinates are outside the world, reports nobody.

ask patch 3 -4 [ set pcolor green ]

;; patch with pxcor of 3 and pycor of -4 turns green

show patch 1.2 3.7

;; prints (patch 1 4); note rounding

show patch 18 19

;; supposing min-pxcor and min-pycor are -17

;; and max-pxcor and max-pycor are 17,

;; in a wrapping topology, prints (patch -17 -16);

;; in a non-wrapping topology, prints nobody

See also [patch-at](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patch-at).

**patch-ahead***2.0*

**patch-ahead *distance*Turtle Command**

Reports the single patch that is the given distance "ahead" of this turtle, that is, along the turtle's current heading. Reports nobody if the patch does not exist because it is outside the world.

ask patch-ahead 1 [ set pcolor green ]

;; turns the patch 1 in front of this turtle

;; green; note that this might be the same patch

;; the turtle is standing on

See also [patch-at](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patch-at), [patch-left-and-ahead](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patch-lr-and-ahead), [patch-right-and-ahead](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patch-lr-and-ahead), [patch-at-heading-and-distance](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patch-at-heading-and-distance).

**patch-at***1.0*

**patch-at *dx* *dy*Turtle Command Patch Command**

Reports the patch at (dx, dy) from the caller, that is, the patch containing the point dx east and dy patches north of this agent.

Reports nobody if there is no such patch because that point is beyond a non-wrapping world boundary.

ask patch-at 1 -1 [ set pcolor green ]

;; if caller is a turtle or patch, turns the

;; patch just southeast of the caller green

See also [patch](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patch), [patch-ahead](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patch-ahead), [patch-left-and-ahead](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patch-lr-and-ahead), [patch-right-and-ahead](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patch-lr-and-ahead), [patch-at-heading-and-distance](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patch-at-heading-and-distance).

**patch-at-heading-and-distance***2.0*

**patch-at-heading-and-distance *heading* *distance*Turtle Command Patch Command**

patch-at-heading-and-distance reports the single patch that is the given distance from this turtle or patch, along the given absolute heading. (In contrast to patch-left-and-ahead and patch-right-and-ahead, this turtle's current heading is not taken into account.) Reports nobody if the patch does not exist because it is outside the world.

ask patch-at-heading-and-distance -90 1 [ set pcolor green ]

;; turns the patch 1 to the west of this patch green

See also [patch](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patch), [patch-at](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patch-at), [patch-left-and-ahead](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patch-lr-and-ahead), [patch-right-and-ahead](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patch-lr-and-ahead).

**patch-here***1.0*

**patch-hereTurtle Command**

patch-here reports the patch under the turtle.

Note that this reporter isn't available to a patch because a patch can just say "self".

**patch-left-and-ahead***2.0***patch-right-and-ahead***2.0*

**patch-left-and-ahead *angle* *distance*patch-right-and-ahead *angle* *distance*Turtle Command**

Reports the single patch that is the given distance from this turtle, in the direction turned left or right the given angle (in degrees) from the turtle's current heading. Reports nobody if the patch does not exist because it is outside the world.

(If you want to find a patch in a given absolute heading, rather than one relative to the current turtle's heading, use patch-at-heading-and-distance instead.)

ask patch-right-and-ahead 30 1 [ set pcolor green ]

;; this turtle "looks" 30 degrees right of its

;; current heading at the patch 1 unit away, and turns

;; that patch green; note that this might be the same

;; patch the turtle is standing on

See also [patch](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patch), [patch-at](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patch-at), [patch-at-heading-and-distance](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patch-at-heading-and-distance).

**patch-set***4.0*

**patch-set *value1*(patch-set *value1* *value2* ...)**

Reports an agentset containing all of the patches anywhere in any of the inputs. The inputs may be individual patches, patch agentsets, nobody, or lists (or nested lists) containing any of the above.

patch-set self

patch-set patch-here

(patch-set self neighbors)

(patch-set patch-here neighbors)

(patch-set patch 0 0 patch 1 3 patch 4 -2)

(patch-set patch-at -1 1 patch-at 0 1 patch-at 1 1)

patch-set [patch-here] of turtles

patch-set [neighbors] of turtles

See also [turtle-set](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#turtle-set), [link-set](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#link-set).

**patch-size***4.1*

**patch-size**

Reports the size of the patches in the view in pixels. The size is typically an integer, but may also be a floating point number.

See also [set-patch-size](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#set-patch-size).

**patches***1.0*

**patches**

Reports the agentset consisting of all patches.

**patches-own**

**patches-own [*var1* ...]**

This keyword, like the globals, breed, *<breed>*-own, and turtles-own keywords, can only be used at the beginning of a program, before any function definitions. It defines the variables that all patches can use.

All patches will then have the given variables and be able to use them.

All patch variables can also be directly accessed by any turtle standing on the patch.

See also [globals](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#globals), [turtles-own](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#turtles-own), [breed](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#breed), [*<breeds>*-own](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#turtles-own).

**pcolor**

**pcolorPatch Command Turtle Command**

This is a built-in patch variable. It holds the color of the patch. You can set this variable to make the patch change color.

All patch variables can be directly accessed by any turtle standing on the patch. Color can be represented either as a NetLogo color (a single number) or an RGB color (a list of 3 numbers). See details in the [Colors section](http://ccl.northwestern.edu/netlogo/docs/programming.html#colors) of the Programming Guide.

See also [color](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#color).

**pen-down***1.0***pd***1.0***pen-erase***3.0***pe***3.0***pen-up***1.0***pu***1.0*

**pen-downpen-erasepen-upTurtle Command**

The turtle changes modes between drawing lines, removing lines or neither. The lines will always be displayed on top of the patches and below the turtles. To change the color of the pen set the color of the turtle using set color.

Note: When a turtle's pen is down, all movement commands cause lines to be drawn, including jump, setxy, and move-to.

Note: These commands are equivalent to setting the turtle variable "pen-mode" to "down" , "up", and "erase".

Note: On Windows drawing and erasing a line might not erase every pixel.

**pen-mode**

**Turtle Command**

This is a built-in turtle variable. It holds the state of the turtle's pen. You set the variable to draw lines, erase lines or stop either of these actions. Possible values are "up", "down", and "erase".

**pen-size**

**Turtle Command**

This is a built-in turtle variable. It holds the width of the line, in pixels, that the turtle will draw (or erase) when the pen is down (or erasing).

**plabel**

**plabelPatch Command Turtle Command**

This is a built-in patch variable. It may hold a value of any type. The patch appears in the view with the given value "attached" to it as text. You can set this variable to add, change, or remove a patch's label.

All patch variables can be directly accessed by any turtle standing on the patch.

See also [plabel-color](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#plabel-color), [label](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#label), [label-color](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#label-color).

**plabel-color**

**plabel-colorPatch Command Turtle Command**

This is a built-in patch variable. It holds a number greater than or equal to 0 and less than 140. This number determines what color the patch's label appears in (if it has a label). You can set this variable to change the color of a patch's label.

All patch variables can be directly accessed by any turtle standing on the patch.

See also [plabel](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#plabel), [label](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#label), [label-color](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#label-color).

**plot***1.0*

**plot *number***

Increments the x-value of the plot pen by plot-pen-interval, then plots a point at the updated x-value and a y-value of *number*. (The first time the command is used on a plot, the point plotted has an x-value of 0.)

**plot-name***1.0*

**plot-name**

Reports the name of the current plot (a string)

**plot-pen-exists?***4.0*

**plot-pen-exists? *string***

Reports true if a plot pen with the given name is defined in the current plot. Otherwise reports false.

**plot-pen-down***1.0***plot-pen-up***1.0*

**plot-pen-downplot-pen-up**

Puts down (or up) the current plot-pen, so that it draws (or doesn't). (By default, all pens are down initially.)

**plot-pen-reset***1.0*

**plot-pen-reset**

Clears everything the current plot pen has drawn, moves it to (0,0), and puts it down. If the pen is a permanent pen, the color, mode, and interval are reset to the default values from the plot Edit dialog.

**plotxy***1.0*

**plotxy *number1 number2***

Moves the current plot pen to the point with coordinates (*number1*, *number2*). If the pen is down, a line, bar, or point will be drawn (depending on the pen's mode).

**plot-x-min***1.0***plot-x-max***1.0***plot-y-min***1.0***plot-y-max***1.0*

**plot-x-minplot-x-maxplot-y-minplot-y-max**

Reports the minimum or maximum value on the x or y axis of the current plot.

