

Appendix A - Top 50 Lynx Primitives

This appendix contains a list of the most popular Lynx primitives - much more than what you need to create the projects in this book. If you wish to see a complete list of about 200 Lynx primitives, find the PDF called **List of Lynx primitives**, on the Lynx web site (lynxcoding.club), click on **Help** and **User Guides**.

You can also check the online Help pages (from inside the Lynx editor), where the definitions and examples are a bit more elaborate.

All the examples below assume you have a **turtle** in the Work Area with its pen down, and in many cases, an empty **text box**.

When you see *number*, it means the Primitive requires an input (a number).

In the left column you have the primitive name and the short form (if any) and examples of use. In the right column you have an explanation of what the primitive does.

Turtles and graphics

TURTLE MOVEMENT

<p>FORWARD <i>number</i> FD <i>number</i></p> <p>forward 100 cg repeat 4 [forward 100 right 90] cg repeat 36 [forward 100 back 80 right 10]</p>	<p>This command makes the turtle move in the direction it is facing. If the pen is down, the turtle leaves a line.</p>
<p>BACK <i>number</i> BK <i>number</i></p> <p>back 100 cg repeat 4 [back 100 right 90] cg repeat 36 [forward 100 back 80 right 10]</p>	<p>This command makes the turtle move backwards. If the pen is down, the turtle leaves a line.</p>
<p>RIGHT <i>number</i> RT <i>number</i></p> <p>repeat 4 [forward 100 right 90] cg right 90 (turns right 90 degrees, based on the previous heading)</p>	<p>This command makes the turtle turn to the right, without moving.</p> <p>Right is relative to the current heading. If you absolutely want to point East, regardless of your current heading, see setheading below.</p>

<p>LEFT <i>number</i> LT <i>number</i></p> <p>repeat 4 [forward 100 left 90] cg left 90 (turns left 90 degrees, based on the previous heading)</p>	<p>This command makes the turtle turn to the left, without moving.</p> <p>Left is relative to the current heading. If you absolutely want to point West, regardless of your current heading, see setheading below.</p>
<p>HOME</p> <p>pd right 11 forward 125 home</p>	<p>This command moves the current turtle to the centre of the page, its “home” base. The turtle’s position is [0 0] and its heading is 0. If the pen is down, a line will appear between the current position and home position.</p>
<p>GLIDE</p> <p>glide 500 1 glide 500 5</p>	<p>This command makes the current turtle move forward. The first number is for distance and the second number is for speed. The effect of glide 500 1 is very similar to repeat 500 [forward 1 wait 1].</p>
<p>POS</p> <p>home show pos right 45 forward 250 show pos</p>	<p>This reporter returns the current position of the current turtle, as a list of two numbers. Use this reporter to find out the position of a turtle, so you can later use this value with setpos in a setup procedure. The position at the centre of the page is [0 0].</p>
<p>SETHEADING <i>number</i> SETH <i>number</i></p> <p>home setheading 90 right 180 setheading 90</p>	<p>This command sets the heading of the current turtle. The heading corresponds to the values on a compass: 0 (or 360) is due North, 90 is due East, 180 is due South and 270 is due West. Setheading is absolute: if you run the instruction setheading 90 twice, the heading will always be due East. Right and left are relative. If you run the instruction right 90 several times, the turtle will keep turning, always 90 degrees relative to its current heading.</p>

Appendix A - Top 50 Lynx Primitives

<p>SETPOS [<i>xcor ycor</i>]</p> <pre>setpos [50 50] setpos [-50 50]</pre>	<p>This command moves the turtle to the position indicated. If the turtle's pen is down, it leaves a line. The position at the centre of the page is [0 0].</p>
<p>SETSHAPE <i>clipart_number_or_name</i> SETSHAPE <i>list_of_numbers_or_names</i></p> <pre>setshape 1 setshape 'cat' setshape [1 2] repeat 50 [forward 3 wait 6]</pre>	<p>This command sets the shape (clipart) of the current turtle. You must first bring at least one clipart into your clipart pane. Right-click on a clipart if you want to give it a name.</p> <p>The input of setshape can be a single number or clipart name, but you can also use a list of numbers or names. If you do so, the turtle will switch shape each time it moves (forward, back, setx, sety, setpos, etc.)</p>
<p>SETX <i>number</i></p> <pre>setx 100 setx -100</pre>	<p>This command moves the turtle horizontally to the x coordinate indicated.</p>
<p>XCOR</p> <pre>home show xcor right 45 forward 100 show xcor</pre>	<p>This reporter returns the x coordinate of the current turtle.</p>
<p>SETY <i>number</i></p> <pre>sety 100 sety -100</pre>	<p>This command moves the turtle vertically to the y coordinate indicated.</p>
<p>YCOR</p> <pre>home show ycor right 45 forward 100 show ycor</pre>	<p>This reporter returns the y coordinate of the current turtle.</p>

