CODE to LEARN | PROGRAMMER pour APPRENDRE

Create a Calculator Application

with

lynxcoding.club

With funding from

Canada

CODEtoLEARN.CA

TakingITGlobal
INSPIRE INFORM INVOLVE
INSPIRER INFORMER ENGAGER

DESCRIPTION

Create a Calculator Application

Students (grades 6-8) will create a Calculator Program.

Students will code this app using Lynx at **lynxcoding.club**.

Students will code the computer to:

- Retrieve input from the user via text boxes
- Create variables and store values
- Create an interactive button
- Perform operations on numbers entered by the user
- Animate a shape using Repeat
- Hide and show text boxes
- Advanced options include:
    - Incorporating additional buttons and operations
    - Try coding other applications that require the same skills

Success Criteria

Co-construct success criteria with your students.

LEARNING GOALS

Students will learn, and use, these...

BIG IDEAS IN CODING		
	MAIN IDEAS	
CODE & CONCEPTS	Designing a user interface with instructions	setshape, visible, transparent, showtext, hidetext, print, cleartext, sentence Manage object characteristics
	Repeat Loops	Text boxes, shapes buttons, wait, procedures Objects for controlling program flow
	Variables To store values, obtain information from user and perform calculations	Math Operators To add numbers

Get a Lynx Account

Details at lynxcoding.club

We suggest:
- teacher gets a **School Administrator Account**
- students get **permanent Individual Accounts**
- teacher creates a **'club'** and invites all students

NO Account
You can try Lynx for free without an account, by clicking on **Create a Lynx Project** on the home page at **lynxcoding.club**.

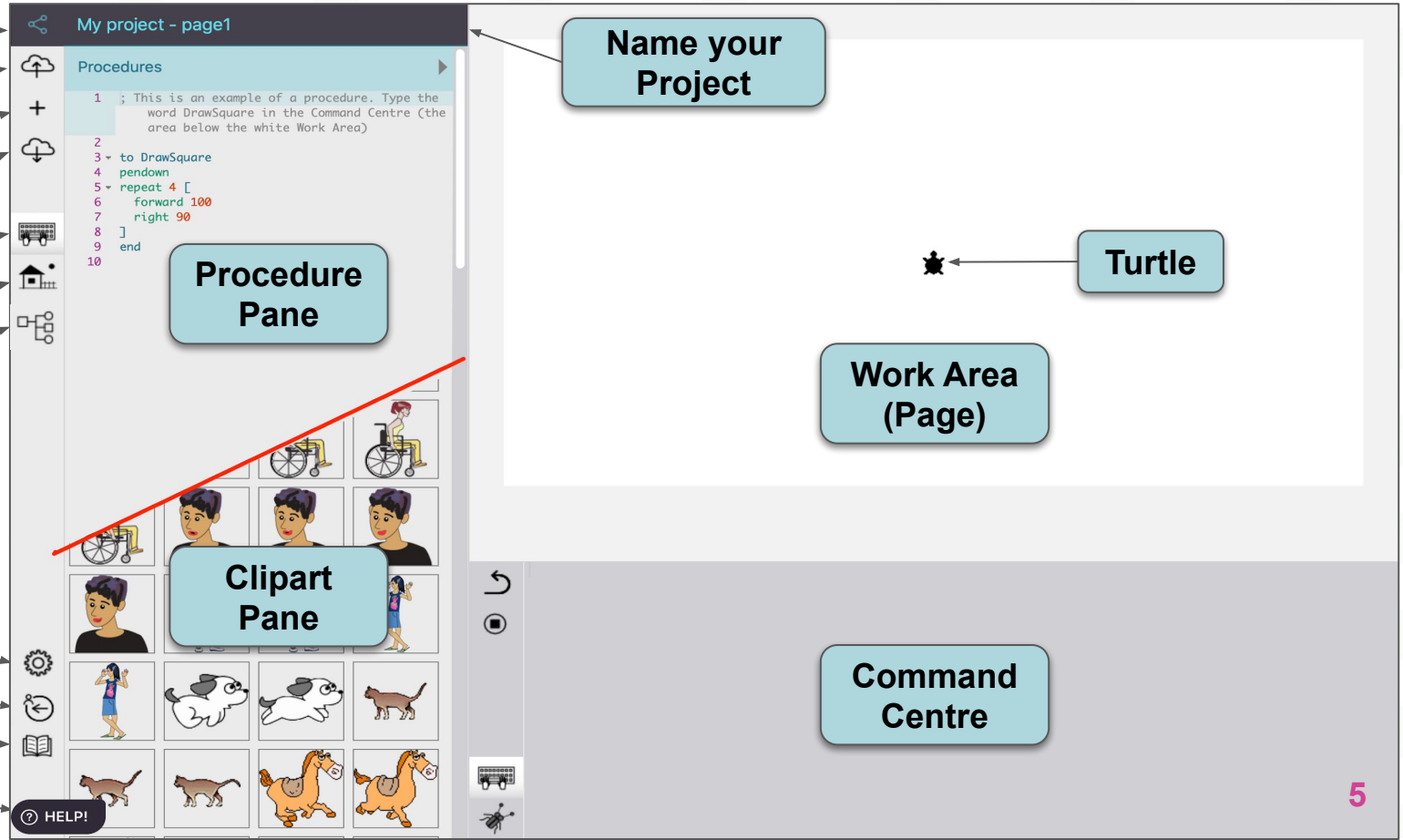
FREE TRIAL Account
For full access, register (click **Login/Register** located at the top, right side of the Lynx web page).