These values can be set with the commands set-plot-x-range and set-plot-y-range. (Their default values are set from the plot Edit dialog.)

**position***1.0*

**position *item* *list*position *string1* *string2***

On a list, reports the first position of *item* in *list*, or false if it does not appear.

On strings, reports the position of the first appearance *string1* as a substring of *string2*, or false if it does not appear.

Note: The positions are numbered beginning with 0, not with 1.

;; suppose mylist is [2 7 4 7 "Bob"]

show position 7 mylist

=> 1

show position 10 mylist

=> false

show position "in" "string"

=> 3

See also [member?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#member).

**precision***1.0*

**precision *number places***

Reports *number* rounded to *places* decimal places.

If *places* is negative, the rounding takes place to the left of the decimal point.

show precision 1.23456789 3

=> 1.235

show precision 3834 -3

=> 4000

See also [round](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#round), [ceiling](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#ceiling), [floor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#floor).

**print***1.0*

**print *value***

Prints *value* in the Command Center, followed by a carriage return.

This agent is *not* printed before the value, unlike [show](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#show).

See also [show](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#show), [type](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#type), [write](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#write), [output-print](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#output-cmds), and [Output (programming guide)](http://ccl.northwestern.edu/netlogo/docs/programming.html#output).

**pxcorpycor**

**pxcorpycorPatch Command Turtle Command**

These are built-in patch variables. They hold the x and y coordinate of the patch. They are always integers. You cannot set these variables, because patches don't move.

pxcor is greater than or equal to min-pxcor and less than or equal to max-pxcor; similarly for pycor and min-pycor and max-pycor.

All patch variables can be directly accessed by any turtle standing on the patch.

See also [xcor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#xcor), [ycor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#ycor).

**R**

**random***1.0*

**random *number***

If *number* is positive, reports a random integer greater than or equal to 0, but strictly less than *number*.

If *number* is negative, reports a random integer less than or equal to 0, but strictly greater than *number*.

If *number* is zero, the result is always 0 as well.

Note: In versions of NetLogo prior to version 2.0, this primitive reported a floating point number if given a non-integer input. This is no longer the case. If you want a floating point answer, you must now use [random-float](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#random-float)instead.

show random 3

;; prints 0, 1, or 2

show random -3

;; prints 0, -1, or -2

show random 3.5

;; prints 0, 1, 2, or 3

See also [random-float](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#random-float).

**random-float***2.0*

**random-float *number***

If *number* is positive, reports a random floating point number greater than or equal to 0 but strictly less than *number*.

If *number* is negative, reports a random floating point number less than or equal to 0, but strictly greater than *number*.

If *number* is zero, the result is always 0.

show random-float 3

;; prints a number at least 0 but less than 3,

;; for example 2.589444906014774

show random-float 2.5

;; prints a number at least 0 but less than 2.5,

;; for example 1.0897423196760796

**random-exponential***1.2.1***random-gamma***2.0***random-normal***1.2.1***random-poisson***1.2.1*

**random-exponential *mean*random-gamma *alpha lambda*random-normal *mean standard-deviation*random-poisson *mean***

Reports an accordingly distributed random number with the *mean* and, in the case of the normal distribution, the *standard-deviation*. (The standard deviation may not be negative.)

random-exponential reports an exponentially distributed random floating point number. It is equivalent to (- *mean*) \* ln random-float 1.0.

random-gamma reports a gamma-distributed random floating point number as controlled by the floating point alpha and lambda parameters. Both inputs must be greater than zero. (Note: for results with a given mean and variance, use inputs as follows: alpha = mean \* mean / variance; lambda = 1 / (variance / mean).)

random-normal reports a normally distributed random floating point number.

random-poisson reports a Poisson-distributed random integer.

show random-exponential 2

;; prints an exponentially distributed random floating

;; point number with a mean of 2

show random-normal 10.1 5.2

;; prints a normally distributed random floating point

;; number with a mean of 10.1 and a standard deviation

;; of 5.2

show random-poisson 3.4

;; prints a Poisson-distributed random integer with a

;; mean of 3.4

**random-pxcor***3.1***random-pycor***3.1*

**random-pxcorrandom-pycor**

Reports a random integer ranging from min-pxcor (or -y) to max-pxcor (or -y) inclusive.

ask turtles [

;; move each turtle to the center of a random patch

setxy random-pxcor random-pycor

]

See also [random-xcor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#random-cor), [random-ycor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#random-cor).

**random-seed***1.0*

**random-seed *number***

Sets the seed of the pseudo-random number generator to the integer part of *number*. The seed must be in the range -2147483648 to 2147483647; note that this is smaller than the full range of integers supported by NetLogo (-9007199254740992 to 9007199254740992).

See the [Random Numbers](http://ccl.northwestern.edu/netlogo/docs/programming.html#random-numbers) section of the Programming Guide for more details.

random-seed 47822

show random 100

=> 50

show random 100

=> 35

random-seed 47822

show random 100

=> 50

show random 100

=> 35

**random-xcor***3.1***random-ycor***3.1*

**random-xcorrandom-ycor**

Reports a random floating point number from the allowable range of turtle coordinates along the given axis, x or y.

Turtle coordinates range from min-pxcor - 0.5 (inclusive) to max-pxcor + 0.5 (exclusive) horizontally; vertically, substitute -y for -x.

ask turtles [

;; move each turtle to a random point

setxy random-xcor random-ycor

]

See also [random-pxcor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#random-pcor), [random-pycor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#random-pcor).

**range***6.0*

**range *stop*(range *start* *stop*)(range *start* *stop* *step*)**

Generates a list of numbers, starting at *start*, ending before *stop*, counting by *step*. *start* defaults to 0 and *step* defaults to 1.

show range 5

=> [0 1 2 3 4]

show (range 2 5)

=> [2 3 4]

show (range 2 5 0.5)

=> [2 2.5 3 3.5 4 4.5]

show (range 10 0 -1)

=> [10 9 8 7 6 5 4 3 2 1]

See also [n-values](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#n-values)

**read-from-string***1.1*

**read-from-string *string***

Interprets the given string as if it had been typed in the Command Center, and reports the resulting value. The result may be a number, list, string, or boolean value, or the special value "nobody".

Useful in conjunction with the [user-input](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#user-input) primitive for converting the user's input into usable form.

show read-from-string "3" + read-from-string "5"

=> 8

show length read-from-string "[1 2 3]"

=> 3

crt read-from-string user-input "Make how many turtles?"

;; the number of turtles input by the user

;; are created

**reduce***1.3*

**reduce *reporter* *list***

Reduces a list from left to right using the given reporter, resulting in a single value. This means, for example, that reduce [ [a b] -> a + b] [1 2 3 4] is equivalent to *(((1 + 2) + 3) + 4)*. If *list* has a single item, that item is reported. It is an error to reduce an empty list. *reporter* may be an anonymous reporter or the name of a reporter.

The first input passed to the reporter is the result so far, and the second input is the next item in the list.

Since it can be difficult to develop an intuition about what reduce does, here are some simple examples which, while not useful in themselves, may give you a better understanding of this primitive:

show reduce + [1 2 3]

=> 6

show reduce - [1 2 3]

=> -4

show reduce [ [result-so-far next-item] -> next-item - result-so-far ] [1 2 3]

=> 2

show reduce [ [result-so-far ignored-item] -> result-so-far ] [1 2 3]

=> 1

show reduce [ [ignored next-item] -> next-item ] [1 2 3]

=> 3

show reduce sentence [[1 2] [3 [4]] 5]

=> [1 2 3 [4] 5]

show reduce [ [result-so-far next-item] -> fput next-item result-so-far ] (fput [] [1 2 3 4 5])

=> [5 4 3 2 1]

Here are some more useful examples:

;; find the longest string in a list

to-report longest-string [strings]

report reduce

[ [longest-so-far next-string] -> ifelse-value (length longest-so-far >= length next-string) [longest-so-far] [next-string] ]

strings

end

show longest-string ["hi" "there" "!"]