TURTLE STATE

<p>ST</p> <pre>repeat 10 [st wait 5 ht wait 5] st</pre>	<p>Stands for Show Turtle. This command makes the current turtle visible. If the turtle is already visible, this command has no effect.</p>
<p>HT</p> <pre>repeat 10 [st wait 5 ht wait 5]</pre>	<p>Stands for Hide turtle. This command makes the current turtle invisible. If the turtle was already invisible, this command has no effect.</p>
<p>SETSIZE <i>number</i></p> <pre>setsize 20 setsize 40 repeat 5 [setsize 20 wait 5 setsize 40 wait 5]</pre>	<p>This command sets the size of the turtle. The original size is 40. The minimum size for visibility is 5, and the maximum size is 160.</p>

TURTLE DRAWINGS

<p>PENDOWN PD</p> <pre>pendown forward 150</pre>	<p>This command puts the pen down for the current turtle. The turtle will draw a line as it moves.</p>
<p>PENUP PU</p> <pre>pendown forward 100 penup forward 100</pre>	<p>This command lifts the pen for the current turtle. The turtle won't draw a line as it moves.</p>
<p>PENERASE PE</p> <pre>pendown forward 125 penerase back 100</pre>	<p>This command sets the pen eraser for the current turtle. The turtle erases as it moves.</p>
<p>CLEAN</p> <pre>pd forward 150 wait 20 clean</pre>	<p>This command removes all the graphics on the current page, without moving the turtle(s). The graphics that are "cleaned" include the turtle graphics and the stamped turtles, including a large "background" stamped turtle.</p>

Appendix A - Top 50 Lynx Primitives

<p>CLEARGRAPHICS CG</p> <p>right 11 forward 9999 cg forward 9999 clean</p>	<p>This command cleans all the graphics on the current page and moves the current turtle to the centre of the page (home). The graphics that are “cleaned” include turtle graphics (lines) and stamped turtles, including a large “background” stamped turtle.</p>
<p>SETPENSIZE <i>number</i></p> <p>setpensize 5 pd forward 100 setpensize 20 forward 100</p>	<p>This command sets the turtle's pen size which determines the thickness of the lines it will draw. The original pen size is 1. The maximum pen size is 99.</p>
<p>SETCOLOUR <i>number_or_colour_name</i> SETC <i>number_or_colour_name</i></p> <p>setcolour 15 setcolour 'blue' repeat 10 [setc colour + 1 wait 5]</p>	<p>This command sets the colour of the turtle and it's pen. If the turtle has its original shape, it changes colour to show the pen colour. The input can be the name of a colour or a number. See Appendix C - Lynx colour chart.</p>

Text stuff

<p>PRINT <i>word_or_list</i> PR <i>word_or_list</i></p> <p>print 'hello' print 'there' print [hello there]</p>	<p>This command prints a word or a list in the current text box, at the position of the cursor (insertion point). The <i>word_or_list</i> is printed in the text box and the cursor drops to the next line. Use insert if you want the cursor to stay on the current line.</p>
<p>INSERT <i>word_or_list</i></p> <p>insert 'hi' insert ' there, ' ; there is a space before the "t" and after the comma wait 20 insert [how are you?]</p>	<p>This command inserts a word or a list in the current text box, at the position of the cursor (insertion point). The <i>word_or_list</i> is printed in the text box and the cursor stays on that line. Use print if you want the cursor to move to the next line after printing.</p>

<p>CLEARTEXT CT</p> <p>Type something in two different text boxes <code>text1, cleartext</code></p>	<p>This command clears the text in the current text box. The current text box is the last text box that you have created, or the last one that you have addressed with a comma (ex.: <code>text1</code>).</p>
<p>ANNOUNCE <i>word_or_list</i></p> <p><code>announce 'hello'</code> <code>announce [hello there]</code> <code>question [Your name is?]</code> Type your name in the dialog box and click OK <code>announce sentence [Good day] answer</code></p>	<p>This command displays a word or a list in an alert box. Any action taking place in your project is stopped while the alert box is showing.</p>
<p>QUESTION <i>word_or_list</i></p> <p><code>question [Your name is?]</code> Type your name in the dialog box and click OK <code>announce sentence [Good day] answer</code></p> <p><code>question [Your name is?]</code> Type KIM in the dialog box click OK <code>if answer = 'KIM' [say sentence 'Hi' answer]</code></p>	<p>This primitive displays a dialog box containing the question (<i><code>word_or_list</code></i>). After the question is answered, use the primitive answer to process what was typed in the dialog box.</p>
<p>ANSWER</p> <p>See the example for <code>question</code> above.</p>	<p>This reporter returns the last answer that was typed in a <code>question</code> dialog box as a word. The value returned by this primitive remains unchanged until a new <code>question</code> is displayed and answered. If you close the <code>question</code> dialog box using the Cancel button, answer returns an empty list. Any action taking place in your project is stopped while the dialog box is showing.</p>
<p>SHOW <i>word_or_list</i></p> <p><code>show 'Hello'</code> <code>show [Hello there]</code> (Lynx displays Hello there without the brackets) <code>show xcor</code> <code>show pos</code></p>	<p>This command prints the <i><code>word_or_list</code></i> in the Command Centre. The quotation marks and square brackets will not be printed. This command is useful when you want to print information without having to create a text box.</p>

Appendix A - Top 50 Lynx Primitives

SAY word_or_list

```
say 'Hello'  
say [This is it!]
```

Type something in a text box, then type this in the Command Centre:

```
say text1
```

This command makes your computer speak the *word_or_list*. This is a nice way to get the computer to say aloud what you have typed in a text box.