INDIVIDUAL Account
Convert your trial account to a permanent individual account before end of trial period.

SCHOOL ADMINISTRATOR Account
Convert your trial account to a School Administrator account before end of trial period.

Accounts are free for Canadians thanks to a subsidy by the Government of Canada.

Layout


Share
Save
Add Objects
Files
Procedures
Clipart
Project Tree
Settings
My Projects
Commands
Help
My project - page1
Procedures
; This is an example of a procedure. Type the word DrawSquare in the Command Centre (the area below the white Work Area)
to DrawSquare
pendown
repeat 4 [
forward 100
right 90
]
end
HELP!
Name your Project
Turtle
Work Area (Page)
Procedure Pane
Clipart Pane
Command Centre

STEPS FOR STUDENTS

The BIG Picture (*Table of Contents*)!

1. *Log in and Create a Lynx Project.*

2. *Create and manage text boxes.*

3. *Create a button.*

4. *Write procedures.*

SAVE often!

5. *Test out the program.*

6. *Add clipart and create an animation.*

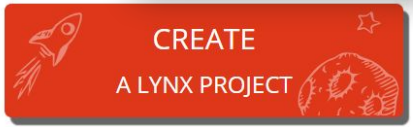
7. *Challenging yourself.*

8. *Make public and share.*

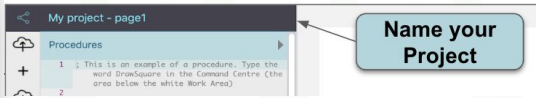
GETTING STARTED

1. *Log in on the Lynxcoding.club site.*

2. *Click on CREATE A LYNX PROJECT.*



3. *Start by renaming the project to Calculator.*

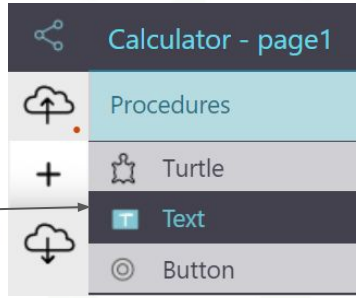


4. *Save*
Click this icon. Remember to save regularly as projects will NOT save automatically.

5. *Follow the next few cards carefully to set-up the Work Area. We will write the code to make your calculator work beginning on Card # 14.*

CREATING TEXT BOXES

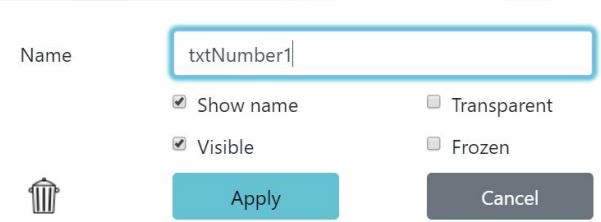
Create and Name your Text Box



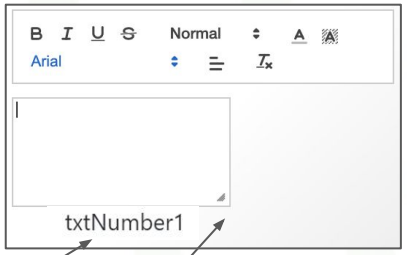
Save your project!