=> "there"

;; count the number of occurrences of an item in a list

to-report occurrences [x the-list]

report reduce

[ [occurrence-count next-item] -> ifelse-value (next-item = x) [occurrence-count + 1] [occurrence-count] ] (fput 0 the-list)

end

show occurrences 1 [1 2 1 3 1 2 3 1 1 4 5 1]

=> 6

;; evaluate the polynomial, with given coefficients, at x

to-report evaluate-polynomial [coefficients x]

report reduce [ [value coefficient] -> (x \* value) + coefficient ] coefficients

end

;; evaluate 3x^2 + 2x + 1 at x = 4

show evaluate-polynomial [3 2 1] 4

=> 57

See also [filter](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#filter), [-> (anonymous procedure](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#arrow).

**remainder***1.2.1*

**remainder *number1* *number2***

Reports the remainder when *number1* is divided by *number2*. This is equivalent to the following NetLogo code:

*number1* - (int (*number1* / *number2*)) \* *number2*

show remainder 62 5

=> 2

show remainder -8 3

=> -2

See also [mod](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#mod). mod and remainder behave the same for positive numbers, but differently for negative numbers.

**remove***1.0*

**remove *item* *list*remove *string1* *string2***

For a list, reports a copy of *list* with all instances of *item* removed.

For strings, reports a copy of *string2* with all the appearances of *string1* as a substring removed.

set mylist [2 7 4 7 "Bob"]

set mylist remove 7 mylist

;; mylist is now [2 4 "Bob"]

show remove "to" "phototonic"

=> "phonic"

**remove-duplicates***1.0*

**remove-duplicates *list***

Reports a copy of *list* with all duplicate items removed. The first of each item remains in place.

set mylist [2 7 4 7 "Bob" 7]

set mylist remove-duplicates mylist

;; mylist is now [2 7 4 "Bob"]

**remove-item***2.0*

**remove-item *index* *list*remove-item *index* *string***

For a list, reports a copy of *list* with the item at the given index removed.

For strings, reports a copy of *string* with the character at the given index removed.

Note that the indices begin from 0, not 1. (The first item is item 0, the second item is item 1, and so on.)

set mylist [2 7 4 7 "Bob"]

set mylist remove-item 2 mylist

;; mylist is now [2 7 7 "Bob"]

show remove-item 2 "string"

=> "sting"

**repeat***1.0*

**repeat *number* [ *commands* ]**

Runs *commands* *number* times.

pd repeat 36 [ fd 1 rt 10 ]

;; the turtle draws a circle

**replace-item***1.0*

**replace-item *index list value*replace-item *index string1 string2***

On a list, replaces an item in that list. *index* is the index of the item to be replaced, starting with 0. (The 6th item in a list would have an index of 5.) Note that "replace-item" is used in conjunction with "set" to change a list.

Likewise for a string, but the given character of *string1* removed and the contents of *string2* spliced in instead.

show replace-item 2 [2 7 4 5] 15

=> [2 7 15 5]

show replace-item 1 "cat" "are"

=> "caret"

**report***1.0*

**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 [to-report](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#to-report) for a discussion of how to use them.

**reset-perspective***3.0***rp***3.0*

**reset-perspective**

The observer stops watching, following, or riding any turtles (or patches). (If it wasn't watching, following, or riding anybody, nothing happens.) In the 3D view, the observer also returns to its default position (above the origin, looking straight down).

See also [follow](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#follow), [ride](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#ride), [watch](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#watch).

**reset-ticks***4.0*

**reset-ticksObserver Command**

Resets the tick counter to zero, sets up all plots, then updates all plots (so that the initial state of the world is plotted).

Normally reset-ticks goes at the end of a setup procedure.

See also [clear-ticks](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#clear-ticks), [tick](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#tick), [ticks](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#ticks), [tick-advance](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#tick-advance), [setup-plots](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#setup-plots), [update-plots](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#update-plots).

**reset-timer***1.0*

**reset-timer**

Resets the timer to zero seconds. See also [timer](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#timer).

Note that the timer is different from the tick counter. The timer measures elapsed real time in seconds; the tick counter measures elapsed model time in ticks.

**resize-world***4.1*

**resize-world *min-pxcor* *max-pxcor* *min-pycor* *max-pycor*Observer Command**

Changes the size of the patch grid.

If the given patch grid coordinates are different than the ones in use, all turtles and links die, and the existing patch grid is discarded and new patches created. Otherwise, existing turtles and links will live if the grid coordinates are unchanged.

Retaining references to old patches or patch sets is inadvisable and may subsequently cause runtime errors or other unexpected behavior.

See also [set-patch-size](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#set-patch-size).

**reverse***1.0*

**reverse *list*reverse *string***

Reports a reversed copy of the given list or string.

show mylist

;; mylist is [2 7 4 "Bob"]

set mylist reverse mylist

;; mylist now is ["Bob" 4 7 2]

show reverse "live"

=> "evil"

**rgb***1.0*

**rgb *red green blue***

Reports a RGB list when given three numbers describing an RGB color. The numbers are range checked to be between 0 and 255.

See also [hsb](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hsb)

**ride***3.0*

**ride *turtle*Observer Command**

Set the perspective to *turtle*.

Every time *turtle* moves the observer also moves. Thus, in the 2D View the turtle will stay at the center of the view. In the 3D view it is as if looking through the eyes of the turtle. If the turtle dies, the perspective resets to the default.

The observer may only watch or follow a single subject. Calling ride will remove the highlight created by prior calls to watch and watch-me, highlighting the ridden turtle instead.

See also [reset-perspective](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#reset-perspective), [watch](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#watch), [follow](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#follow), [subject](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#subject).

**ride-me***3.0*

**ride-meTurtle Command**

Asks the observer to ride this turtle.

The observer may only watch or follow a single subject. Calling ride-me will remove the highlight created by prior calls to watch and watch-me, highlighting this turtle instead.

See also [ride](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#ride).

**right***1.0***rt***1.0*

**right *number*Turtle Command**

The turtle turns right by *number* degrees. (If *number* is negative, it turns left.)

**round***1.0*

**round *number***

Reports the integer nearest to *number*.

If the decimal portion of *number* is exactly .5, the number is rounded in the **positive** direction.

Note that rounding in the positive direction is not always how rounding is done in other software programs. (In particular, it does not match the behavior of StarLogoT, which always rounded numbers ending in 0.5 to the nearest even integer.) The rationale for this behavior is that it matches how turtle coordinates relate to patch coordinates in NetLogo. For example, if a turtle's xcor is -4.5, then it is on the boundary between a patch whose pxcor is -4 and a patch whose pxcor is -5, but the turtle must be considered to be in one patch or the other, so the turtle is considered to be in the patch whose pxcor is -4, because we round towards the positive numbers.

show round 4.2

=> 4

show round 4.5

=> 5

show round -4.5

=> -4

See also [precision](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#precision), [ceiling](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#ceiling), [floor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#floor).

**run***1.3***runresult***1.3*

**run *command*(run *command* *input1* ...)run *string*runresult *reporter*(runresult *reporter* *input1* ...)runresult *string***

The run form expects the name of a command, an anonymous command, or a string containing commands. This agent then runs them.

The runresult form expects the name of a reporter, an anonymous reporter, or a string containing a reporter. This agent runs it and reports the result.

Note that you can't use run to define or redefine procedures. If you care about performance, note that the code must be compiled first which takes time. However, compiled bits of code are cached by NetLogo and thus using run on the same string over and over is much faster than running different strings. The first run, though, will be many times slower than running the same code directly, or in an anonymous command.

Anonymous procedures are recommended over strings whenever possible. (An example of when you must use strings is if you accept pieces of code from the user of your model.)

Anonymous procedures may freely read and/or set local variables and procedure inputs. Trying to do the same with strings may or may not work and should not be relied on.

When using anonymous procedures, you can provide them with inputs, if you surround the entire call with parentheses. For example:

(run [ [turtle-count step-count] -> crt turtle-count [ fd step-count ] ] 10 5)

;; creates 10 turtles and move them forward 5 steps

show (runresult [ [a b] -> a + b ] 10 5)

=> 15

;; adds 10 and 5

See also [foreach](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#foreach), [-> (anonymous procedure)](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#arrow).

**S**

**scale-color***1.0*

**scale-color *color number range1 range2***

Reports a shade of *color* proportional to the value of *number*.

If *range1* is less than *range2*, then the larger the number, the lighter the shade of *color*. But if *range2* is less than *range1*, the color scaling is inverted.

If *number* is less than *range1*, then the darkest shade of *color* is chosen.

If *number* is greater than *range2*, then the lightest shade of *color* is chosen.

