

LYNN

TM



INTERACTIVE STORY PROJECT CODERS' GUIDE



Introduction

Project Goal

Storytelling is an important part of communicating, whether to entertain, illustrate a point, or move people to action. In this project, you'll tell a story about a trip or adventure you had – whether to another country, another city, or to your friend's house. As you do so, you'll learn how to add the basic components of an engaging, interactive, multimedia story.

Plan

Before you start:

- Decide on the trip or adventure that you'll describe. Who were you with? What did you do that you particularly enjoyed or that was most memorable in any other way? What are the key details you'll include?
- If you have photos of a place you visited, make sure you have access to them on your computer. You may want to add them as backgrounds. If you don't have any, think about where you can find some images on the internet that you'd like to use (respect copyright, please).
- Think of how you'll use Lynx tools to share what happened on your trip, whether with graphics, animation, sounds, music or text. How will you bring your story to life?
- Look at the Terry Fox sample project for ideas on how to start your project and see what is possible.
- It is important to remember that Lynx **does not automatically save your work**, so you must remember to **save your projects often!** More details on how to save are included in *Lesson 1, Step 5* below.

Here's what you'll do in this project:

- Learn the basic commands in Lynx to control turtle (sprite) movement and create your own programs.
- Add interactive objects, such as buttons and interactive turtles to trigger actions and create a navigation system.
- Create multiple sprites doing a variety of actions and more complex animations.
- Add text, music and sound for a true multimedia experience.

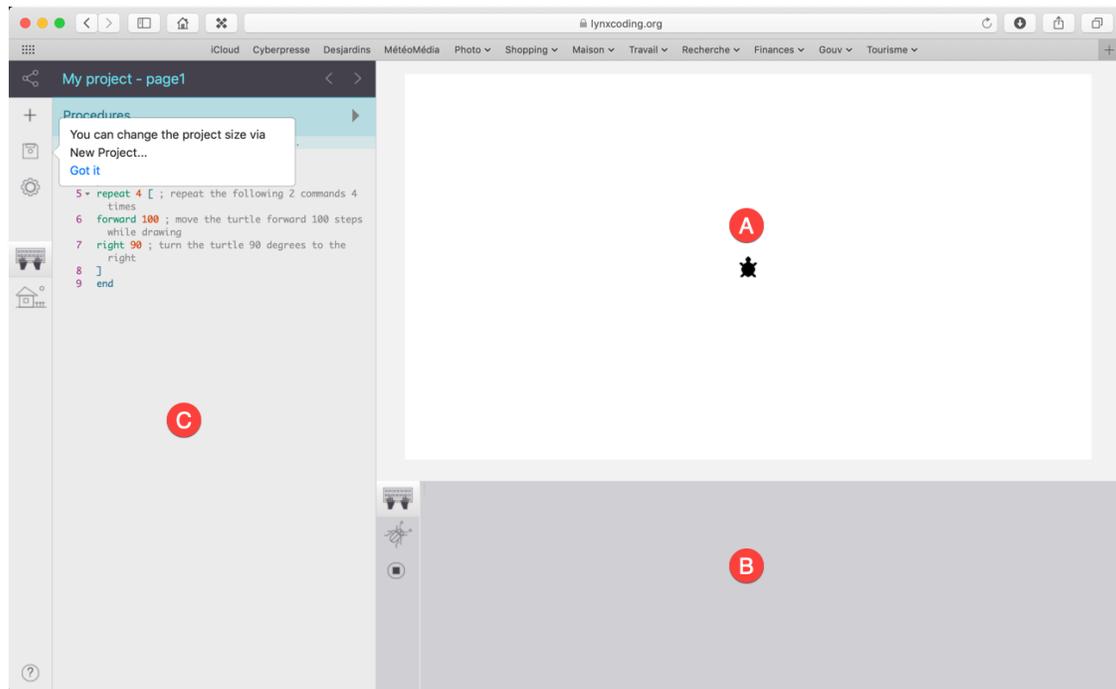
Lesson 1 – Start Your Story

Step 1: Introducing Lynx

You should have already created a Lynx account. (If you haven't, talk to your teacher first then look in the [User Guides](#) section on the Lynx Home Page and refer to *How to Create a Lynx account*).