1. Click the + sign and select Text.
*A new text box with the name **text1** appears on your screen.*

2. Right-click on it.
A dialog box appears.



3. Type the name txtNumber1. All one word!



4. Click Apply.

5. Repeat steps 1 through 4 to add a 2nd text box and name it txtNumber2. All one word!

Move the text box: Grab it by the name and drag it.

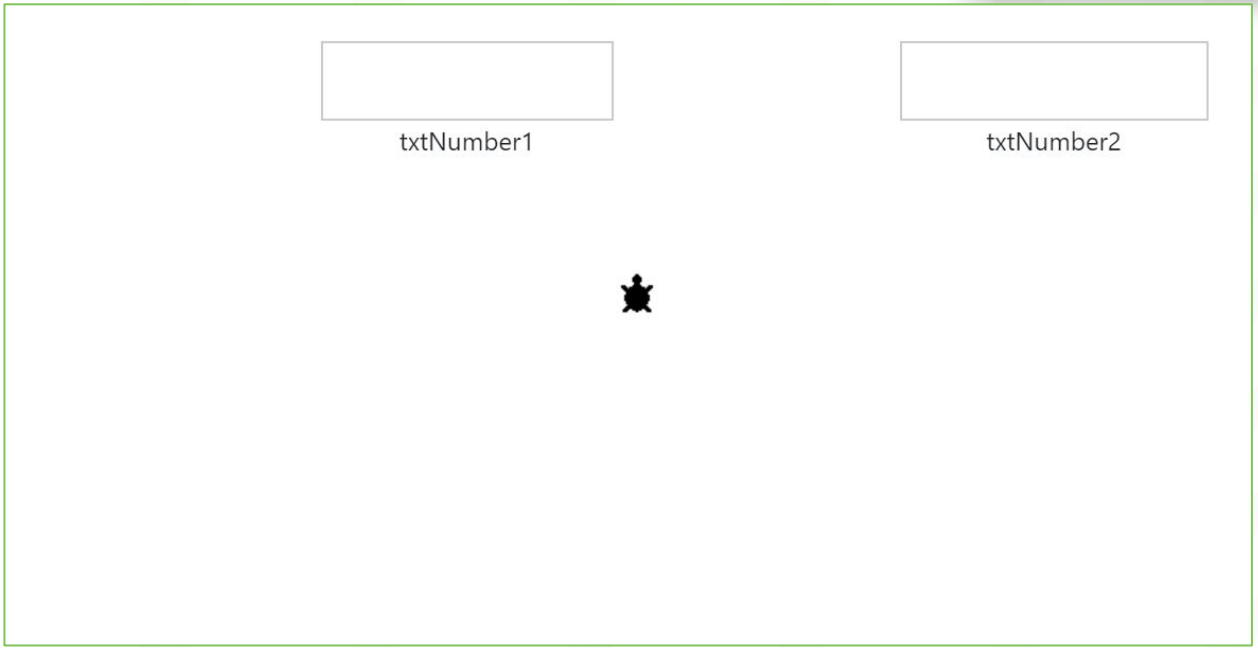
Resize it: Drag on the small triangle in the lower right corner that appears when you hover over it.

Format: Click inside the text box—or select text—and use the formatting commands above the text box.

MAYBE YOUR PAGE LOOKS LIKE THIS!

You Will Be Ready For the Next Step!

1. Make adjustments so you have a similar set-up.



You can see that I:

- adjusted the sizes of the text boxes
- moved their position on the page

MANAGING TEXT BOXES

Hiding the text box names.

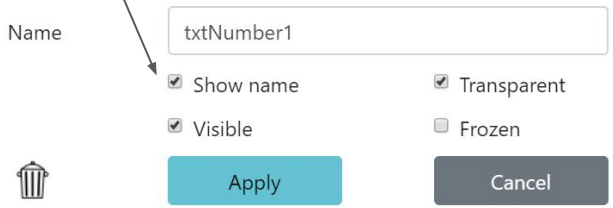
1. Right-click on the *txtNumber1* Text Box *again.*
The dialog box appears.