Note: for *color* shade is irrelevant, e.g. green and green + 2 are equivalent, and the same spectrum of colors will be used.

ask turtles [ set color scale-color red age 0 50 ]

;; colors each turtle a shade of red proportional

;; to its value for the age variable

**self***1.3*

**selfTurtle Command Patch Command Link Command**

Reports this turtle, patch, or link.

"self" and "myself" are very different. "self" is simple; it means "me". "myself" means "the agent who asked me to do what I'm doing right now."

Note that it is always redundant to write [foo] of self. This is always equivalent to simply writing foo.

See also [myself](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#myself).

**; (semicolon)**

**; *comments***

After a semicolon, the rest of the line is ignored. This is useful for adding "comments" to your code -- text that explains the code to human readers. Extra semicolons can be added for visual effect.

NetLogo's Edit menu has items that let you comment or uncomment whole sections of code.

**sentence***1.0***se***1.0*

**sentence *value1* *value2*(sentence *value1* ...)**

Makes a list out of the values. If any value is a list, its items are included in the result directly, rather than being included as a sublist. Examples make this clearer:

show sentence 1 2

=> [1 2]

show sentence [1 2] 3

=> [1 2 3]

show sentence 1 [2 3]

=> [1 2 3]

show sentence [1 2] [3 4]

=> [1 2 3 4]

show sentence [[1 2]] [[3 4]]

=> [[1 2] [3 4]]

show (sentence [1 2] 3 [4 5] (3 + 3) 7)

=> [1 2 3 4 5 6 7]

**set***1.0*

**set *variable* *value***

Sets *variable* to the given value.

Variable can be any of the following:

* A global variable declared using "globals"
* The global variable associated with a slider, switch, chooser, or input box.
* A variable belonging to this agent
* If this agent is a turtle, a variable belonging to the patch under the turtle.
* A local variable created by the [let](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#let) command.
* An input to the current procedure.

**set-current-directory***2.0*

**set-current-directory *string***

Sets the current directory that is used by the primitives [file-delete](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-delete), [file-exists?](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-exists), and [file-open](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#file-open).

The current directory is not used if the above commands are given an absolute file path. This is defaulted to the user's home directory for new models, and is changed to the model's directory when a model is opened.

Note that in Windows file paths the backslash needs to be escaped within a string by using another backslash "C:\\"

The change is temporary and is not saved with the model.

set-current-directory "C:\\NetLogo"

;; Assume it is a Windows Machine

file-open "my-file.txt"

;; Opens file "C:\\NetLogo\\my-file.txt"

**set-current-plot***1.0*

**set-current-plot *plotname***

Sets the current plot to the plot with the given name (a string). Subsequent plotting commands will affect the current plot.

**set-current-plot-pen***1.0*

**set-current-plot-pen *penname***

The current plot's current pen is set to the pen named *penname* (a string). If no such pen exists in the current plot, a runtime error occurs.

**set-default-shape***1.0*

**set-default-shape turtles *string*set-default-shape links *string*set-default-shape *breed* *string*Observer Command**

Specifies a default initial shape for all turtles or links, or for a particular breed of turtles or links. When a turtle or link 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 either turtles, links, or the name of a breed. The given string must be the name of a currently defined shape.

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.

create-turtles 1 ;; new turtle's shape is "default"

create-cats 1 ;; new turtle's shape is "default"

set-default-shape turtles "circle"

create-turtles 1 ;; new turtle's shape is "circle"

create-cats 1 ;; new turtle's shape is "circle"

set-default-shape cats "cat"

set-default-shape dogs "dog"

create-cats 1 ;; new turtle's shape is "cat"

ask cats [ set breed dogs ]

;; all cats become dogs, and automatically

;; change their shape to "dog"

See also [shape](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#shape).

**set-histogram-num-bars***1.0*

**set-histogram-num-bars *number***

Set the current plot pen's plot interval so that, given the current x range for the plot, there would be *number* number of bars drawn if the histogram command is called.

See also [histogram](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#histogram).

**\_\_set-line-thickness**

**\_\_set-line-thickness *number*Turtle Command**

Specifies the thickness of lines and outlined elements in the turtle's shape.

The default value is 0. This always produces lines one pixel thick.

Non-zero values are interpreted as thickness in patches. A thickness of 1, for example, produces lines which appear one patch thick. (It's common to use a smaller value such as 0.5 or 0.2.)

Lines are always at least one pixel thick.

This command is experimental and may change in later releases.

**set-patch-size***4.1*

**set-patch-size *size*Observer Command**

Sets the size of the patches of the view in pixels. The size is typically an integer, but may also be a floating point number.

See also [patch-size](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patch-size), [resize-world](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#resize-world).

**set-plot-background-color***6.0.2*

**set-plot-background-color *color***

Sets the background color of the current plot. The color may be specified as a number or a list. See the [Colors](http://ccl.northwestern.edu/netlogo/docs/programming.html#colors) section of the programming guide for more details. This change is temporary and is not saved with the model. When the plot is cleared, the background color will revert to white.

**Note:** Plot backgrounds do not support transparency. If a list is used to set the color, the alpha component will be ignored.

**set-plot-pen-color***1.0*

**set-plot-pen-color *color***

Sets the color of the current plot pen to *color*.

**set-plot-pen-interval***1.0*

**set-plot-pen-interval *number***

Tells the current plot pen to move a distance of *number* in the x direction during each use of the plot command. (The plot pen interval also affects the behavior of the histogram command.)

**set-plot-pen-mode***1.0*

**set-plot-pen-mode *number***

Sets the mode the current plot pen draws in to *number*. The allowed plot pen modes are:

* 0 (line mode) the plot pen draws a line connecting two points together.
* 1 (bar mode): the plot pen draws a bar of width plot-pen-interval with the point plotted as the upper (or lower, if you are plotting a negative number) left corner of the bar.
* 2 (point mode): the plot pen draws a point at the point plotted. Points are not connected.

The default mode for new pens is 0 (line mode).

**setup-plots***5.0*

**setup-plots**

For each plot, runs that plot's setup commands, including the setup code for any pens in the plot.

[reset-ticks](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#reset-ticks) has the same effect, so in models that use the tick counter, this primitive is not normally used.

See the [Plotting section](http://ccl.northwestern.edu/netlogo/docs/programming.html#plotting) of the Programming Guide for more details.

See also [update-plots](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#update-plots).

**set-plot-x-range***1.0***set-plot-y-range***1.0*

**set-plot-x-range *min max*set-plot-y-range *min max***

Sets the minimum and maximum values of the x or y axis of the current plot.

The change is temporary and is not saved with the model. When the plot is cleared, the ranges will revert to their default values as set in the plot's Edit dialog.

**setxy***1.0*

**setxy *x y*Turtle Command**

The turtle sets its x-coordinate to *x* and its y-coordinate to *y*.

Equivalent to set xcor x set ycor y, except it happens in one time step instead of two.

If *x* or *y* is outside the world, NetLogo will throw a runtime error, unless wrapping is turned on in the relevant dimensions. For example, with wrapping turned on in both dimensions and the default world size where min-pxcor = -16, max-pxcor = 16, min-pycor = -16 and max-pycor = 16, asking a turtle to setxy 17 17 will move it to the center of patch (-16, -16).

setxy 0 0

;; turtle moves to the middle of the center patch

setxy random-xcor random-ycor

;; turtle moves to a random point

setxy random-pxcor random-pycor

;; turtle moves to the center of a random patch

See also [move-to](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#move-to).

**shade-of?***1.0*

**shade-of? *color1* *color2***

Reports true if both colors are shades of one another, false otherwise.

show shade-of? blue red

=> false

show shade-of? blue (blue + 1)

=> true

show shade-of? gray white

=> true

**shape**

**shapeTurtle Command Link Command**

This is a built-in turtle and link variable. It holds a string that is the name of the turtle or link's current shape. You can set this variable to change the shape. New turtles and links have the shape "default" unless the a different shape has been specified using [set-default-shape](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#set-default-shape).

Example:

ask turtles [ set shape "wolf" ]

;; assumes you have made a "wolf"

;; shape in NetLogo's [Turtle Shapes Editor](http://ccl.northwestern.edu/netlogo/docs/shapes.html)

ask links [ set shape "link 1" ]

;; assumes you have made a "link 1" shape in

;; the Link Shapes Editor

See also [set-default-shape](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#set-default-shape), [shapes](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#shapes).

**shapes***2.1*

**shapes**

Reports a list of strings containing all of the turtle shapes in the model.