On the Lynx home page (www.lynxcoding.org), click on **Create a Lynx Project**.

The screen looks like this:



The largest area is your **Work area (A)** where you'll be doing your work (surprise!). A turtle is sitting in the middle.

The area below it is the **Command Centre (B)** where you'll be typing code to test in the work area.

The area to the left is the **Procedures Pane (C)**, where you'll write your coding programs (a sample program is already included). In this area you can also access various menus and the **Clipart Pane**.

Now that you know the layout, it's time to get to work!

Lesson 1 – Start Your Story

Step 2: Create a Background

In Lynx, sprites are called turtles. And they look like turtles, at least to start. They are used in a number of ways. For example, use them to create your page background.

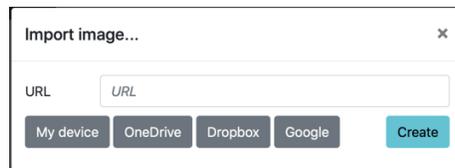
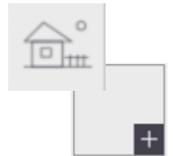
In the "+" menu, select **Sample Clipart**. The Clipart Pane opens, populated with sample art. It includes both small images that can be used as shapes for the turtle and scenes that can be used as backgrounds. There are also a number of blank spaces you can use to add your own graphics.

There are two ways to find a background that fits the beginning of your trip or adventure.

1. **Method 1:** Use our sample clipart
2. **Method 2:** Add your own clipart
 - Use any paint program and draw your own images. Save your art as a JPG or PNG (PNG supports transparency around your objects if you need it).
 - Download images from any image search web site or a photo you've taken. Respect copyright issues!
 - Use the camera on your smartphone or tablet.

To add your image to the Clipart Panel, either:

- Copy the clipart. Then click on the **House** icon to open the Clipart Pane. Click on an empty box to reveal a "+" sign. Next press **Command-V** on a Mac, or **Ctrl-V** on other computers to paste your image in the box.
- Click on an **empty box** in the Clipart Pane and click on the "+" that appears. Then use the dialog box to locate a clipart file on your device or online. Finish by clicking the **Create** button.



For both methods, follow these steps to create a background:

1. Put the mouse over the background image in the Clipart Pane to see the shape number.



Lesson 1 – Start Your Story

2. In the Command Centre type:

```
setshape 18
```

Use whatever number your shape is. Press **Enter** to run the command.

The turtle's shape changes to the shape you chose.

3. In the Command Centre, type:

```
stamp
```

Press **Enter**. This stamps the turtle's shape on the background.

Once stamped, this background image cannot move... but, the turtle can! Click on the turtle and drag it a little. You'll see that the "live" turtle can move, but the background layer is static.



Here is another handy command just in case you change your mind about the background you stamped:

```
cg
```

Or `cleargraphics`

The stamped background disappears but your moveable turtle will still have your chosen background.

To change your turtle back to its original shape, type this in the Command Centre:

```
setsh 0
```

0 is the turtle's shape number

Step 3: Add Background Features

You can stamp other, smaller turtle shapes on the background. For instance, if you want to add additional trees to a scene, find a tree shape (or import one), set the turtle to that shape, adjust the turtle's size, and stamp the shape.

For example:

```
setsh 35
```

`Setsh` is short for `setshape`

```
setsize 20
```

```
stamp
```

Once stamped, the image becomes part of the background and doesn't move (although the turtle, on the top layer, can be moved).

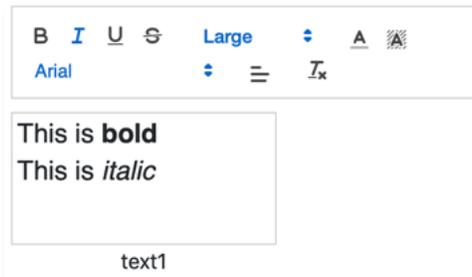
Add some additional images to your background.