2. Uncheck *Show name*.

3. Click *Apply*.

4. Repeat steps 1-3 on this card for the *txtNumber2* Text Box

Tip - *If you need to move a text box, you will need to show the name again. Right-click inside the Text Box and check* ***Show name*** **then** ***Apply***.



Delete *a text box by clicking the trash can.*

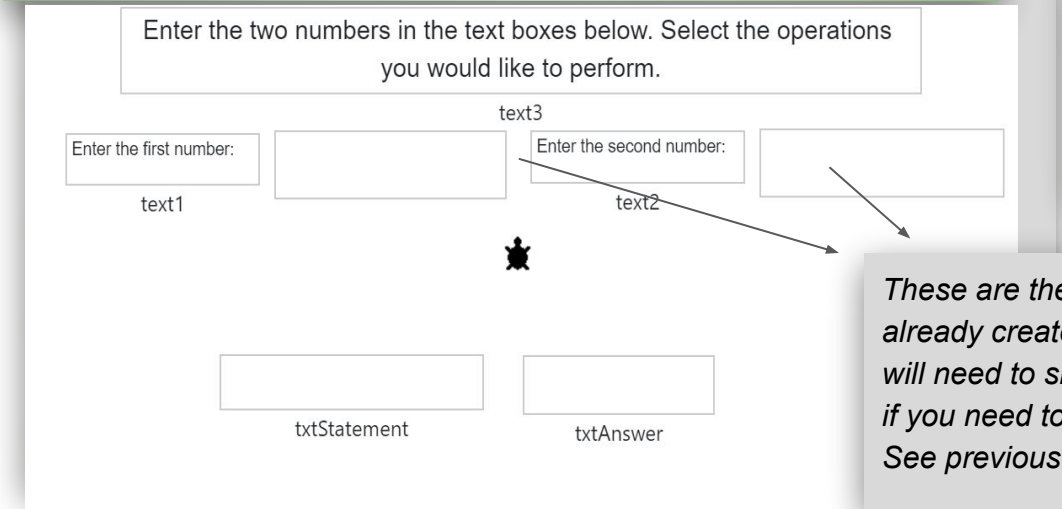
Save your project!

MANAGING TEXT BOXES

Let's add additional text boxes to give instructions to users.

1. Add 5 more text boxes. Position and name them as shown below. Add text to 3 of the text boxes by clicking inside the Text Box and typing the words below. Name 2 Text Boxes: txtStatement and txtAnswer. The 3 other Text Boxes can keep default names.

2. Right click on each textbox that you just added (not the original two) and check the Transparent property.



These are the text boxes you have already created. Rememberl, you will need to show the name again if you need to move them.
See previous Card.

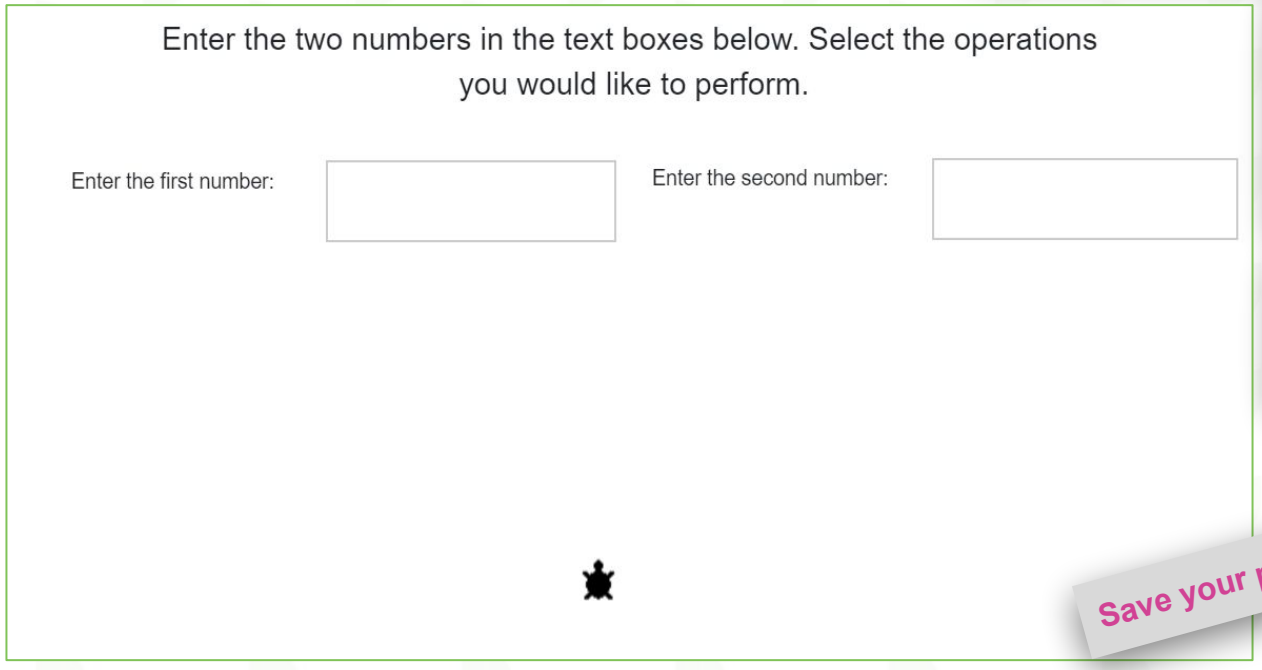
***Tip -** By making text boxes transparent, they appear as labels, rather than a text field to obtain input.*

