Introduction to Lynx Words, Lists and Ciphers

Gary Stager, Ph.D.

Words in Lynx begin with quotation marks as in:

```
show “Gary
```

Lynx lists are a collection of words or other lists, such as:

```
show [lemon grape [apple pie] strawberry]
```

The list above has 4 elements, 3 words and 1 list. A good deal of computer programming involves taking things apart and putting things together. In this activity, we will take things apart.

1) `ASCII` is a reporter. Try typing the following in the command center:

```
show ascii “a
show ascii “e
show ascii “z
```

What does `ASCII` do? _____

2) `CHAR` is another reporter. Try the following in the command center:

```
show char 97
show char 98
show char 111
```

What does `CHAR` do? _____

If `ascii “a = 97`, how can we change that number to equal 1?

`97` ☐ `____ = 1`

3) Try the following in the command center:

```
show first “apple
show last “apple
```

What does the reporter, `first`, do?

What does the reporter, `last`, do?

4) Predict what each of these instructions will do before you try them.

```
show first [apple peach pear]
show last [apple peach pear]
```

How accurate were your predictions?

http://lynxcoding.club

Lynx is a web-based version of the Logo programming language. Programs written for its predecessor, MicroWorlds, or other Logo dialects should work with minor tweaks

Logo was developed as the first programming language specifically for children in 1967. It is based on LISP, a language still in use today, especially in artificial intelligence applications

Scratch, SNAP!, MakeCode, Turtle Art, Turtle Stitch, Beetle Blocks, and even Python owe a great deal to Logo.

5) What do you think will happen if you type the following? Make a prediction and then run the instructions in the command center. Write the results next to the instruction.

```
show first first [apple peach pear]
show last first [apple peach pear]
show first last [apple peach pear]
show last last [apple peach pear]
```

How accurate were your predictions?

6) Predict the result of the following instructions before typing them into the command center. Write the results next to the instruction.

```
show bf “lemon
show bl “grape
show bf bf “grape
show bl bf “grape
show bf bl “grape
show bl bl “grape
```

What does bf do? _____

What does bl do? _____

7) Predict the result of the following instructions before typing them into the command center. Write the results next to the instruction.

```
show bf [apple grape peach]
show bl [apple grape peach]
show bf bf [apple grape peach]
show bl bf [apple grape peach]
show bf bl [apple grape peach]
show bl bl [apple grape peach]
show bf bl [apple grape peach]
```

8) Predict the result of the following instructions before typing them into the command center. Write the results next to the instruction.

```
show first bf “grape
show first bl “grape
show last bf bf [apple grape peach]
show first bl bf [apple grape peach]
show first bf bl “grape
show first bl bl “grape
show last bf bl “grape
```

9) Type this procedure in the procedures center:

```
to eat :thing

show first :thing
eat bf :thing
end
```

Try running the procedure above by typing the following in the command center:

```
eat "lemon
eat [apple peach grape lemon]
```

An error message, first does not like as input in eat, is generated. It means that the procedure tried to grab the first thing out of nothing after you ate all of the other items in the word or list. Therefore, we need a common instruction, called a *stop rule* added to the procedure.

Change the `eat` procedure in the procedures center to include the the stop rule (beginning with `IF`)

```
to eat :thing
if empty? :thing [stop]
show first :thing
eat bf :thing
end
```

Try running the procedure above by typing the following in the command center:

```
eat "lemon
eat [apple peach grape lemon]
```

Is the error message gone?

9.5) Can you write a procedure that spells a word backwards?

10) Caesar's Cipher Level 1

```
to caesar :word
if empty? :word [stop]
show (ascii first :word) - 96
caesar bf :word
end
```

Try running the procedure above by typing the following in the command center:

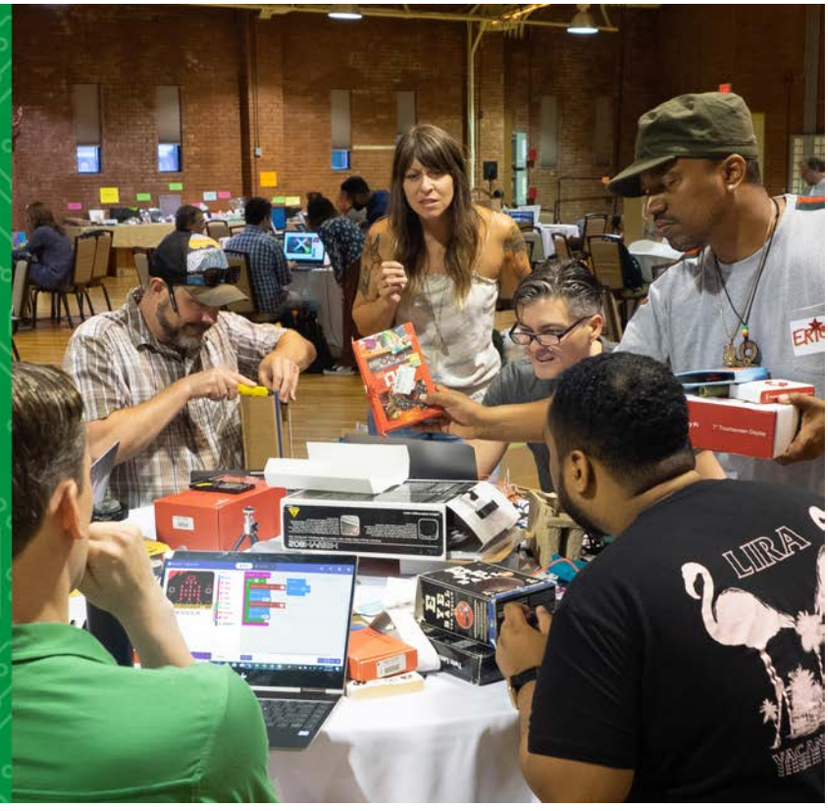
```
caesar "touchdown
caesar "school
```

11) Think about how we should improve our cipher program!

Constructing Modern Knowledge

An exceptional summer learning adventure for creative educators

July 11 - 14, 2023
Manchester, NH USA
constructingmodernknowledge.com



Constructing Modern Knowledge

STEAM · Project-based Learning · Coding · Making

The **Constructing Modern Knowledge Summer Institute** is a life-changing educational experience for PK-12 educators and teacher educators. Participants will enjoy 4 days immersed in remarkable cross-curricular project development with modern materials and supported by a world-class faculty. *Space is limited–register today!*

Extraordinary pre-institute workshops

Scintillae – Play and Learning in the Digital Age – Led by educators from the scintillae team of Reggio Emilia, Italy. *A once-in-a-lifetime collaboration between the Reggio Children Foundation and Constructing Modern Knowledge.*

Introduction to Learning with Electronics – Get a jumpstart on learning with electronics and microcontrollers. *A popular choice for first-time CMK participants.*

Bring a team and transform your school!

July 11 - 14, 2023
Manchester, NH USA
constructingmodernknowledge.com

Remarkable Guest Speakers

Gareth Stockdale: Lead designer of the BBC micro:bit and CEO of the micro:bit Educational Foundation.

Charlie Rosen: Composer, orchestrator, player of 70 instruments, Tony, Obie, & Grammy winner. Leader of the 8-Bit Big Band.

Cynthia Solomon: Mother of Educational Computing and co-designer of the Logo programming language.

Howard Gardner: MacArthur Genius educator, philosopher, author, scholar, father of multiple intelligences theory. Professor Gardner will lead a virtual fireside chat, exclusively for CMK participants.

Hosted by the authors of the best-selling book, *Invent To Learn - Making, Tinkering, and Engineering in the Classroom*