New shapes can be created, or imported from the shapes library or from other models, in the [Shapes Editor](http://ccl.northwestern.edu/netlogo/docs/shapes.html).

show shapes

=> ["default" "airplane" "arrow" "box" "bug" ...

ask turtles [ set shape one-of shapes ]

**show***1.0*

**show *value***

Prints *value* in the Command Center, preceded by this agent, and followed by a carriage return. (This agent is included to help you keep track of what agents are producing which lines of output.) Also, all strings have their quotes included similar to [write](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#write).

See also [print](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#print), [type](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#type), [write](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#write), [output-show](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#output-cmds), and [Output (programming guide)](http://ccl.northwestern.edu/netlogo/docs/programming.html#output).

**show-turtle***1.0***st***1.0*

**show-turtleTurtle Command**

The turtle becomes visible again.

Note: This command is equivalent to setting the turtle variable "hidden?" to false.

See also [hide-turtle](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hide-turtle).

**show-link***4.0*

**show-linkLink Command**

The link becomes visible again.

Note: This command is equivalent to setting the link variable "hidden?" to false.

See also [hide-link](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hide-link).

**shuffle***2.0*

**shuffle *list***

Reports a new list containing the same items as the input list, but in randomized order.

show shuffle [1 2 3 4 5]

=> [5 2 4 1 3]

show shuffle [1 2 3 4 5]

=> [1 3 5 2 4]

**sin***1.0*

**sin *number***

Reports the sine of the given angle. Assumes angle is given in degrees.

show sin 270

=> -1

**size**

**sizeTurtle Command**

This is a built-in turtle variable. It holds a number that is the turtle's apparent size. The default size is 1, which means that the turtle is the same size as a patch. You can set this variable to change a turtle's size.

**sort***1.0*

**sort *list*sort *agentset***

Reports a sorted list of numbers, strings, or agents.

If the input contains no numbers, strings, or agents, the result is the empty list.

If the input contains at least one number, the numbers in the list are sorted in ascending order and a new list reported; non-numbers are ignored.

Or, if the input contains at least one string, the strings in the list are sorted in ascending order and a new list reported; non-strings are ignored.

Or, if the input is an agentset or a list containing at least one agent, a sorted list of agents (never an agentset) is reported; non-agents are ignored. Agents are sorted in the same order the < operator uses. (Patches are sorted with the top left-most patch first and the bottom right-most patch last, turtles are sorted by who number).

show sort [3 1 4 2]

=> [1 2 3 4]

show sort [2 1 "a"]

=> [1 2]

show sort (list "a" "c" "b" (patch 0 0))

=> ["a" "b" "c"]

show sort (list (patch 0 0) (patch 0 1) (patch 1 0))

=> [(patch 0 1) (patch 0 0) (patch 1 0)]

;; label patches with numbers in left-to-right, top-to-bottom order

let n 0

foreach sort patches [ the-patch ->

ask the-patch [

set plabel n

set n n + 1

]

]

;; some additional examples to clarify behavior in strange cases

show sort (list patch 0 0 patch 0 1 patch 1 0 turtle 0 turtle 1) ; turtles are always sorted lower than patches

=> [(turtle 0) (turtle 1) (patch 0 1) (patch 0 0) (patch 1 0)]

show sort (list nobody false true) ; booleans and nobody cannot be sorted

=> []

show sort (list [1 2 3] turtles) ; lists and agentsets are not included if they are inside a list passed to sort

=> []

See also [sort-by](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#sort-by), [sort-on](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#sort-on).

**sort-by***1.3*

**sort-by *reporter* *list*sort-by *reporter* *agentset***

If the input is a list, reports a new list containing the same items as the input list, in a sorted order defined by the boolean reporter. *reporter* may be an anonymous reporter or the name of a reporter.

The two inputs to *reporter* are the values being compared. The reporter should report true if the first argument comes strictly before the second in the desired sort order, and false otherwise.

If the input is an agentset or a list of agents, reports a list (never an agentset) of agents.

If the input is a list, the sort is stable, that is, the order of items considered equal by the reporter is not disturbed. If the input is an agentset, ties are broken randomly.

show sort-by < [3 1 4 2]

=> [1 2 3 4]

show sort-by > [3 1 4 2]

=> [4 3 2 1]

show sort-by [ [string1 string2] -> length string1 < length string2 ] ["Grumpy" "Doc" "Happy"]

=> ["Doc" "Happy" "Grumpy"]

See also [sort](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#sort), [sort-on](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#sort-on), [-> (anonymous procedure)](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#arrow).

**sort-on***5.0*

**sort-on [*reporter*] *agentset***

Reports a list of agents, sorted according to each agent's value for *reporter*. Ties are broken randomly.

The values must be all numbers, all strings, or all agents of the same type.

crt 3

show sort-on [who] turtles

=> [(turtle 0) (turtle 1) (turtle 2)]

show sort-on [(- who)] turtles

=> [(turtle 2) (turtle 1) (turtle 0)]

foreach sort-on [size] turtles

[ the-turtle -> ask the-turtle [ do-something ] ]

;; turtles run "do-something" one at a time, in

;; ascending order by size

See also [sort](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#sort), [sort-by](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#sort-by).

**sprout***1.0*

**sprout *number* [ *commands* ]sprout-*<breeds>* *number* [ *commands* ]Patch Command**

Creates *number* new turtles on the current 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.

sprout 5

sprout-wolves 10

sprout 1 [ set color red ]

sprout-sheep 1 [ set color black ]

See also [create-turtles](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#create-turtles), [hatch](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#hatch).

**sqrt***1.0*

**sqrt *number***

Reports the square root of *number*.

**stamp***1.0*

**stampTurtle Command Link Command**

This turtle or link leaves an image of its shape in the drawing at its current location.

Note: The shapes made by stamp may not be pixel-for-pixel identical from computer to computer.

**stamp-erase***3.1*

**stamp-eraseTurtle Command Link Command**

This turtle or link removes any pixels below it in the drawing inside the bounds of its shape.

Note: The shapes made by stamp-erase may not be pixel-for-pixel identical from computer to computer.

**standard-deviation***1.0*

**standard-deviation *list***

Reports the sample standard deviation of a *list* of numbers. Ignores other types of items.

(Note that this estimates the standard deviation for a *sample*, rather than for a whole *population*, using Bessel's correction.)

show standard-deviation [1 2 3 4 5 6]

=> 1.8708286933869707

show standard-deviation [energy] of turtles

;; prints the standard deviation of the variable "energy"

;; from all the turtles

**startup**

**startupObserver Command**

User-defined procedure which, if it exists, will be called when a model is first loaded in the NetLogo application.

to startup

setup

end

startup does not run when a model is run headless from the command line, or by parallel BehaviorSpace.

**stop***1.0*

**stop**

This agent exits immediately from the enclosing procedure, ask, or ask-like construct (e.g. crt, hatch, sprout). Only the enclosing procedure or construct stops, not all execution for the agent.

if not any? turtles [ stop ]

;; exits if there are no more turtles

Note: stop can also be used to stop a forever button. See [Buttons](http://ccl.northwestern.edu/netlogo/docs/programming.html#buttons) in the Programming Guide for details.

stop can also be used to stop a BehaviorSpace model run. If the go commands directly call a procedure, then when that procedure calls *stop*, the run ends.

**stop-inspecting***5.2*

**stop-inspecting *agent***

Closes the agent monitor for the given agent (turtle or patch). In the case that no agent monitor is open, stop-inspecting does nothing.

stop-inspecting patch 2 4

;; the agent monitor for that patch closes

ask sheep [ stop-inspecting self ]

;; close all agent monitors for sheep

See [inspect](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#inspect) and [stop-inspecting-dead-agents](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#stop-inspecting-dead-agents).

**stop-inspecting-dead-agents***5.2*

**stop-inspecting-dead-agents**

Closes all agent monitors for dead agents. See [inspect](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#inspect) and [stop-inspecting](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#stop-inspecting).

**subject***3.0*

**subject**

Reports the turtle (or patch) that the observer is currently watching, following, or riding. Reports [nobody](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#nobody) if there is no such turtle (or patch).

See also [watch](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#watch), [follow](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#follow), [ride](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#ride).

**sublist***2.1***substring***1.0*

**sublist *list position1 position2*substring *string position1 position2***

Reports just a section of the given list or string, ranging between the first position (inclusive) and the second position (exclusive).