*The two textboxes at the bottom **with nothing in the text fields** will appear invisible. That's okay!*

MAYBE YOUR PAGE LOOKS LIKE THIS!

You Will Be Ready For the Next Step!

1. ***Make adjustments so you have a similar set-up.***



2. ***Drag the turtle shape to the bottom of the Work Area. We will be working with this shape later.***

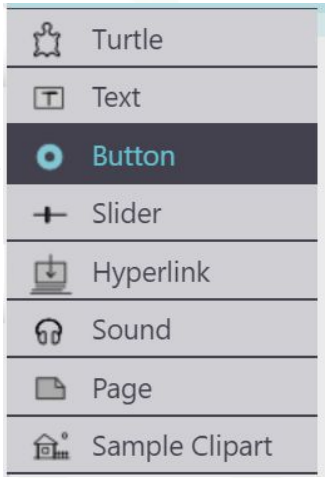
3. ***You may want to format the words inside the text boxes!***

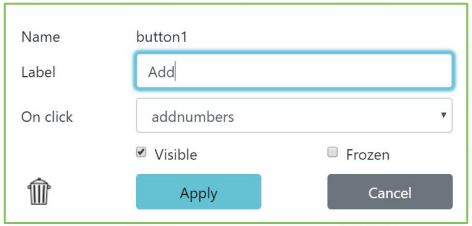
Save your project!

ADDING A BUTTON

Create an *Add* button for our calculator

1. Click the + and choose Button.
A button named **'nothing'** appears





2. Right click on the button and change the Label to Add.

3. Click on *Apply*.

4. Resize the button and move it to the middle of the *Work Area*.

Move the button: Click anywhere on the button and hold to move it.

Resize it: Drag on the small triangle in the lower right corner that appears when you hover over it.

Save your project!

CODING TIME – ADDING PROCEDURES

Let's write some procedures so that the numbers the user enters into the 2 text boxes are added together and the sum is displayed. We will start with a procedure that will store the numbers the user types in.

1. Click on the keyboard beside the Procedure Pane.

2. Type the following:

```
to storeNumbers
      txtAnswer, cleartext
      txtStatement, ct
      make "num1 txtNumber1
      make "num2 txtNumber2
end
```

Ct = cleartext

```
1  to storeNumbers  ;indicates the start of a procedure
     called storeNumbers
2    txtAnswer, cleartext ;clears the txtAnswer textbox
3    txtStatement, ct  ;ct is a shortform for cleartext
4                      ;clears the txtStatement textbox
5
6    ;stores the first number the user types into the
       textbox into a variable called num1
7    make "num1 txtNumber1
8     ;stores the second number the user types into the
        textbox into a variable called num2
9    make "num2 txtNumber2
10 end ;indicates the end of this procedure
```

Tip - *The semicolon (;) in the code above indicates the start of a comment. This helps people who are looking at your code understand what is happening. It will not affect how the program is run. It is simply an explanation.*

Tip - *I have indented the code inside of the procedure. This is not required, but is considered good programming style.*

CODING TIME – ADDING PROCEDURES

This next procedure will add up the two numbers the user enters that were stored into variables in our first procedure.

