

Leone Learning Systems, Inc.

Wonder. Create. Grow.

Leone Learning Systems, Inc.
237 Custer Ave
Evanston, IL 60202
Email tj@leonelearningsystems.com

Phone 847 951 0127
Fax 847 733 8812

Kinds of Primitives

TJ Leone

July 2009



Introduction

MicroWorlds has a set of built in procedures called *primitives*. Primitives fit into different categories—*commands* and *reporters*. Commands have some effect on the MicroWorlds environment. For example, commands are used to move a turtle, change the contents of a textbox, or write words and lists in the Command Center. Reporters provide information needed by other procedures, like a turtle's position on the Main Screen, a list of pages in a project, or the sum of two numbers.

This document puts Logo primitives into different categories in order to help readers quickly recognize the role of those primitives in MicroWorlds projects.



Getters and Setters

Getters are reporters used to get information about the MicroWorlds environment. Setters are commands used to change or update information.

Getters are Reporters

Sometimes we want to ask questions about a certain value, for example, the color of a turtle, the number a slider is set to, or the contents of a text box.

To answer these questions, we use reporters called *getters*. Examples are `color`, `fontsize`, and `pagelist`.

Getters don't need any input.

Getters are reporters because they report information about turtles, text boxes, or projects. They can also report the values of sliders, the contents of text boxes, or the value of project variables.

Setters are Commands

When we want to change information that belongs to something in the MicroWorlds environment, we use setters like `setc` or `setfont`.

Setters take one input, which is the new value for the color, `fontsize`, or whatever.

Setters are commands because they give the order to change a value.

Setters and their getter partners

Most setters and getters come in pairs. For example, there's a setter (`setx`) to set the value of a turtle's x coordinate and a getter (`xcor`) to get the value of a turtle's `xcor`. The following tables show some setters and their getter partners (when there is one).

⋮

Getters and setters (continued)

Turtle setters and getters

Setter (Command)	Getter (Reporter)
setc	color
<none>	colorunder
seth	heading
setinstruction	<none>
setpensize	pensize
setpos	pos
setsh	shape
setshape	shape
setsize	size
setx	xcor
sety	ycor

Textbox setters and getters

Setter (Command)	Getter (Reporter)
setfont	<none>
setfontsize	fontsize
setstyle	<none>
settc	tc

⋮

Getters and setters (continued)

Setters and Getters using object names

Sometimes we can use the name of an object to get the value contained in an object. For example, we can use the name of a slider to give us the value in the slider. In this case, the name of the slider--without a quote--is a getter.

For these objects, we can also use set together with the object name as a setter.

Setter (Command)	Getter (Reporter)
<code>set<slider name> number</code>	<code><slider name></code>
<code>set<text box name> word-or-list</code>	<code><text box name></code>
<code>set<projectvar name> word-or-list</code>	<code><projectvar name></code>

For example, if we created a slider named *Speed*, we could set the value of the slider to 50 with the instruction `setSpeed 50`, and we could use the reporter *Speed* as input to other procedures. Here are some other examples:

```
setScore 5      ; puts the number 5 into a textbox named Score
show Gravity    ; shows the value of a slider named Gravity
fd SideLength  ; uses the value of a projectvar call SideLength as input to fd
```



Kinds of reporters

Getters are only one kind of reporter. Others include predicates, math operators, word and list manipulators, and reporters used in the context of `onTouching` and `onMessage` events.

Predicates

Predicates are reporters that have only two possible outputs: `true` or `false`.

```
touching? turtle-name turtle-name  
equal? word-or-list1 word-or-list2  
member? word-or-list1 word-or-list2  
word? word-or-list  
empty? word-or-list  
identical? word-or-list  
number? word-or-list
```

Math Operators

```
sum number1 number2  
difference number1 number2  
product number1 number2  
quotient number1 number2  
remainder number1 number2  
number1 + number2 ; The + is a special reporter called an infix operator  
number1 - number2 ; The - is a special reporter called an infix operator  
number1 * number2 ; The * is a special reporter called an infix operator  
number1 / number2 ; The / is a special reporter called an infix operator  
abs number  
random number
```

⋮

Kinds of reporters (continued)

Word and List Manipulators

Selectors

Some reporters that manipulate words and lists are *selectors*. They select information out of the word or list.

Here are some selectors:

first *word-or-list*

last *word-or-list*

item *number word-or-list*

Constructors

Some reporters that manipulate words and lists are *constructors*. They construct new words or lists by putting together inputs.

Here are some constructors:

word *word-or-list1 word-or-list2*

list *word-or-list1 word-or-list2*

sentence *word-or-list1 word-or-list2*

⋮

Kinds of reporters (continued)

Reporters used with onTouching and onMessage

Some reporters only have meaning in the context of an event. For example, the `touchedturtle` reporter can be used in the context of an `onTouch` event. Suppose you have two turtles on your screen—one named Dog, another named Cat and another named Mouse. In the `onTouch` field of the Dog’s backpack, you have an instruction like this:

```
ask touchedturtle [bk 50]
```

The reporter `touchedturtle` reports the name of the turtle touched. If Dog and Cat touched, then `touchedturtle` would report the name “Cat. If Dog and Mouse touched, then `touchedturtle` would report the name “Mouse.

You could also use the name `touchedturtle` inside of a procedure, *but only if that procedure is invoked when a turtle is touched.*

There are also two event reporters that can be used in the context of an `onMessage` event:

<code>message</code>	reports the message sent with the <code>broadcast</code> or <code>tell</code> commands
<code>sender</code>	reports the name of the sender of the message



Kinds of Commands

In addition to the Setter category, there are other categories we can make for commands.

Turtle commands

```
rt number  
fd number  
pd
```

Most setters are also turtle commands, but as we have seen, some setters are commands that affect other objects like textboxes, sliders, and project variables.

Text box commands

```
print word-or-list
```

It is important to understand the difference between reporting information (producing output) and printing information to a text box or showing it in the command center.

Reports report information that is used as input to commands or other reporters. Commands give orders, like “move the turtle forward 50 steps” or “display information in the text box” or “display information in the command center”.

Creators

Some commands are used to create new objects.

```
newslider name [x y] [min max current]
```

```
newtext name [x y] [xsize ysize]
```

```
createprojectvar word
```



Challenges

Write a procedure that uses the reporters below to report the number 17. Have each one report its output to the show command.

- A getter.
- A math operator.
- A word manipulator (constructor).
- A list manipulator (selector).

Write a procedure that uses the following as the first input to each of five different if statements:

- A predicate
- The name of a text box
- A projectvar (see createprojectvar)
- A word constructor
- A list selector

Write a procedure that uses the following as the second input to each of three different if statements:

- A list constructor
- A list selector
- The parse reporter and the name of a text box



The Author

TJ Leone is a teacher at Chiaravalle Montessori School in Evanston, Illinois. He has a BA in Math and an MS in Computer Science, both from the City College of New York. He spent two years in graduate studies in education and computer science at Northwestern University, and six years developing educational software there. He teaches gifted children on a part time basis at the Center for Talent Development at Northwestern University in addition to his tutoring and software development work. He is certified by the Association Montessori Internationale (AMI) to teach children aged 6 to 12 and by Sun as a java programmer. His web site is <http://www.tjleone.com>.