Note: The positions are numbered beginning with 0, not with 1.

show sublist [99 88 77 66] 1 3

=> [88 77]

show substring "apartment" 1 5

=> "part"

**subtract-headings***2.1*

**subtract-headings *heading1 heading2***

Computes the difference between the given headings, that is, the number of degrees in the smallest angle by which heading2 could be rotated to produce heading1. A positive answer means a clockwise rotation, a negative answer counterclockwise. The result is always in the range -180 to 180, but is never exactly -180.

Note that simply subtracting the two headings using the - (minus) operator wouldn't work. Just subtracting corresponds to always rotating clockwise from heading2 to heading1; but sometimes the counterclockwise rotation is shorter. For example, the difference between 5 degrees and 355 degrees is 10 degrees, not -350 degrees.

show subtract-headings 80 60

=> 20

show subtract-headings 60 80

=> -20

show subtract-headings 5 355

=> 10

show subtract-headings 355 5

=> -10

show subtract-headings 180 0

=> 180

show subtract-headings 0 180

=> 180

**sum***1.0*

**sum *list***

Reports the sum of the items in the list.

show sum [energy] of turtles

;; prints the total of the variable "energy"

;; from all the turtles

**T**

**tan***1.0*

**tan *number***

Reports the tangent of the given angle. Assumes the angle is given in degrees.

**thickness**

**thicknessLink Command**

This is a built-in link variable. It holds a number that is the link's apparent size as a fraction of the patch size. The default thickness is 0, which means that regardless of patch-size the links will always appear 1 pixel wide. You can set this variable to change a link's thickness.

**tick***4.0*

**tickObserver Command**

Advances the tick counter by one and updates all plots.

If the tick counter has not been started yet with reset-ticks, an error results.

Normally tick goes at the end of a go procedure.

See also [ticks](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#ticks), [tick-advance](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#tick-advance), [reset-ticks](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#reset-ticks), [clear-ticks](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#clear-ticks), [update-plots](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#update-plots).

**tick-advance***4.0*

**tick-advance *number*Observer Command**

Advances the tick counter by *number*. The input may be an integer or a floating point number. (Some models divide ticks more finely than by ones.) The input may not be negative.

When using [tick-based view updates](http://ccl.northwestern.edu/netlogo/docs/programming.html#view-updates), the view is normally updated every 1.0 ticks, so using tick-advance with a number less then 1.0 may not always trigger an update. If you want to make sure that the view is updated, you can use the display command.

If the tick counter has not been started yet with reset-ticks, an error results.

Does not update plots.

See also [tick](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#tick), [ticks](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#ticks), [reset-ticks](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#reset-ticks), [clear-ticks](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#clear-ticks).

**ticks***4.0*

**ticks**

Reports the current value of the tick counter. The result is always a number and never negative.

If the tick counter has not been started yet with reset-ticks, an error results.

Most models use the tick command to advance the tick counter, in which case ticks will always report an integer. If the tick-advance command is used, then ticks may report a floating point number.

See also [tick](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#tick), [tick-advance](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#tick-advance), [reset-ticks](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#reset-ticks), [clear-ticks](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#clear-ticks).

**tie***4.0*

**tieLink Command**

Ties *end1* and *end2* of the link together. If the link is a directed link *end1* is the *root turtle* and *end2* is the *leaf turtle*. The movement of the *root turtle* affects the location and heading of the *leaf turtle*. If the link is undirected the tie is reciprocal so both turtles can be considered *root turtles* and *leaf turtles*. Movement or change in heading of either turtle affects the location and heading of the other turtle.

When the root turtle moves, the leaf turtles moves the same distance, in the same direction. The heading of the leaf turtle is not affected. This works with forward, jump, and setting the xcor or ycor of the root turtle.

When the root turtle turns right or left, the leaf turtle is rotated around the root turtle the same amount. The heading of the leaf turtle is also changed by the same amount.

If the link dies, the tie relation is removed.

crt 2 [ fd 3 ]

;; creates a link and ties turtle 1 to turtle 0

ask turtle 0 [ create-link-to turtle 1 [ tie ] ]

See also [untie](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#untie)

**tie-mode**

**tie-modeLink Command**

This is a built-in link variable. It holds a string that is the name of the tie mode the link is currently in. Using the [tie](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#tie) and [untie](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#untie) commands changes the mode of the link. You can also set tie-mode to "free" to create a non-rigid joint between two turtles (see the [Tie section](http://ccl.northwestern.edu/netlogo/docs/programming.html#tie) of the Programming Guide for details). By default links are not tied.

See also: [tie](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#tie), [untie](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#untie)

**timer***1.0*

**timer**

Reports how many seconds have passed since the command [reset-timer](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#reset-timer) was last run (or since NetLogo started). The potential resolution of the clock is milliseconds. (Whether you get resolution that high in practice may vary from system to system, depending on the capabilities of the underlying Java Virtual Machine.)

See also [reset-timer](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#reset-timer).

Note that the timer is different from the tick counter. The timer measures elapsed real time in seconds; the tick counter measures elapsed model time in ticks.

**to**

**to *procedure-name*to *procedure-name* [*input1* ...]**

Used to begin a command procedure.

to setup

clear-all

crt 500

end

to circle [radius]

crt 100 [ fd radius ]

end

**to-report**

**to-report *procedure-name*to-report *procedure-name* [*input1* ...]**

Used to begin a reporter procedure.

The body of the procedure should use report to report a value for the procedure. See [report](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#report).

to-report average [a b]

report (a + b) / 2

end

to-report absolute-value [number]

ifelse number >= 0

[ report number ]

[ report (- number) ]

end

to-report first-turtle?

report who = 0 ;; reports true or false

end

**towards***1.0*

**towards *agent*Turtle Command Patch Command**

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.

set heading towards turtle 1

;; same as "face turtle 1"

See also [face](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#face).

**towardsxy***1.0*

**towardsxy *x* *y*Turtle Command Patch Command**

Reports the heading from the turtle or patch towards the point (*x*,*y*).

If wrapping is allowed by the topology and the wrapped distance (around the edges of the world) is shorter, towardsxy will use the wrapped path.

Note: asking for the heading to the point the agent is already standing on will cause a runtime error.

See also [facexy](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#facexy).

**turtle***1.0*

**turtle *number*<breed> *number***

Reports the turtle with the given who number, or [nobody](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#nobody) if there is no such turtle. For breeded turtles you may also use the single breed form to refer to them.

ask turtle 5 [ set color red ]

;; turtle with who number 5 turns red

**turtle-set***4.0*

**turtle-set *value1*(turtle-set *value1* *value2* ...)**

Reports an agentset containing all of the turtles anywhere in any of the inputs. The inputs may be individual turtles, turtle agentsets, nobody, or lists (or nested lists) containing any of the above.

turtle-set self

(turtle-set self turtles-on neighbors)

(turtle-set turtle 0 turtle 2 turtle 9)

(turtle-set frogs mice)

See also [patch-set](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patch-set), [link-set](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#link-set).

**turtles***1.0*

**turtles**

Reports the agentset consisting of all turtles. This is a special agentset that can grow as turtles are added to the world, see [the programming guide for more info](http://ccl.northwestern.edu/netlogo/docs/programming.html#special-agentsets).

show count turtles

;; prints the number of turtles

**turtles-at***1.0*

**turtles-at *dx* *dy<breeds>*-at *dx* *dy*Turtle Command Patch Command**

Reports an agentset containing the turtles on the patch (dx, dy) from the caller. (The result may include the caller itself if the caller is a turtle.)

create-turtles 5 [ setxy 2 3 ]

show count [turtles-at 1 1] of patch 1 2

=> 5

If the name of a breed is substituted for "turtles", then only turtles of that breed are included.

**turtles-here***1.0*

**turtles-here*<breeds>*-hereTurtle Command Patch Command**

Reports an agentset containing all the turtles on the caller's patch (including the caller itself if it's a turtle).

crt 10

ask turtle 0 [ show count turtles-here ]

=> 10

If the name of a breed is substituted for "turtles", then only turtles of that breed are included.

breed [cats cat]

breed [dogs dog]

create-cats 5

create-dogs 1

ask dogs [ show count cats-here ]

=> 5

**turtles-on***2.0*

**turtles-on *agent*turtles-on *agentset<breeds>*-on *agent<breeds>*-on *agentset*Turtle Command Patch Command**

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.

ask turtles [

if not any? turtles-on patch-ahead 1

[ fd 1 ]

]

ask turtles [

if not any? turtles-on neighbors [

die-of-loneliness

]

]

If the name of a breed is substituted for "turtles", then only turtles of that breed are included.