1. Click on the keyboard beside the Procedure Pane.

*The **sum** primitive can also be used to add numbers.*

2. Type the following:

```
To addNumbers
        storeNumbers
        make “sum :num1 + :num2
        txtStatement, print (sentence ‘The sum is:’)
        txtAnswer, print :sum
end
```

```
13  to addNumbers
14      storeNumbers  ;calls the storeNumbers procedure to
          make it happen
15      make "sum :num1 + :num2  ;adds up the values of
          num1 and num2 variables and stores the value
          into a variable called sum
16      txtStatement, print (sentence 'The sum is:')
          ;prints this statement into the textbox
17      txtAnswer, print :sum ;prints the value of sum
          into the textbox
18  end
```

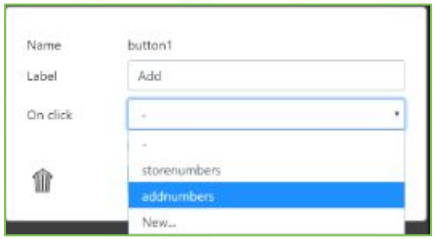
*A **variable** is a stored memory location that can hold values.*

***Tip -** Lynx helps you by “colour coding” your code as you type. You will start to notice what each colour means as you become familiar with Lynx.*
You will find this useful as you debug your code.

PROGRAMMING THE BUTTON TO BE INTERACTIVE

We will assign a procedure to the *Add* button.

1. Right-click on the *Button* called *Add* in the Work Area.



2. Click on the *On click* drop down menu. Choose *addnumbers*.

3. Click *Apply*.

Save your project!

TESTING THE PROGRAM

We will enter values into the text boxes to see if our program works!

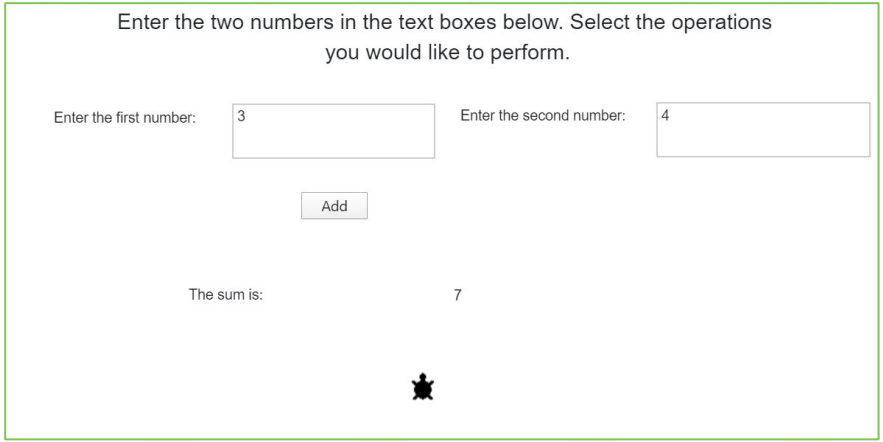
1. To test the program, enter the number 3 in the first text box.

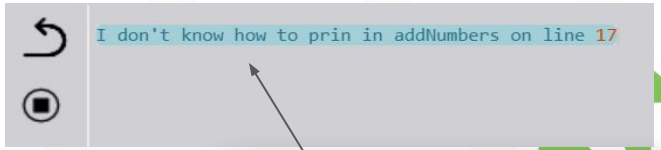
2. Enter the number 4 in the second text box.

3. Click on the Add button.

4. Does "The sum is:" and the number 7 appear?

5. If the program doesn't work, Lynx gives you a hint about what line of code has an error in the Command Centre.





This is an example of an error message that you would get if you misspelled the word print in your code.

ADDING CLIPART

Let's add a shape (image) which we will then animate to make it look like a character is "thinking" when the *Add* button is clicked.