Conditionals and other stuff

IF
condition
list_of_instructions

```
home  
if heading = 0 [ht wait 5 st]
```

If instructions are often used to stop recursive procedures. Create this procedure:

```
to spiral :length  
if :length > 100 [stop]  
forward :length  
right 90  
spiral :length + 10  
end
```

Now type this in the Command Centre:

```
spiral 0
```

Without the *if* instruction, this spiral would grow forever.

This command or reporter runs the condition. If the condition reports *true*, it executes the *list_of_instructions*. If the condition is *false*, nothing happens.

<p>IFELSE <i>condition</i> <i>list_to_run_if_true</i> <i>list_to_run_if_false</i></p> <p>(Put a turtle on the page and two shapes in the clipart pane) ifelse heading = 0 [setshape 1] [setshape 2] The turtle takes shape 1 if it is heading North; otherwise it takes shape 2.</p> <pre>to getage ifelse score = 10 [page2] [announce 'keep playing'] end</pre> <p>If the value in the text box named score is zero, Lynx displays the page2. Otherwise, it displays an alert box.</p>	<p>This command or reporter runs the condition.</p> <p>If the condition reports true, it executes the <i>list_to_run_if_true</i> (the instructions in the 1st set of brackets).</p> <p>If the condition reports false, it executes the <i>list_to_run_if_false</i> (the instructions in the 2nd set of brackets)</p>
<p>REPEAT <i>number</i> <i>list_of_instructions</i></p> <pre>repeat 4 [forward 125 right 90 wait 5] repeat 5 [ht wait 5 st wait 5] repeat 10 [setsize 20 wait 2 setsize 40 wait 2] cleartext repeat 5 [print 'hi' wait 10]</pre>	<p>This command runs the <i>list_of_instructions</i> the number of times indicated.</p> <p>The instructions inside the square brackets will be repeated the number of times specified.</p>
<p>WAIT <i>number</i></p> <pre>home right 90 repeat 4 [forward 10 wait 10] repeat 10 [setsize 20 wait 2 setsize 40 wait 2] cleartext repeat 5 [print 'hello' cleartext wait 10]</pre>	<p>This command creates a pause in the execution of instructions. The duration is in tenths of a second: wait 10 means wait 1 second. wait 5 means wait 1/2 second</p>

Appendix A - Top 50 Lynx Primitives

FOREVER *list_of_instructions*

Create the **Move** procedure then type this in the Command Centre:

forever [move].

```
5 ▾ to move
6   forward random 100
7   right random 360
8   wait 10
9   end
```

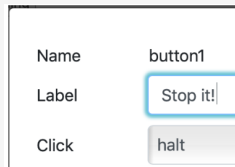
This command runs the *list_of_instructions* inside the square brackets repeatedly... forever. Use the **Stopall** icon, to the immediate left of the Command Centre, to stop the action, or use the command **stopall**.

STOPALL

Create a procedure such as the one below. Create a button that runs this procedure.



```
My project
Procedures
1 ▾ to halt
2   stopall
3   end
```



Name	button1
Label	Stop it!
Click	halt

Type this in the Command Centre.

forever [forward 1 wait 1]

Click on the button **Stop it!**

This command stops all running procedures and processes, including turtles and buttons. You can create a procedure that uses **stopall** and then place that procedure inside a button or turtle. You can also type **stopall** in the Command Centre.

EVERYONE

list_of_instructions

Put two or more turtles on a page then type this in the Command Centre:

everyone [forward 50]

They seem to move at the same time, but...

everyone [forward 50 wait 10]

Now they move clearly one after the other.

everyone [setsize 75 wait 10]

This command makes every turtle on the page run the *list_of_instructions* one at a time, until all the turtles have done it.

If you want them to move concurrently (at the same time) use the **talkto** (**tto**) command.

tto [t1 t2] forward 50 wait 10

```
ASK turtle_name
list_of_instructions
ASK list_of_turtle_names
list_of_instructions
ASK text_box_name
list_of_instructions
ASK list_of_text_box_names
list_of_instructions
```

```
ask 't1' [forward 50]
ask [t1 t2] [forward 50 wait 10]
Notice that t2 moves only AFTER t1
has completed the forward and the
wait.
```

```
t1, forward 50
(t1 does it)
forward 50
(t1 is still listening)
ask 't2' [forward 50]
(now t2 does it)
forward 50
(But if you don't use "ask", t1 is still
listening and will go forward 50)
```

```
TALKTO
turtle_or_list_of_turtles
TTO
turtle_or_list_of_turtles
TALKTO text_box_name
TTO text_box_name
```

```
talkto 't1'
forward 100
talkto [t1 t2]
forward 100
t2,
back 100
```

This command asks the turtle (or the turtles) to run the *list_of_instructions*. If you ask a *list_of_turtles* to run the instructions, the first turtle in the list will run the instructions, then the next turtle will do the same only after the first turtle is done, and so on.