**turtles-own**

**turtles-own [*var1* ...]*<breeds>*-own [*var1* ...]**

The turtles-own keyword, like the globals, breed, *<breeds>*-own, and patches-own keywords, can only be used at the beginning of a program, before any function definitions. It defines the variables belonging to each turtle.

If you specify a breed instead of "turtles", only turtles of that breed have the listed variables. (More than one turtle breed may list the same variable.)

breed [cats cat ]

breed [dogs dog]

breed [hamsters hamster]

turtles-own [eyes legs] ;; applies to all breeds

cats-own [fur kittens]

hamsters-own [fur cage]

dogs-own [hair puppies]

See also [globals](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#globals), [patches-own](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#patches-own), [breed](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#breed), [*<breeds>*-own](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#turtles-own).

**type***1.0*

**type *value***

Prints *value* in the Command Center, *not* followed by a carriage return (unlike [print](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#print) and [show](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#show)). The lack of a carriage return allows you to print several values on the same line.

This agent is *not* printed before the value. unlike [show](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#show).

type 3 type " " print 4

=> 3 4

See also [print](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#print), [show](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#show), [write](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#write), [output-type](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#output-cmds), and [Output (programming guide)](http://ccl.northwestern.edu/netlogo/docs/programming.html#output).

**U**

**undirected-link-breed**

**undirected-link-breed [*<link-breeds>* *<link-breed>*]**

This keyword, like the globals and breeds keywords, can only be used at the beginning of the Code tab, before any procedure definitions. It defines an undirected link breed. Links of a particular breed are always either all directed or all undirected. The first input defines the name of the agentset associated with the link breed. The second input defines the name of a single member of the breed.

Any link of the given link breed:

* is part of the agentset named by the link breed name
* has its built-in variable breed set to that agentset
* is directed or undirected as declared by the keyword

Most often, the agentset is used in conjunction with ask to give commands to only the links of a particular breed.

undirected-link-breed [streets street]

undirected-link-breed [highways highway]

to setup

clear-all

crt 2

ask turtle 0 [ create-street-with turtle 1 ]

ask turtle 0 [ create-highway-with turtle 1 ]

end

ask turtle 0 [ show sort my-links ]

;; prints [(street 0 1) (highway 0 1)]

See also [breed](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#breed), [directed-link-breed](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#directed-link-breed)

**untie***4.0*

**untieLink Command**

Unties *end2* from *end1* (sets [tie-mode](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#tie-mode) to "none") if they were previously tied together. If the link is an undirected link, then it will untie *end1* from *end2* as well. It does **not** remove the link between the two turtles.

See also [tie](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#tie)

See the [Tie](http://ccl.northwestern.edu/netlogo/docs/programming.html#tie) section of the Programming Guide for more details.

**update-plots***5.0*

**update-plots**

For each plot, runs that plot's update commands, including the update code for any pens in the plot.

[tick](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#tick) has the same effect, so in models that use the tick counter, this primitive is not normally used. Models that use fractional ticks may need update-plots, since [tick-advance](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#tick-advance) does not update the plots.

See the [Plotting section](http://ccl.northwestern.edu/netlogo/docs/programming.html#plotting) of the Programming Guide for more details.

See also [setup-plots](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#setup-plots).

**uphill***1.0***uphill4***1.0*

**uphill *patch-variable*uphill4 *patch-variable*Turtle Command**

Moves the turtle to the neighboring patch with the highest value for *patch-variable*. If no neighboring patch has a higher value than the current patch, the turtle stays put. If there are multiple patches with the same highest value, the turtle picks one randomly. Non-numeric values are ignored.

uphill considers the eight neighboring patches; uphill4 only considers the four neighbors.

Equivalent to the following code (assumes variable values are numeric):

move-to patch-here ;; go to patch center

let p max-one-of neighbors [*patch-variable*] ;; or neighbors4

if [*patch-variable*] of p > *patch-variable* [

face p

move-to p

]

Note that the turtle always ends up on a patch center and has a heading that is a multiple of 45 (uphill) or 90 (uphill4).

See also [downhill](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#downhill), [downhill4](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#downhill).

**user-directory***3.1*

**user-directory**

Opens a dialog that allows the user to choose an existing directory on the system.

It reports a string with the absolute path or false if the user cancels.

set-current-directory user-directory

;; Assumes the user will choose a directory

**user-file***3.1*

**user-file**

Opens a dialog that allows the user to choose an existing file on the system.

It reports a string with the absolute file path or false if the user cancels.

file-open user-file

;; Assumes the user will choose a file

**user-new-file***3.1*

**user-new-file**

Opens a dialog that allows the user to choose a location and name of a new file to be created. It reports a string with the absolute file path or false if the user cancels.

file-open user-new-file

;; Assumes the user will choose a file

Note that this reporter doesn't actually create the file; normally you would create the file using file-open, as in the example.

If the user chooses an existing file, they will be asked if they wish to replace it or not, but the reporter itself doesn't cause the file to be replaced. To do that you would use file-delete.

**user-input***1.1*

**user-input *value***

Reports the string that a user types into an entry field in a dialog with title *value*.

*value* may be of any type, but is typically a string.

show user-input "What is your name?"

See the [User Interaction Primitives section](http://ccl.northwestern.edu/netlogo/docs/programming.html#user-interaction-primitives) of the Programming Guide for additional details.

**user-message***1.1*

**user-message *value***

Opens a dialog with *value* displayed as the message to the user.

*value* may be of any type, but is typically a string.

user-message (word "There are " count turtles " turtles.")

Note that if a user closes the user-message dialog with the "X" in the corner, the behavior will be the same as if they had clicked "OK".

See the [User Interaction Primitives section](http://ccl.northwestern.edu/netlogo/docs/programming.html#user-interaction-primitives) of the Programming Guide for additional details.

**user-one-of***3.1*

**user-one-of *value* *list-of-choices***

Opens a dialog with *value* displayed as the message and *list-of-choices* displayed as a popup menu for the user to select from.

Reports the item in *list-of-choices* selected by the user.

*value* may be of any type, but is typically a string.

if "yes" = user-one-of "Set up the model?" ["yes" "no"]

[ setup ]

See the [User Interaction Primitives section](http://ccl.northwestern.edu/netlogo/docs/programming.html#user-interaction-primitives) of the Programming Guide for additional details.

**user-yes-or-no?***2.0*

**user-yes-or-no? *value***

Reports true or false based on the user's response to *value*.

*value* may be of any type, but is typically a string.

if user-yes-or-no? "Set up the model?"

[ setup ]

See the [User Interaction Primitives section](http://ccl.northwestern.edu/netlogo/docs/programming.html#user-interaction-primitives) of the Programming Guide for additional details.

**V**

**variance***1.0*

**variance *list***

Reports the sample variance of a *list* of numbers. Ignores other types of items.

(Note that this computes an unbiased estimate of the variance for a *sample*, rather than for a whole *population*, using Bessel's correction.)

The sample variance is the sum of the squares of the deviations of the numbers from their mean, divided by one less than the number of numbers in the list.

show variance [2 7 4 3 5]

=> 3.7

**W**

**wait***1.0*

**wait *number***

Wait the given number of seconds. (This needn't be an integer; you can specify fractions of seconds.) Note that you can't expect complete precision; the agent will never wait less than the given amount, but might wait slightly more.

repeat 10 [ fd 1 wait 0.5 ]

While the agent is waiting, no other agents can do anything. Everything stops until the agent is done.

See also [every](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#every).

**watch***3.0*

**watch *agent*Observer Command**

Puts a spotlight on *agent*. In the 3D view the observer will also turn to face the subject.

The observer may only watch or follow a single subject. Calling watch will undo perspective changes caused by prior calls to follow, follow-me, ride, and ride-me.

See also [follow](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#follow), [subject](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#subject), [reset-perspective](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#reset-perspective), [ride](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#ride), [ride-me](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#ride-me), [watch-me](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#watch-me).

**watch-me***3.0*

**watch-meTurtle Command Patch Command**

Asks the observer to watch this agent.

The observer may only watch or follow a single subject. Calling watch will undo perspective changes caused by prior calls to follow, follow-me, ride, and ride-me.

