

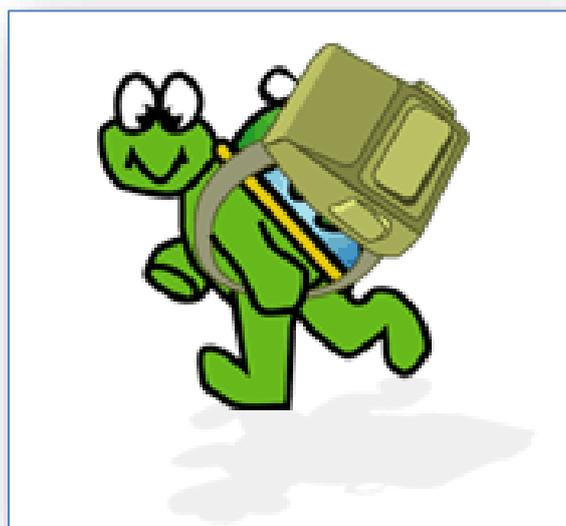
**EVANGELOS C. ZIOULAS**

IT Teacher



**CHAPTER  
5**

**LOGO BASICS**



## INTRODUCTION TO LOGO

- Logo is a programming language.
- In contrast with other natural human languages, it has a smaller vocabulary as well as a simpler set of grammatical rules.
- While a natural language creates sentences to express itself, Logo has **instructions**.
- To form instructions in Logo, we must follow a set of rules which are simpler but stricter and constitute the "**grammar**" of Logo.

## VOCABULARY

- In MicroWorlds, every word is interpreted as a **request** to do something.
- There are some words that MicroWorlds already knows about because they are built in.
- Words that are built into MicroWorlds are called **primitives** e.g. `forward`, `right`, `setcolor` etc.
- The MicroWorlds vocabulary has an extensive list of primitives that are used as **reserved words** by the Logo language and have a special meaning.
- However, there are also some words that the user can define and add them to the vocabulary, such as the **names** of the **objects** he creates (e.g. buttons, text boxes, shapes etc.) and the **procedures** he creates (his own commands).
- In fact, adding objects and procedures to MicroWorlds vocabulary is what programming in MicroWorlds is all about.

## COMMANDS & REPORTERS

- All of the words in MicroWorlds, **primitives**, **procedures** and **object names** included, can be classified into two categories: commands and reporters.

**Commands** do something.

### Examples

`forward`, `show`, `right` and `page1` are commands.

<code>forward 100</code>	<code>forward</code> tells MicroWorlds to move the turtle a given amount.
<code>show heading</code>	<code>show</code> tells MicroWorlds to display the turtle's heading in the Command Center.
<code>right 90</code>	<code>right</code> tells MicroWorlds to turn the turtle 90 degrees to the right
<code>page1</code>	opens <code>page1</code> of our project (given that you do have a page with that name).

**Reporters** report something. They tell us the answer to a calculation or the state of an object. The reported data must be received by another procedure. `heading`, `first` and `text1` are reporters.

### Examples

<code>show heading</code>	<code>heading</code> reports the current heading of the turtle to the <code>show</code> command.
<code>setc 15</code>	<code>15</code> reports the color number (red) of the turtle to the <code>setc</code> command
<code>setsh 2</code>	<code>2</code> reports the current shape of the turtle to the <code>setsh</code> command
<code>show text1</code>	<code>text1</code> reports the contents of the text box (given that we have this text box)

## DATA TYPES

MicroWorlds can work with three kinds of data: **numbers**, **words**, and **lists**. These data can be used as **input** to a procedure, and can be the **results** that a procedure reports, or can be the **value** of a variable.

### NUMBERS

They are **integer** or **floating point** (real) values used in arithmetic operations or functions (we use a comma or a period to express the decimal point) e.g.

113  
15.2  
-16.03  
323. (equals 323.0)  
.5 (equals 0.5)  
1E4 (equals  $1 \times 10^4 = 10000$ )

Floating-point numbers consist of:

An optional **minus sign (-)**.

An **integer part** which needs not to be present if its value is zero.

A **decimal point**, needed only if the number has a fractional part.

A **fractional part** which needs not to be present if its value is zero.

An **optional exponent**: the letter E, a sign, and an integer up to 300.