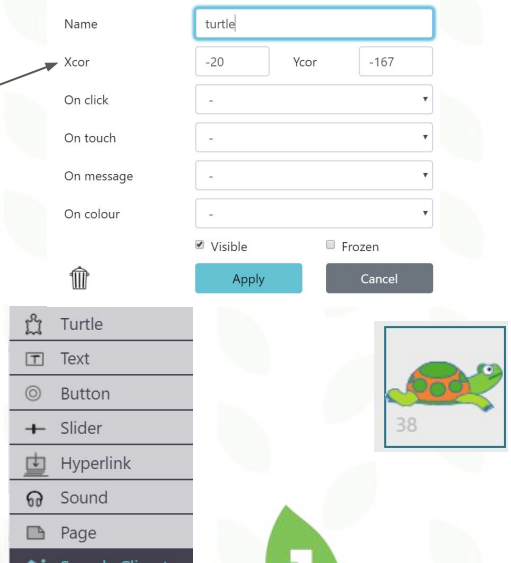
1. Right-Click on the black turtle, that is at the bottom of your Work Area, and rename it turtle.

2. Click on the + icon and select Sample Clipart and then in the submenu, select Animations.

3. The Clipart Pane will open with all Animations. The turtle clipart is number 38 and 39. Remember these numbers!

4. Right-Click on the black turtle, that is on the bottom of your Work Area and rename it turtle.

5. The Clipart Pane will open with all Animations. The turtle cursor to the Work Area, the cursor should appear as a hand.



ADDING ANIMATION

Let's animate the turtle shape to make it look like it is "thinking" when the user clicks on the *Add* button.

1. Click on the keyboard icon

You may have to adjust the shape numbers depending on your Sample clipart. You can also change the input to Wait

2. Add an animation procedure by typing:

To animate

Repeat 8 [turtle, setshape 38, wait 1, setsh 39, wait 1]

end

```
21 to animate
22   repeat 8 [turtle, setsh 38, wait 1, setsh 39, wait
       1] ;this will change shape 38 to shape 39, wait
       one second and repeat, 8 times so it looks like
       an animation
23 end
```

3. We need to call the Animate procedure inside the addNumbers procedure. Add this line of code: animate after the make "sum :num1 + :num2 line of code inside the addNumbers procedure:

```
25 to addNumbers
26   storeNumbers
27   make "sum :num1 + :num2
28   animate
29   txtStatement, print (sentence 'The sum is:')
30   txtAnswer, print :sum
31 end
```

*Tip: Animate is a subprocedure inside the **addNumbers** procedure*

TESTING THE PROGRAM

Let's run the program to see if the animation works properly.

1. To test the program, enter the number 25 into the first text box.

2. Enter the number 75 into the second text box.

3. Click on the Add button.

4. Does the turtle animation appear?

5. Does "The sum is:" and the number 100 appear AFTER the turtle animation?

6. Remember - If the program doesn't work, Lynx gives you a hint about which line of code has an error in the Command Centre.

MANAGING TEXT BOXES

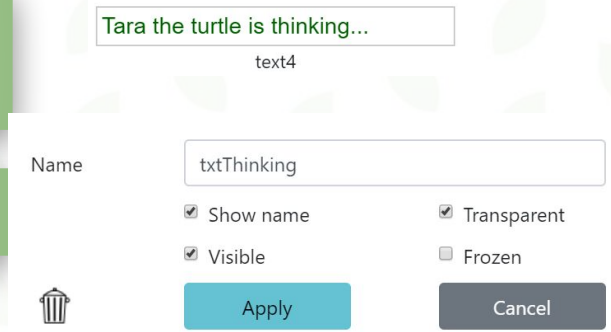
Let's learn how to hide and show a text box.

1. Add one more text box to your program beneath the turtle. In the text field, type: *Tara the turtle is thinking...*

2. Right-click on the text box and rename it *txtThinking*.

3. Uncheck *Visible* and check *Transparent*.

4. Click *Apply*.



Save your project!

Tip - *If you make the box* ***invisible****, and you need it back, no worries!*
Type this in the Command Centre: ***showtext***
The text box reappears.