See also [follow](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#follow), [subject](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#subject), [reset-perspective](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#reset-perspective), [ride](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#ride), [ride-me](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#ride-me), [watch](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#watch).

**while***1.0*

**while [*reporter*] [ *commands* ]**

If *reporter* reports false, exit the loop. Otherwise run *commands* and repeat.

The reporter may have different values for different agents, so some agents may run *commands* a different number of times than other agents.

while [any? other turtles-here]

[ fd 1 ]

;; turtle moves until it finds a patch that has

;; no other turtles on it

**who**

**whoTurtle Command**

This is a built-in turtle variable. It holds the turtle's "who number" or ID number, an integer greater than or equal to zero. You cannot set this variable; a turtle's who number never changes.

Who numbers start at 0. A dead turtle's number will not be reassigned to a new turtle until you use the [clear-turtles](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#clear-turtles) or [clear-all](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#clear-all) commands, at which time who numbering starts over again at 0.

Example:

show [who] of turtles with [color = red]

;; prints a list of the who numbers of all red turtles

;; in the Command Center, in random order

crt 100

[ ifelse who < 50

[ set color red ]

[ set color blue ] ]

;; turtles 0 through 49 are red, turtles 50

;; through 99 are blue

You can use the turtle reporter to retrieve a turtle with a given who number. See also [turtle](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#turtle).

Note that who numbers aren't breed-specific. No two turtles can have the same who number, even if they are different breeds:

clear-turtles

create-frogs 1

create-mice 1

ask turtles [ print who ]

;; prints (in some random order):

;; (frog 0): 0

;; (mouse 1): 1

Even though we only have one mouse, it is mouse 1 not mouse 0, because the who number 0 was already taken by the frog.

**with***1.0*

***agentset* with [*reporter*]**

Takes two inputs: on the left, an agentset (usually "turtles" or "patches"). On the right, a boolean reporter. Reports a new agentset containing only those agents that reported true -- in other words, the agents satisfying the given condition.

show count patches with [pcolor = red]

;; prints the number of red patches

**<breed>-withlink-with***4.0*

**<breed>-with *turtle*link-with *turtle*Turtle Command**

Reports a link between *turtle* and the caller (directed or undirected, incoming or outgoing). If no link exists then it reports nobody. If more than one such link exists, reports a random one.

crt 2

ask turtle 0 [

create-link-with turtle 1

show link-with turtle 1 ;; prints link 0 1

]

See also: [in-link-from](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#in-link-from), [out-link-to](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#out-link-to)

**with-max***2.1*

***agentset* with-max [*reporter*]**

Takes two inputs: on the left, an agentset (usually "turtles" or "patches"). On the right, a reporter. Reports a new agentset containing all agents reporting the maximum value of the given reporter.

show count patches with-max [pxcor]

;; prints the number of patches on the right edge

See also [max-one-of](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#max-one-of), [max-n-of](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#max-n-of).

**with-min***2.1*

***agentset* with-min [*reporter*]**

Takes two inputs: on the left, an agentset (usually "turtles" or "patches"). On the right, a reporter. Reports a new agentset containing only those agents that have the minimum value of the given reporter.

show count patches with-min [pycor]

;; prints the number of patches on the bottom edge

See also [min-one-of](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#min-one-of), [min-n-of](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#min-n-of).

**with-local-randomness***4.0*

**with-local-randomness [ *commands* ]**

The commands are run without affecting subsequent random events. This is useful for performing extra operations (such as output) without changing the outcome of a model.

Example:

;; Run #1:

random-seed 50 setup repeat 10 [ go ]

;; Run #2:

random-seed 50 setup

with-local-randomness [ watch one-of turtles ]

repeat 10 [ go ]

Since one-of is used inside with-local-randomness, both runs will be identical.

Specifically how it works is, the state of the random number generator is remembered before the commands run, then restored afterwards. (If you want to run the commands with a fresh random state instead of the same random state that will be restored later, you can begin the commands with random-seed new-seed.)

The following example demonstrates that the random number generator state is the same both before the commands run and afterwards.

random-seed 10

with-local-randomness [ print n-values 10 [random 10] ]

;; prints [8 9 8 4 2 4 5 4 7 9]

print n-values 10 [random 10]

;; prints [8 9 8 4 2 4 5 4 7 9]

**without-interruption***1.1*

**without-interruption [ *commands* ]**

This primitive exists only for backwards compatibility. We don't recommend using it in new models.

The agent runs all the commands in the block without allowing other agents using ask-concurrent to "interrupt". That is, other agents are put "on hold" and do not run any commands until the commands in the block are finished.

Note: This command is only useful in conjunction with ask-concurrent.

See also [ask-concurrent](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#ask-concurrent).

**word***1.0*

**word *value1* *value2*(word *value1* ...)**

Concatenates the inputs together and reports the result as a string.

show word "tur" "tle"

=> "turtle"

word "a" 6

=> "a6"

set directory "c:\\foo\\fish\\"

show word directory "bar.txt"

=> "c:\foo\fish\bar.txt"

show word [1 54 8] "fishy"

=> "[1 54 8]fishy"

show (word 3)

=> "3"

show (word "a" "b" "c" 1 23)

=> "abc123"

**world-width***3.1***world-height***3.1*

**world-widthworld-height**

These reporters give the total width and height of the NetLogo world.

The width equals max-pxcor - min-pxcor + 1 and the height equals max-pycor - min-pycor + 1.

See also [max-pxcor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#max-pcor), [max-pycor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#max-pcor), [min-pxcor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#min-pcor), and [min-pycor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#min-pcor)

**wrap-color***1.0*

**wrap-color *number***

wrap-color checks whether *number* is in the NetLogo color range of 0 to 140 (not including 140 itself). If it is not, wrap-color "wraps" the numeric input to the 0 to 140 range.

The wrapping is done by repeatedly adding or subtracting 140 from the given number until it is in the 0 to 140 range. (This is the same wrapping that is done automatically if you assign an out-of-range number to the color turtle variable or pcolor patch variable.)

show wrap-color 150

=> 10

show wrap-color -10

=> 130

**write***2.0*

**write *value***

This command will output *value*, which can be a number, string, list, boolean, or nobody to the Command Center, *not* followed by a carriage return (unlike [print](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#print) and [show](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#show)).

This agent is *not* printed before the value, unlike [show](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#show). Its output also includes quotes around strings and is prepended with a space.

write "hello world"

=> "hello world"

See also [print](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#print), [show](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#show), [type](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#type), [output-write](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#output-cmds), and [Output (programming guide)](http://ccl.northwestern.edu/netlogo/docs/programming.html#output).

**X**

**xcor**

**xcorTurtle Command**

This is a built-in turtle variable. It holds the current x coordinate of the turtle. You can set this variable to change the turtle's location.

This variable is always greater than or equal to (min-pxcor - 0.5) and strictly less than (max-pxcor + 0.5).

See also [setxy](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#setxy), [ycor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#ycor), [pxcor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#pcor), [pycor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#pcor),

**xor***1.0*

***boolean1* xor *boolean2***

Reports true if either *boolean1* or *boolean2* is true, but not when both are true.

if (pxcor > 0) xor (pycor > 0)

[ set pcolor blue ]

;; upper-left and lower-right quadrants turn blue

**Y**

**ycor**

**ycorTurtle Command**

This is a built-in turtle variable. It holds the current y coordinate of the turtle. You can set this variable to change the turtle's location.

This variable is always greater than or equal to (min-pycor - 0.5) and strictly less than (max-pycor + 0.5).

See also [setxy](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#setxy), [xcor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#xcor), [pxcor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#pcor), [pycor](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#pcor),

**->**

**->***6.0*

**[ [*args*] -> *commands* ][ [*args*] -> *reporter* ]**

Creates and reports an anonymous procedure - a command or reporter - depending on the input. Within *commands* or *reporter* the listed *args* may be used just as you would use let or procedure variables. The variable names in *args* have the same restrictions as variable names of commands and reporters. In addition, they must not match the name of any let or procedure variable in their procedure.

Anonymous procedures are commonly used with the primitives [foreach](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#foreach), [map](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#map), [reduce](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#reduce), [filter](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#filter), [sort-by](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#sort-by), and [n-values](http://ccl.northwestern.edu/netlogo/docs/dictionary.html#n-values). See those entries for example usage.

See the [Anonymous Procedures section](http://ccl.northwestern.edu/netlogo/docs/programming.html#anonymous-procedures) of the Programming Guide for details.