Lesson 1 – Start Your Story

Step 4: Add Text

To start writing the text component of your story, you first need to add a text box.

Select **Text** in the "+" menu. A text box named `text1` appears in the work area.



If you can't see all your text, drag the bottom right corner of the text box to make it larger. If you can't see the formatting tools, click on the name of the text box and drag the box lower on the screen.

Click in the text box and add some text. A set of formatting tools appears above the text box. If you have selected text, the formatting applies to it. Otherwise, it sets the format for the text you are about to type at the insertion point.

You can place the text box wherever you want it on the screen.

Right-click on the text box to hide the text box name or make the box transparent. If you need to move the text box, make the text box opaque again, drag the box, and then make it transparent.

If you make the text box invisible by mistake, in the Command Centre type:

```
showtext
```

Step 5: Save your Project!

Lynx does not automatically save, so you must remember to save your projects regularly.

In the top left corner of Lynx, you will see the name of the project, followed by the name of the current page.



Click on that field and choose a meaningful name for your project like: **Camp 2019**.

Next, click on the **File (disk)** icon and select **Save**.

Lynx will save your work area and Procedures Pane (it does not save the Command Centre).

Save often! It's no fun to work on a project and then lose it by forgetting to save.

Lesson 2 - Add Story Features

Step 1: The Turtle

Since turtles are a key component in bringing your story to life, it's important to know how to control them. You are always looking at the turtle from a "bird's eye view".

You can control a turtle in two ways:

- By dragging it with your mouse.
- By using commands that you type in the Command Centre.

Dragging is a convenient way to move the turtle when testing or stamping something, but commands let you precisely control the turtle movements, distance and direction.

In the Command Centre, type:

```
forward 100
```

Press **Enter** after each
command line

Note: If you have enabled Auto-complete in your Settings, Lynx will suggest primitives (commands) that match the 1st letters you type.

The turtle moves forward 100 pixels.

Here are several more commands to try. Each of these commands needs an input. For these commands, the inputs are numbers. Feel free to change the inputs.

```
right 80
```

```
BACK 150
```

```
left 145
```

Commands are not case-sensitive.

Remember to click **Enter** after each line. Did the turtle move or change heading?

The input for right and left is in degrees. If a turtle turns all the way around, it turns 360 degrees, so if you type:

```
rt 360
```

....it looks like the turtle hasn't moved at all.

Each of these commands also has a short form: `fd`, `rt`, `bk`, `lt`.

```
fd 100
```

```
rt 30
```

```
bk 50 lt 50
```

Multiple commands can be on the same line.

You can do a lot with just these four commands. Try different inputs to see what happens. What happens if the number is very large? How big is just one pixel? How wide is the work area?

Lesson 2 - Add Story Features

Step 2: Seeing Patterns

One of the key skills in coding is recognizing patterns. For example, in movement, the same forward motion is repeated again and again, step by step.

```
fd 1
fd 1
fd 1
fd 1
```

Recognizing patterns lets you write more efficient code. When you see a pattern, use the repeat command instead of writing the same lines of code over and over.

```
repeat 50 [fd 1 wait 1]
```

`Wait` creates a pause
in tenths of a second.

Is the turtle moving too quickly? Too slowly? Try different inputs with `forward` and `wait` until you find a speed you like.

Or try this:

```
glide 300 2
```

Glide takes two inputs,
distance and step size.

`Repeat` requires two inputs. The first is always a *number*, the second is always a *list of coding commands* enclosed in *square brackets*.

There are always multiple ways to do things in Lynx. Test each one and see which one you prefer.

Step 3: Automate the process

Once you have an action you like, it would be convenient to have a command to start it, for example, a `move` command. Type this:

```
move
```

You get the following message:

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

`Move` isn't a built-in command in Lynx but you can add it by writing a procedure.

In the Procedures Pane (to the left of the work area), type the following:

```
to move
```

All procedures must start with the word "to" and the procedure name. Choose a one word name, for now.

```
glide 300 1
```

All commands go here

```
end
```

A procedure must finish with the word "end" on its own line.

