**Common errors and how to fix them**

Contents

[No closing bracket for this open bracket 1](#_Toc113362547)

[Expected command 1](#_Toc113362548)

[Nothing names [name] has been defined 2](#_Toc113362549)

[END expected 2](#_Toc113362550)

[OF expected this input to be a reporter block, but got a number instead. 3](#_Toc113362551)

[WITH expected this input to be a TRUE/FALSE block, but got a number instead 3](#_Toc113362552)

[ASK expected this input to be an agent or agentset, but got NOBODY instead. 4](#_Toc113362553)

[… expected this input to be an … here, but got a ….instead. 5](#_Toc113362554)

[You can’t use … in a turtle/patch context, because … is observer only. 5](#_Toc113362555)

[Keyword expected 6](#_Toc113362556)

# No closing bracket for this open bracket

This error occurs when you are missing a closing bracket.

*Ask turtles*

*[*

*if size < 3*

*[*

*set size size + 1*

*]*

To fix, click each bracket and its pair will be highlighted. If no other bracket is highlighted, you know where the missing one is

*Ask turtles*

*[*

*if size < 3*

*[*

*set size size + 1*

*]*

*]*

# Expected command

This error occurs when you have one too many brackets.

Ask turtles

[

if size < 3

[

set size size + 1

]

]

]

Error will either state which line the extra bracket is on or will highlight the line. Just delete the extra bracket

Ask turtles

[

if size < 3

[

set size size + 1

]

]

# Nothing names [name] has been defined

A button will turn red if it doesn’t have a matching procedure defined in the code tab. Another example of code being thrown is due to the term not being defined as a global or breed attribute.

To seutp

Clear-all

End

To go

Ask turtles [ if stress-level > 100 [ rest ] ]

Tick

end

Making sure the spelling is correct or that the term is defined correctly at the top of the code will fix error.

To setup

Clear-all

End

Turtles-own [ stress-level ]

…

To go

Ask turtles [ if stress-level > 100 [ rest ] ]

Tick

End

# END expected

Occurs when we forget to add “end” to the end of a procedure.

To go

Ask turtles [ forward 1 ]

Netlogo will either state which line the “end” is missing from, or will highlight the line of code.

To go

Ask turtles [ forward 1 ]

end

# OF expected this input to be a reporter block, but got a number instead.

Occurs when a reporter preceding the “of” primitive is not enclosed in brackets ([ ]).

Ask turtles

[

If mean size of turtles > size

[

Set size size + 1

]

]

To fix, make sure the preceding reporter is in brackets

Ask turtles

[

If mean [ size ] of turtles > size

[

Set size size + 1

]

]

# WITH expected this input to be a TRUE/FALSE block, but got a number instead

Occurs when you forget to add brackets around the true/false statement that follows the “with” primitive.

Ask turtles with size > 1

[

Forward 1

]

To fix this error, add brackets around the proceeding reporter.

Ask turtles with [size > 1 ]

[

Forward 1

]

# ASK expected this input to be an agent or agentset, but got NOBODY instead.

Occurs when trying to access an agent that does not exist. Agent may have died or never existed in the first place.

To setup

Clear-all

Create-turtles

[

Set color one-of [ red blue ]

]

Ask turtles with [color = green ]

[

Forward 1

]

End

There are 2 hypothetical fixes for this error. Either make some of the turtles green or change our “ask” statement and use either “red” or “blue” in the conditional

To setup

Clear-all

Create turtles

[

Set color one-of [ red blue green ]

]

Ask turtles with [color = green ]

[

Forward 1

]

End

To setup

Clear-all

Create turtles

[

Set color one-of [ red blue green ]

]

Ask turtles with [ color = red ]

[

Forward 1

]

# … expected this input to be an … here, but got a ….instead.

Generic error that means there is a syntax error such as giving a primitive the wrong type of input.

Ask turtles

[

Face neighbors

]

Since in this context “neighbors” has multiple agents, so the “one-of” primitive needs to be used.

Ask turtles

[

Face one-of neighbors

]

# You can’t use … in a turtle/patch context, because … is observer only.

Occurs when trying to use a primitive in a context it is not meant for (i.e., patch-specific, turtle-specific, observer-specific)

Ask patches

[

Create-turtles 1

]

“create-turtles” is an observer-specific primitive. Is you want to ask patches, the “sprout” primitive is similar to “create-turtles”

Ask patches

[

Sprout 1

]

This error applies to other agent types also. Some examples are:

You can’t use SETXY in a patch context, because SETXY is turtle only.

You can’t use DIE in a patch context, because DIE is turtle/link only.

You can’t use CREAT-TURTLES in a turtle context, because CREATE-TURTLES is observer only.

# Keyword expected

Occurs when code is not contained within a procedure. In the “CODE” tab, all code must be writing with in a procedure, except for defining global, breed, turtle, or patch variables.

Globals [ stress-level ]

Clear-all

Ask turtles

[

Set stress-level 0

]

To fix, enclose the code within a procedure, beginning with “to” and ending with “end”

Globals [ stress-level ]

To setup

Clear-all

Ask turtles

[

Set stress-level 0

]

end