### Note:

The + (plus) sign can be used to indicate positive numbers but its use is optional. Both the - (**minus**) sign and the + (**plus**) sign must immediately precede the number to indicate a negative or positive number e.g.

```
show -2 + 5
3
```

If a space is inserted between the "-" sign and the number, the "-" is interpreted as the subtraction operation.

```
show - 2 + 5
- needs more inputs
```

### WORDS

A **word** is made from a **sequence of characters** (e.g. letters, digits or punctuation) which are called **elements** and must always begin with **quotation marks "** e.g.

"Hello	(word with 5 elements)
"Hi	(word with 2 elements)
"x	(word with 1 element)
"3.14	(word with 4 elements)
"314	(word with 3 elements)
"MICROWORLDS	(word with 11 elements)
"Micro.Pro	(word with 9 elements)
"Maria35	(word with 7 elements)
"who?	(word with 4 elements)

```
show hello
I don't know how to hello

show "hello
hello

show "hello"
hello"
```

### LISTS

A list is a **sequence of words** separated by **spaces** and enclosed in **square brackets [ ]**.

The words in a list do not need quotation marks and extra spaces are ignored.

The outer brackets are not displayed when **show** is used.

[Hello my dear friend]	(list with 4 words)
[My name is Maria]	(list with 4 words)
[Hello]	(list with 1 word)
[today is 24/5/13]	(list with 3 words)

```
show [hello my friend]
hello my friend

show [red GREEN blue]
red GREEN blue
```

## INPUT – OUTPUT COMMANDS

### SHOW

It is an **output command** because it prints pieces of information in the command center such as: **numbers, words, lists or sentences** (a sentence is a list of items that can be either words or lists).

A) **show numbers:** Microworlds can do all the arithmetic operations ( + - \* / ).

In an arithmetic expression we should always place a **space** before and after the sign of operation. Otherwise, the system gives back a message of a syntax error.

```
show 7
7
```

```
show 3 * 8
24
```

```
show 45 / (5 * 3)
3
```

```
show 18 / 3 * 4
24
```

```
show 5*6
I don't know how to 5*6
```

```
show 5 + 15/5
I don't know how to 15/5
```

B) **show words:** (we should not forget the **quotation marks** ")

```
show "hello"
hello
```

```
show "hello"
hello"
```

```
show "2+3"
2+3
```

C) **show lists:** (we should not forget the **square brackets** [ ])

```
show [My name is Evangelos Zioulas]
My name is Evangelos Zioulas
```

```
show [Programming in Logo]
Programming in Logo
```

D) **show sentences:** (we should not forget the **se command**)

If a sentence includes more than 2 items, they should be enclosed in parentheses.

```
show (se [Cost of the trip is] 200 / 25 "euro)
Cost of the trip is 8 euro
```

## ANNOUNCE

It is an **output command** that displays the *word-or-list* message that follows in an **alert box**. To close the box, we should click OK. If we drag the alert box to a new position while it is being displayed, this is the position where the next alert box is opened in the project.

```
announce [Programming in Logo...]
```



## QUESTION

It opens a **dialogue box** displaying the **question** and an **area to type** the answer. The **Answer** command after that, reports what was typed in the dialogue box.

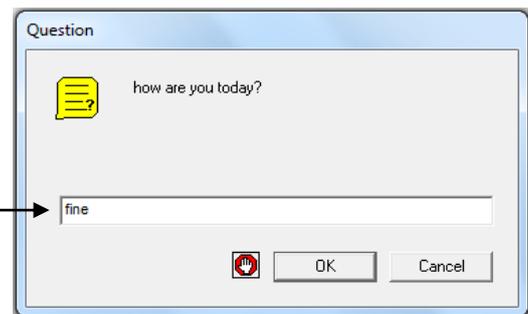
If we write a very long question, only the part that fits is displayed. If we drag the dialogue box to a new position while the question is being displayed, this is the position where the next question dialogue is opened in the project.

Question is an **input command** that allows us to insert a value as an answer to the system.

```
question [How are you today?]
```



Let's say the answer is  
fine



## ANSWER

It is an **output reporter** that reports the contents of the last answer typed in the **question** dialogue box.

The value reported by answer is always overwritten by the last usage of a **question** or a dialogue box.

Using the **question** and **answer** commands, we can create an interactive program that performs a dialogue between the system and the user.

```
announce (se [Today you feel] answer)
```