Lesson 2 - Add Story Features

Place your turtle on your background where you want it to start moving.

Now, in the Command Centre, type:

`move` Press **Enter**.

Did it move far enough? Fast or slow enough? Adjust your procedure until you're satisfied.

You'll see the Procedures Pane already contains the `square` procedure. Just add your `move` procedure below it.

Procedures are the programs you write. Each procedure is a new command you add to your project. Since they are commands, procedure names must be single words and cannot be a name already used by a built-in Lynx command. A procedure only works in the project in which it's created.

It's good programming practice to add comments before, after, or in the middle of a procedure to explain what it does or add additional explanatory information. To indicate something is a comment, start the comment line with a semi-colon (;).

```
11 ; Next procedure moves the target - 300 is  
    half the length of the field  
12 to move  
13 glide 300 1  
14 end
```



Did we mention that you should save your work **often**?

Lesson 3 – Animation

Step 1: Single-frame Animation

Instead of a turtle, maybe you want a car to drive across the background or a train or a cloud.

First, open the Clipart Pane and find an image you like. Remember the shape number.

Next, change the shape of the turtle:

```
setsh 11
```

 ...or whatever shape number you like.

Move the turtle to the best start position.

Now get your turtle moving. In the Command Centre, type:

```
move
```

You can also add your own clipart. Follow the instructions in *Lesson 1*. For smaller shapes, make sure you import art in PNG format.

Step 2: Debugging

Is the shape pointing to the left, but the turtle is moving to the right?

Bugs appear in all programming projects as they are being developed and even as program versions are released. Debugging is an essential programming skill (some professional programmers say it's the most interesting part of programming).

In this case, the turtle moves in whatever direction the turtle's nose is pointing. To see what that is, type:

```
setsh 0
```

Change the direction the turtle is pointing by using commands. For example:

```
rt 180
```

Or, use `setheading` (or `seth` for short). `Setheading` sets a precise, absolute heading whereas right or left pivots the turtle a certain amount *depending on which direction the turtle was originally pointing*.

The input for `setheading` corresponds to the degrees on a compass. So:

- `seth 90` always points East.
- `seth 180` always points South.
- `seth 270` always points West.

Try your animation again.

To adjust your `move` procedure for the new shape, edit the procedure in the Procedures Pane.

Lesson 3 – Animation

Step 3: Multiple Turtles

You may want to have multiple turtles moving on a page as part of your story, for example two cars driving on a road.

A page can have more than one turtle. In the "+" menu, select **Turtle**. A second turtle appears in the centre of the screen. If it is on top of the first turtle, just drag it somewhere else.

Each turtle has a name. Right-click on the turtle and in the dialog box that opens, you'll see there's information about the turtle, including its name. Take note of each turtle's name. You can change the name of the turtle too.

You may want to write a new `drive` procedure for your car (or use your `move` procedure), for example:

```
to drive
repeat 30 [fd 3 wait 1]
end
```

To get all the turtles on the page to do the same thing, type in the Command Centre:

```
everyone [drive]          Use your procedure name.
```

Turtle1 will execute `drive` then **Turtle2** will execute `drive`.

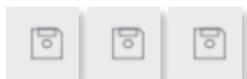
On the other hand, you may want different turtles to do different things. For example, you may want one to move slowly and the other to move fast.

Add a comment to the <code>drive</code> procedure so you know which speed it's moving, for example:	Then write another <code>drivefast</code> procedure:
<pre>to driveslow ;this is for slow driving cars repeat 30 [fd 3 wait 1] end</pre>	<pre>to drivefast ;this is for fast driving cars repeat 30 [fd 9 wait 1] end</pre>

To talk to one turtle at a time, type the turtle's name followed by a comma. For example:

```
t1, driveslow          The comma means you're addressing or "talking to" a specific turtle.
t2, drivefast
```

Always test your procedures. Adjust them to fit your design and get your cars rolling!



Did we mention that you should save your work **often**?