This command does not change who is the current turtle. The current turtle is the last turtle that you created, or the last turtle that you clicked on, or the last turtle that you addressed using the comma (ex.: t2,). The same rules apply to text boxes (see the examples below). Also used with *everyone* and *talkto*.

This command makes the turtle(s) or text box current. This is the only way of making many turtles do the same thing at the same time.

You must use the *talkto* method when you wish to talk to (address) more than one turtle.

For a single turtle or text box, you can also use the *comma* feature: the name of a turtle or a text box followed by a comma is equivalent to a *talkto* instruction. The *comma* method works for one turtle or one text box.

Appendix A - Top 50 Lynx Primitives

CLICKON

Put 2 turtles on the page with their pen down. Space them apart. Create the **Circle** procedure and put the procedure in the On Click instruction inside both turtles.

```
1 to circle
2 repeat 36 [fd 10 rt 10]
3 end
```

Name	<input type="text" value="t1"/>
Xcor	<input type="text" value="0"/>
Click	<input type="button" value="circle"/>

Then type this in the Command Centre:

`everyone [clickon]`

This command simulates a click on the current turtle.

If a turtle is programmed to react to a mouse click, using the command `clickon` is the same as actually clicking on that turtle.

CLICKOFF

Click on the turtle and wait for a while as it wanders around. Then type this in the Command Centre:

`clickoff`

```
1 to wander
2 forever [move]
3 end
4
5 to move
6 forward random 100
7 right random 360
8 wait 10
9 end
```

Name	<input type="text" value="t1"/>
Xcor	<input type="text" value="-121"/>
Click	<input type="button" value="wander"/>

This command simulates an “unclick” on the current turtle.

If a turtle is programmed to react to a mouse click and is currently running its “on click” action, `clickoff` is the same as actually clicking off that turtle.

If the turtle is not running its “click” action, then `clickoff` does nothing.

<p>RANDOM number</p> <p>Make sure you have a turtle and text box on the page.</p> <pre>right random 360 forward random 100 setcolour random 140 setsize random 120 print random 6</pre>	<p>This reporter returns a random non-negative integer (including 0) that is one number less than the number indicated.</p>
<p>PICK word_or_list</p> <pre>show pick 'Hello' show pick [Kim Sam Leo]</pre> <p>Assuming you have three turtles on the page:</p> <pre>ask pick [t1 t2 t3] forward 50</pre>	<p>This reporter returns one random character of a word, or one random word of a list.</p>

Appendix B - Common coding errors

SPELLING MISTAKES!

Lynx understands **forward**, **right** and **setsize** for example.

Lynx will give you an Error Message if you type **forwrdr**, **rite** and **stesize**, for example.

Always check your spelling if you get an error message.

SPACES IN OBJECT NAMES

When naming turtles, text boxes, sliders and pages be sure you **don't** use spaces before, inside, or after the name. If a text box is called "**My text**" with a space between MY and TEXT, the instructions **my text**, **setmy text** and **say my text** will not work:

```
I don't know how to my
```

If a text box is called "**Mytext**", the instructions **mytext**, **setmytext** and **say mytext** will work

If a text box is called "**My_text**", the instructions **my_text**, **setmy_text** and **say my_text** will work

SPACES IN PROCEDURE NAMES

When creating procedures, be sure to use single words, with **no spaces**. If you created a procedure called "big square", you won't be able to run it. Lynx will display an error message:

```
I don't know how to big
```

My.square, **my_square** and **MySquare** are all names that work.

PROCEDURES THAT DON'T WORK

There are many rules to follow when creating procedures.

```
to square
```

```
; This draws a square
```

```
repeat 4 [fd 50 rt 90]
```

```
end
```

THE WORD **TO**, A SPACE, AND ONE WORD
COMMENTS START WITH A SEMICOLON ";"

INSTRUCTIONS, AS MANY AS YOU WANT

THE WORD **END**, BY ITSELF ON THE LAST LINE.

Also, remember that procedure names must be new to Lynx. You cannot use a Lynx primitive (built-in vocabulary) or other procedures you have already created for this project.

If you forget the word **end** at the end of a procedure, all the procedures that follow that procedure will not work.

WRONG TYPE OF QUOTATION MARK

Sometimes, when copying and pasting code from other sources, you may end up with typographical (curly or slanted or French) quotation marks such as these: `'yellow'` or `"yellow` or `«yellow`. These will **not** work. Make sure you use straight quotation marks, like `'yellow'` or `"yellow`. If you get the wrong type of quotation mark, just erase and retype the quote in the Lynx editor.

NOT TALKING TO THE RIGHT TURTLE OR TEXT BOX

When you run a turtle command, the turtle that will execute it is the current turtle. The current turtle is:

- the last turtle that you have created, or
- the last turtle that you have clicked on, or
- the last turtle that you called using the “comma” method (`paddle,`) or the `talkto` command.

If a procedure is meant to be executed by a specific turtle, simply state its name followed by a comma (no space) at the beginning:

```
to bounce  
ball,  
setheading...  
end
```

The same rules apply to text boxes.