Lesson 4 – Build Your Story

Step 1: Add Pages

Open your [Camp 2019](#) project.

Your project can have multiple pages.

Select **Page** in the "+" menu.

A new, blank page appears. To return to page1, click on the arrow keys above the Procedures Pane. You'll see the name of the page next to the project name:



Click the arrow again and you'll see page2 appear next to the project name. Page1 and page2 are not just page names, they're also commands that take you to the appropriate page in your project.

When on [page2](#), type this in the Command Centre:

```
page1
```

You should flip back to the first page.

Go back to [page2](#). Add a background, stamped background features, and some text on [page2](#) to continue your story.

If there's no turtle on page2 select Turtle in the "+" menu.

Step 2: Navigation

You can use the arrows above the Procedures Pane to move from page to page as you're building your project, but you'll need to make the process easier for anyone who views your project once it is finished.

There are two ways to add navigation "buttons" to change pages.

Start each method the same way. In the Procedures Pane, add two new procedures:

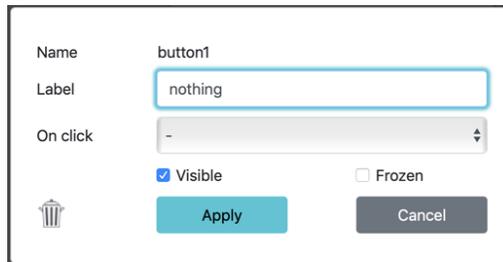
```
to 2NDpage
page2
end
to 1STpage
page1
end
```

Lesson 4 – Build Your Story

Method 1: Buttons

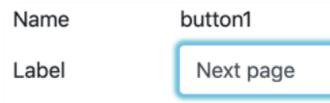
Buttons make it easy to initiate actions. To create a button:

1. Select **Button** in the "+" menu. You'll see this: 
2. Right-click on the button. You'll see:



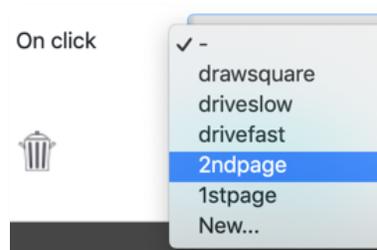
Name: button1
Label: nothing
On click: -
 Visible Frozen
Apply Cancel

3. Add a label. It can be anything, and one *or* several words. This is just a button label, not a command. Type something that lets you and your project viewers know what the button does. This example assumes your button is on [Page1](#).



Name: button1
Label: Next page

4. Click on the [down arrow](#) in the box next to [On Click](#) and select the appropriate procedure. In this case, select [2NDpage](#)



On click: -
drawsquare
driveslow
drivefast
2ndpage
1stpage
New...

5. Click [Apply](#).

If the label is not completely visible, drag the bottom, right corner of the button to make it larger. Move the button to where you want it to be. Test it. You should go to [Page2](#) now.

Now you have the start of a navigation system.

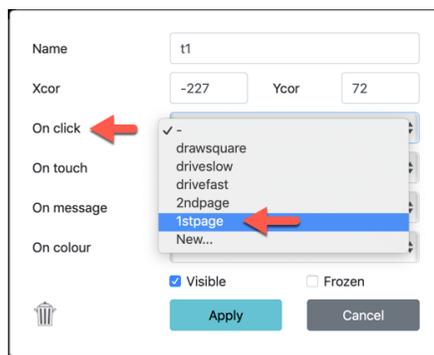
Lesson 4 – Build Your Story

Method 2: Turtles as Buttons

Turtles can also act like buttons.

Add a "navigation" turtle to the page. Set its shape to something appropriate, such as a left arrow shape found in our Sample Clipart.

1. Right-click on the turtle.
2. Click on the down arrow in the box next to **On click** and select the appropriate procedure. In this case, select **1stpage**.



3. Click **Apply**.

Click on the turtle to test it. You should be back on **Page1**.

Now do the same on **Page1** of your story.

Both buttons and "turtles as buttons" can be used to initiate any actions or run any procedures in your story. Be creative!

Step 3: Turtles as Buttons

Use turtles to initiate other actions in your story.

For example,

- Right-click on the turtle on **page1**.
- Click the down arrow in the box next to **On click** and select **move**.
- Click **Apply**.
- Click on the turtle to test it. Does it move?

If you want your users to know how to do this, add some instructions.

Lesson 4 – Build Your Story

There are a number of debugging tools built into Lynx. For example, you may see the message: **I don't know how to _____ or _____ does not like _____ as input.** Pay attention to these messages. They point out some command you need to fix.

There are also debugging tools to help you look for errors in your procedures. Every line in the Procedures Pane is numbered. So, if you see a message like this: **I don't know how to _____ in _____ (procedure name) on line _____** it tells you to check a specific line in the named procedure.

Of course, sometimes the problem isn't related to a typo or using the wrong type of input. Sometimes it has to do with the logic of your procedures. Take time to review step-by-step what each procedure does to figure out how to get the results you want.



Did we mention that you should save your work **often**?

Lesson 5 – Adding Other Fun Stuff!

Step 1: Multi-frame Animation and Multi-processing

You can create more complex animations by using multiple rapidly changing shapes to simulate motion, for example, a flying bird, flapping its wings or a running animal.

For example, create a running lynx using the sample clipart images.

In the Command Centre, type:

```
setsh [1 2 3 4]
```

This creates a process that rotates through the shapes in the square brackets as the turtle moves.

```
move
```

Use your procedure name. This initiates a second process.

Both processes run at the same time.

When using multiple shapes, always include the list of shape numbers in square brackets with a space between each number. You may want to edit the move procedure or write a different procedure to simulate the appropriate speed.

Step 2: Adding Sound and Music

To add a WAV or MP3 sound file to your project, choose **Sound** in the "+" menu. The **Import Sound** dialog box appears. Choose a sound file on your device or online and click **Create**.

A sound icon appears on your page.

Add your music or sound to one of your procedures. For example, add a car horn sound named **Honk**. Edit your `drive` procedure.

```
to drive
repeat 30 [fd 3 wait 1]
;this moves faster than the move procedure
honk
end
```

Test it.

Does the sound occur at the right time? Edit your procedure until you're satisfied.

Add additional sounds where appropriate in your story. Or record sound effects, using some other recording tool, or some spoken words (for example, you, narrating your story), save it as a WAV or MP3, and add it to one of your procedures.

You can right-click on the sound icon to edit its name. Use a single word, with no spaces because the name is now a command that plays the sound. You can also hide the icon by unselecting **Visible**.



Did we mention that you should save your work **often**?

Lesson 6 – Complete Your Story and Sharing It

Spend this session adding additional pages to your story, ensuring your navigation system works, and tweaking your procedures.

You may want to add a cover page or a table of contents. Include buttons or turtles as buttons to move to each page.

When you've finalized your story and are ready to go public, there are several ways to share your project. You should talk to your teacher about the best way to share your projects.

1. From the Lynx Editor

- From within the Lynx editor, simply click on the **Share** icon in the top-left corner of the editor.
- In the dialog box that comes up, choose an image file to use as a preview (use the buttons to get a file from your device, OneDrive, Dropbox or GoogleDrive).
- Enter a title and a description, then click on one of the Share sites (Twitter, Facebook) or copy the link to paste it where needed or click on E-Mail to send the link by email.

2. From within your Lynx personal space in the cloud

- From your Lynx personal space, click on your project to open it in Play Mode.
- Then click on Share and then the + sign in the URL field to obtain a link which you can copy and paste into an email or elsewhere.

There is also a button to post the project to your Facebook account.

3. Enjoy (and edit) my project

You can not only let someone view your project, but also let them edit it.

- Before sharing a project, go to its Properties in your Lynx personal space in the cloud
- Uncheck the Private check box
- Click on the Share Button and then the URL address field and Copy the Link.
- Send the link to a friend. Using the link, he / she will be able to make and save changes to your project.



Did we mention that you should save your work **often**?