BUTTON NOT WORKING

Did you forget to “link” the button to a procedure? If yes, do a right-click on the button and, in the On Click field, choose the procedure you want to run when the button is clicked.

If you ever change the name of the procedure, the button will stop working. In that case, reopen the button’s dialog box and relink it to the actual procedure, found in the **On click** drop down menu.

Appendix B - Common coding errors

NOT PAYING ATTENTION TO ERROR MESSAGES

Error messages are important. Extremely important. They always contain a clue about what the problem is, and *generally, on which line the error is*, in the case of procedures.

I don't know how to announce in winner on line 43

You tried to execute a word that is not a Lynx primitive nor a procedure you have created. Sometimes this is just due to a typo (`announce` instead of `announce`). Sometimes it is because you forgot a typographical element (`print hello` instead of `print 'hello'` - in the first case `'hello'` is just a word to be printed. In the second case, `hello` looks like a command to execute).

I don't know what to do with...

Some primitives “report” a value. In these cases, you have to tell Lynx what to do with the value. `Heading`, for example, reports the heading of the current turtle. If you execute just the word `heading`, you get an error message. If you execute `show heading`, or `print heading`, or `setheading heading + 90`, the value reported by `heading` will be used by `show`, `print`, or `setheading` - this is fine.

No ... found for ...

You tried to execute a turtle command and there is no turtle on the page. Or a text command, and there is no text box on the page.

... doesn't like ... as input

Certain primitives like only certain types of inputs. `Print` can print anything (`print 'hello' print 44`). `Setheading` likes only numbers. `Forward` likes numbers not greater than 9999.

UNWANTED SPACE

So many times in this book, we told you “one word, no space”. This is true for the names of pages, turtles, text boxes, sliders, sounds... That also means that there are no spaces when you use these words.

```
show text1
t1,
ask 't1'
setslider1 slider1 + 10
```

These are all good. There is no space in `text1`.

Remember: `text1` and the other examples above are the names of objects, **no space needed**.

`forward 50` is a primitive with an input, **it needs a space**.

Appendix C - Lynx colour chart

Lynx turtles can be any colour. Colours in Lynx are numbered like this:

0	10	20	30	40	50	60	70	80	90	100	110	120	130
1	11	21	31	41	51	61	71	81	91	101	111	121	131
2	12	22	32	42	52	62	72	82	92	102	112	122	132
3	13	23	33	43	53	63	73	83	93	103	113	123	133
4	14	24	34	44	54	64	74	84	94	104	114	124	134
5	15	25	35	45	55	65	75	85	95	105	115	125	135
6	16	26	36	46	56	66	76	86	96	106	116	126	136
7	17	27	37	47	57	67	77	87	97	107	117	127	137
8	18	28	38	48	58	68	78	88	98	108	118	128	138
9	19	29	39	49	59	69	79	89	99	109	119	129	139

To set a turtle or background to a particular colour, run the `setcolour` (`setc` for short) or `setbg` command, followed by the chosen colour number:

`setc 127` or `setbg 'green'`

	BLACK 9
	WHITE 0
	GREY 5
	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

Colour numbers go in tens. Each set of ten contains the shades of a particular colour. For example, shades of yellow are from 40 to 49 and shades of orange — from 20 to 29. The smaller the number in a set of ten, the lighter the shade, the bigger the number — the darker the shade.

16 colours are considered “basic”. They have not only numbers, but also names. These colours and their names are shown on the left. For these colours you can type `setc colour_name` in the Command Centre instead of just `setc colour_number`

For example:
`setc 'violet'` and `setc 115`
do the same thing.

If you want your turtle to be one shade darker than the VIOLET 115 colour, you need to run an instruction such as:

```
setc 116 No name is available for that shade.
```

In short: 110 to 119 is the range of violet shades, 115 is the middle point and corresponds to:

```
setc 'violet'
```

In that set of ten, anything above 115 is a darker shade, anything below 115 is a lighter shade.

And the same is true for all the other sets of shades.

When to use names and when to use numbers? Names only exist for 16 basic shades. Using names makes your code more “human”. But when you do calculations, use numbers, like this:

```
repeat 140 [setcolour colour + 1 stamp forward 20]
```

Appendix D - Finding things

Free Resources on the web

At the time of printing of this book (September 2020) the sites listed below were *up-to-date* and had free resources. We cannot guarantee this will still be true when you read this coding book. Please verify copyright rules *before* using resources from these sites.

SITES TO FIND CLIPART WITH TRANSPARENCY

PNG and JPEG are 2 file formats that work with Lynx. Take a tour of these two sites:

<http://pixabay.com> and <https://www.freepngimg.com>

Search their databases for clipart with a transparent surrounding.

Search a term (ex.: car), pick one, click on Choose a size. 200 x 200 is generally sufficient for a turtle to use as a shape.

See next page about the importance of transparency for PNG files.

SITE TO FIND BACKGROUND PHOTOS

<https://unsplash.com>

Search their database for background photos (wallpaper type).

Some images may be quite large, but Lynx resizes the image to the project size when you import it.

SITES TO RECORD YOUR VOICE

Your computer or phone may already have a voice recording tool. If not, here are two options:

<https://vocaroo.com> or <https://www.audacityteam.org/>

Save your recording in WAV or MP3 file format and don't make the recording too long.

SITES TO FIND SOUND EFFECTS

BBC Sound Effects has a searchable database of WAV files.

<http://bbcfx.acropolis.org.uk>

Click on **Duration** to sort by length. Use **short** audio clips with Lynx.

A second option is below. Read the Attribution rules carefully.

<http://soundbible.com>

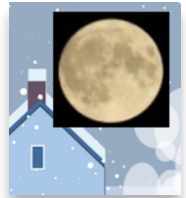
Web site where you can edit your images

At the site below there is a free editor that will allow you to modify images. Be aware that there is a free mode, with ads, and also a paid version.

<https://pixlr.com/>

About transparency of PNG files

JPG files, like photos, simply do not support transparency. The same applies to screen captures, even if you capture it in PNG. These are always **rectangular objects**. For example, here is a picture of the moon, screen captured on a black background used as Lynx clipart and turtle shape. The image on the right is what you will see - probably not what you want.



In order to get a nice “cut out” image of the moon, or any object, you have two options. Either you:

- Edit the screen capture or jpg file to **erase** the surrounding (can be easy or difficult), and **save it as PNG with transparency**, or
- Search for true PNG images with transparency. They often have a grey or checkered surrounding indicating the transparent area, like this image of a tree.



You need to **download** the file to your computer or your personal work space in the cloud (Google Drive for example), and **import** it to an empty clipart box. Click anywhere in the empty box and a “+” sign appears.

Appendix E - FAQ

For an up-to-date version of this FAQ, please choose **Help** -> **FAQ** on the Lynx home page.

Using Lynx on Chromebook

When using a Chromebook computer, we assume that all your Lynx files are in the Lynx cloud (this is the same for any Lynx user), and all the external files that you will try to import into Lynx (clipart, sounds, voice recordings, screen captures) *will come from your Google drive*. Screen captures and other files *may* be in your local **Downloads** folder, but for the sake of simplicity, we assume that you have brought these into your Google drive as well. In any case, you should check both **Google Drive** and **Downloads** for the clipart, sounds and screen captures you want to use.

Your **Google Drive** is in the left column of any Browser window or Open dialog box.

Using Lynx on iPad

When using Lynx on an iPad, be aware that a “click” in any instruction you read in this book, becomes a “touch” on an iPad. Also, since there is no “right-click” on an iPad, you must use the **Project tree** to open the dialog box of a turtle, a text box, a button, etc. Click on the **Project tree** icon as shown here, click on the **small triangle** to expand a page and see its contents, click on an object for example a Turtle or Button, and finally click on **Edit** to open its dialog box, or **Delete** to permanently delete the object.



Screen captures are much harder to do, so we suggest that you find pre-made clipart or create your clipart on a regular computer.

How to do a screen capture to use as a Preview image or as a clipart

As mentioned above, screen capture will produce rectangular clipart, with *no* transparent surrounding. You have to edit the screen capture to erase the surrounding yourself and save the image as a PNG file.

IN WINDOWS:

Launch **Snipping Tool** in the **Windows Accessories**. In **Mode**, choose **Rectangular Snip**. Use the **Rectangular** mode. Select a region on your screen, that is tight around the image you wish to copy. Copy (**Ctrl+C**) the resulting clipart from **Snipping Tool**. Now click in an empty spot in the Lynx Clipart Pane and paste (**Ctrl+V**).

If you want to use the image as a Preview for your project, you first have to save it as a PNG file, then navigate to it when selecting the Preview image.



ON A MAC:

Pre-Mojave, macOS 10.13 or lower: Press **Command-Shift-4**. You get a crosshair pointer. Select a region on your screen, that is tight around the image you wish to copy. The image is saved on your desktop. Double-click on that file to open it in **Preview**, and copy it from there. Click in an empty spot in your Clipart Pane and paste the clipboard (**Command-V**).

Mojave, Catalina, macOS 10.14 or higher: Press **Command-Shift-5**. Select **Clipboard** in the **Options**. Select a region on your screen, that is tight around the image you wish to use and copy it (**Command-C**). Now click in an empty spot in your Lynx Clipart Pane and paste the clipboard (**Command-V**).

You can save your screen capture on your computer, open it in **Preview** and use the tool **Instant Alpha** to determine which colour should be transparent. It is better, for that, to make screen captures on a solid white background.

ON A CHROMEBOOK:

Press **Ctrl-Shift-SwitchWindows** () simultaneously and select the region you wish to capture. This will create a file in the **My Files> Downloads** folder. Then use the  method to load the image as clipart in the Clipart Pane or in **Project Properties**, choose **Select a Preview image**

Appendix F - Sharing

There are many reasons to share your projects. In this book, there is an interactive card (Thank you essential workers, Mother's day, Happy Birthday...), secret code and flashy art projects and two games! As a proud coder, you might just want to share all projects that you create.

Lynx makes sharing easy. You can share a project with **some friends** of your choosing (text message, email), or anybody who follows you on Facebook or Twitter, or with **every Lynx user in Canada**.

To read more about project management and sharing, see the documents **Sharing your projects**: go to the www.lynxcoding.club, click on **Help**, then **User Guides**.

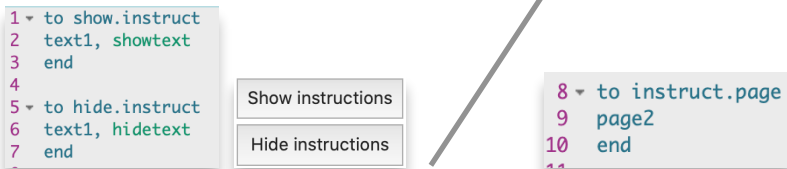
Before sharing

PREPARE YOUR PROJECT

You must name your project and save it before it can be shared. You should be on Page1 (or the starting page of your project) when you save your project. Very important: Do NOT include in your project any personal info that can identify you or say where you live. No photo of yourself or full name.

Make sure your project is usable in **Play mode**. You may need

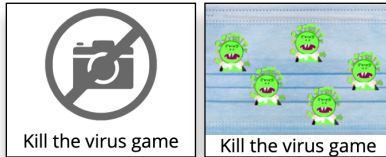
- a button or a clickable turtle to **start** the action, if any;
- a button to **stop** the action, if you use a **forever** action without a stop rule;
- a text box with instructions, if needed. The text box can be visible at all times, or only appear when you click an "Instructions" button (use the commands **showtext** and **hidetext**), or if there are lots of instructions, the **Instructions button** could lead to an **Instructions page**;



- a **startup** procedure that has some instructions to reset things in your project, and open a specific page. See this feature in *Project 3 - Interactive Thank You card*.

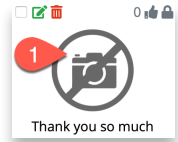
CREATE A PREVIEW IMAGE

Creating a **Preview image** is always a good idea, even when you keep your project private. When your project has a **Preview image**, it is recognizable in your list of projects in the Lynx cloud. Here's your project without and with a **Preview image**:

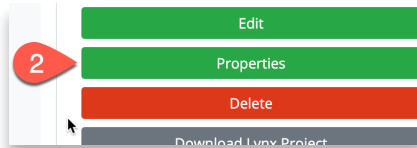


Creating a **Preview image** for your project is not mandatory for sharing your project. *However, it is highly recommended*, as it attracts more viewers when you share it. Before you can select a **Preview image** for your project, you must first create the image file. For example, save a screen capture of the nicest page in your project. See *Appendix E - FAQ* for more information about making screen captures.

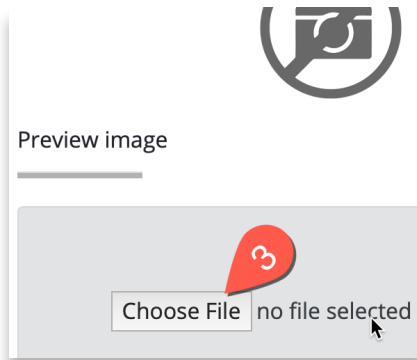
Now go to **My Projects**, where your projects reside in the Lynx cloud. Click on your project once (1), to open it in **Play mode**.



In **Play mode**, click on **Properties** (2).



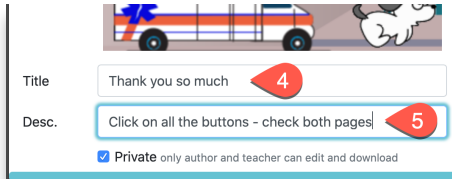
On the **Properties** page, click on **Choose File** (3), under the default preview image:



Appendix F - Sharing

Navigate to the file that you have prepared for the **Preview image**. It will **not** appear immediately on the **Properties** page, but it will appear when you click on **Save** in a moment.

Type a **Title** (4), and maybe include some instructions in the **Desc** (description) field (5) or write some info about the project.



Title Thank you so much 4

Desc. Click on all the buttons - check both pages 5

Private only author and teacher can edit and download

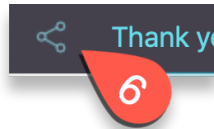
Click on **Save** when you are done. Your project is now recognizable on the page **My Projects**.

You are all set for sharing now.

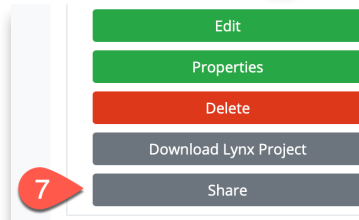


Start sharing

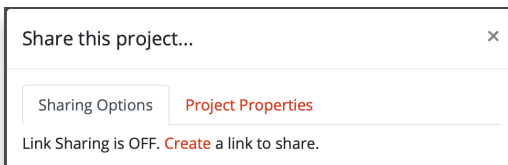
There are two places to launch this sharing process: from **within the Lynx editor**: click on this icon (6) at the top-left of the page...



or look at your project in **Play mode** (click on your project, on the **My Projects** page), and click on **Share** (7):



Both methods will open the **Sharing** dialog box.



Share this project... X

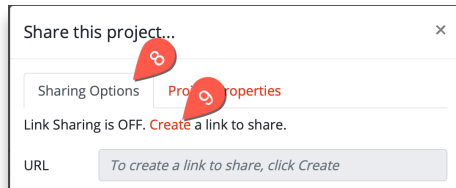
Sharing Options Project Properties

Link Sharing is OFF. Create a link to share.

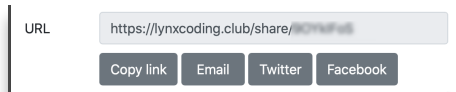
Sharing with some people you know

If your project does not have a **Preview image**, a **Title** and a **Description**, you can click on the **Properties** tab now to do that, following the instructions in the section just above.

If you have done all this, stay on the **Sharing Options** tab (8) and click on **Create** (9). Lynx will create a URL that represents your project in the Lynx cloud.



The four buttons just below the URL allow you to:



- **COPY LINK:** Choose this option so you can paste the link anywhere, for example in an e-mail or in a messaging application on your computer.
- **E-MAIL:** This button opens a new e-mail message, with the link already in the body of the message. Just add the other details and send!
- **TWITTER / FACEBOOK:** Remember to choose the friends you would like to share your project with otherwise everyone will see your project. Your friends will see the **Preview** of your project and then click / press on it to be taken to lynxcoding.club

Appendix F - Sharing

Sharing with all (private vs public)

By default, the projects that you create and save are **Private**. A private project only appears in your **My Projects** page. **Only you can see and modify this project**, unless you share it.

If you uncheck the box labelled **Private** in the **Project properties**, the project becomes **public**. It will also appear on the **All Projects** page. When a project is **public**, other members of the Lynx community will be able to see it, make changes to its code and other content and use it as a starting point for their own projects. They will be able to make changes to your original project, but they can only save the modified project as their own, in their own Lynx cloud. *They cannot change your personal project even if it appears in a public space.*

IN SHORT

KEEPING YOUR PROJECT PRIVATE: Nobody will see or edit your project, but if you choose to share it by e-mail, text message or social media, people specifically selected by you will see your project in **Play mode**. They will not see the Lynx editor, the tools, the Command Centre and the Procedure and Clipart Panes.

If you *uncheck* Private in the Sharing dialog box, then the following is also true:

MAKING YOUR PROJECT PUBLIC (UNCHECK PRIVATE): The project will appear on the **All Projects** page. Anyone visiting the Lynx home page will see it, and will be able to make changes to it. If they have a Lynx account, they will be able to save the modified project as their own. **YOUR project will stay exactly the way you made it.** If they don't have a Lynx account, they will still be able to see, open and edit your project, but they won't be able to save the modified project.

HOW TO EMBED YOUR PROJECT IN A WEBSITE OR BLOG

You can also embed your project in a Web site or blog. Copy the code provided in the field **Embed on your site**, in the lower part of the **Sharing** dialog box.

MORE INFO ABOUT SHARING INCLUDING HOW TO EMBED

To read more about project management and sharing, see the documents *Sharing your projects*: go to the www.lynxcoding.club, click on **Help**, then **User Guides**.

ADDITIONAL RESOURCES

On the Home page of lynxcoding.club, you will see **Help** on the right side. Click on **User Guides** and you will find more projects and reference materials.

Also, there are wonderful resources, including short **How To** videos and long recorded webinars at:
sites.google.com/view/lynxcoding-org



What is Lynx?

Lynx is a designed-in-Canada coding app from a team with decades of experience making coding apps for K-12 students. All the core designers have worked with the late Dr. Seymour Papert of the MIT Media Lab, the father of educational computing.

Lynx is a cloud-based programming environment for learners to create sophisticated projects. This text-based language supports computational thinking without the strict rules of other coding languages, such as JavaScript.

Think of Lynx as the next step after using block-based coding tools!

You can easily share your Lynx Projects with your friends.

Thanks to a Federal Government CanCode grant from ISED Canada, the Lynx coding app is free to use by Canadian residents.

We have successfully integrated Lynx into our culturally responsive mathematics project in a number of Grade 3 to Grade 8 classrooms in Ontario. The dynamic nature of Lynx allowed students to explore the different interpretations of mathematical concepts inherent in the work, which then allowed us to more fully analyze students' mathematical understanding. The students loved writing code and in doing so they had numerous opportunities for problem solving.

*Ruth Beatty, PhD
Faculty of Education
Lakehead University*



Not sure if this book is for you or your child? You can watch a two-minute movie called *The Missing Link* at lynxcoding.club which gives you an idea of what you can do with Lynx.

Canadians! Publish a Lynx project before March 31, 2021 and fill out a form to enter a random draw for a chance to win a Windows laptop, Chromebook, or an Arduino robotics kit. Contest not valid in Quebec.

For more info, go here: bit.ly/lynxcodingbook

What you need to run Lynx:

An Internet connection and a Windows computer, a Mac or a Chromebook. Lynx will run on iPads and Android tablets, though the tablet screen size may be challenging—and a keyboard is recommended!