Important: If you have more than one text box, you will have to call it by its name like this: ***txtThinking, showtext***

hidetext *does the opposite.*

PROGRAMMING A TEXT BOX TO APPEAR

Let's program the text box to appear when the turtle is "thinking"

1. Click on the keyboard icon

2. Add the following lines of code to the animate procedure:

```
To animate
        txtThinking, showtext
        Repeat 8 [turtle, setshape 38, wait 1, setsh 39, wait 1]
        txtThinking, hidetext
end
```

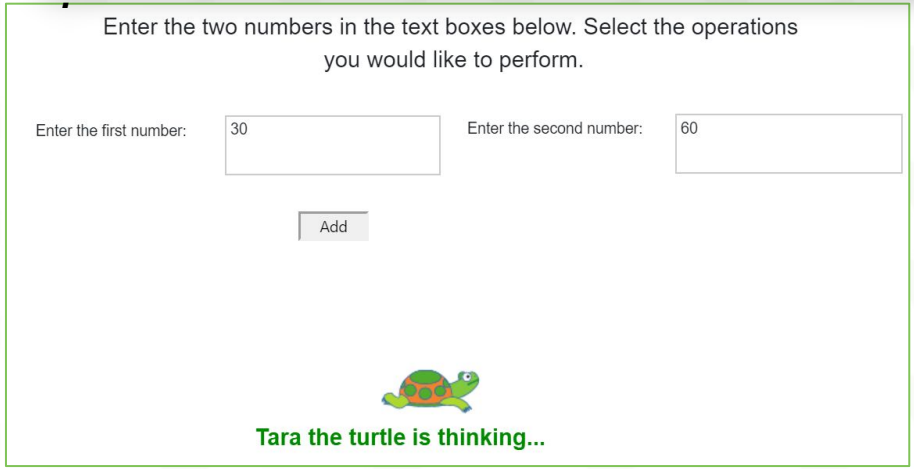
```
21 to animate
22    txtThinking, showtext ;displays the text box
23    repeat 8 [turtle, setsh 38, wait 1, setsh 39,
        wait 1] ;this will change shape 38 to shape 39
        , wait one second and repeat, 8 times so it
        looks like an animation
24    ;set "txtThinking "visible? "false
25    txtThinking, hidetext  ;hides the text box
26 end
```

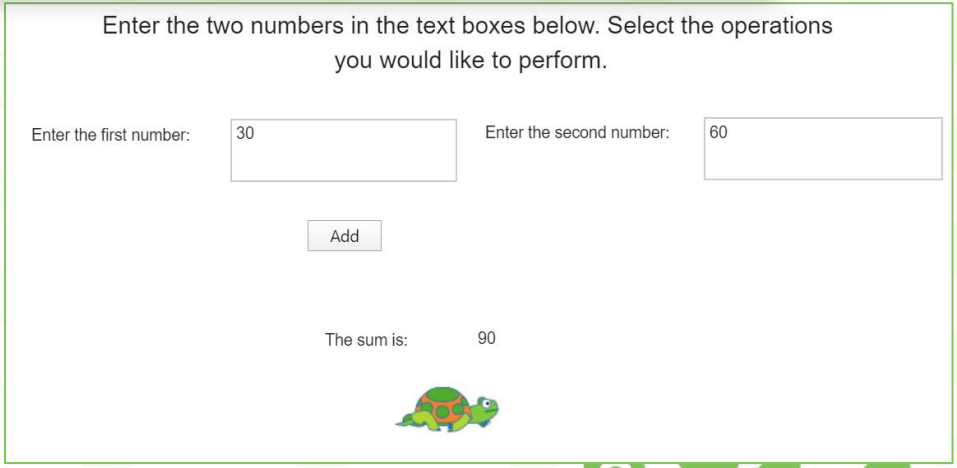
Save your project!

TESTING THE PROGRAM

Let's run the program one last time to make sure it's working.

1. Click the Add button. Immediately after the button is clicked, the screen should look something like the one on the left. After the animation stops, the screen should look like the one on the right.





CHALLENGE YOURSELF!

Check out these additional enhancements/challenges...

Enhancing your Calculator Program:

*Share your program with your friends and family by clicking on the **Share** icon.*
*If you would like them to be able to edit the code, uncheck **Private** in the **Project Properties** tab.*

***Add buttons and procedures for the following operations**: Subtract, Multiply, Division (see hints on card #25).*

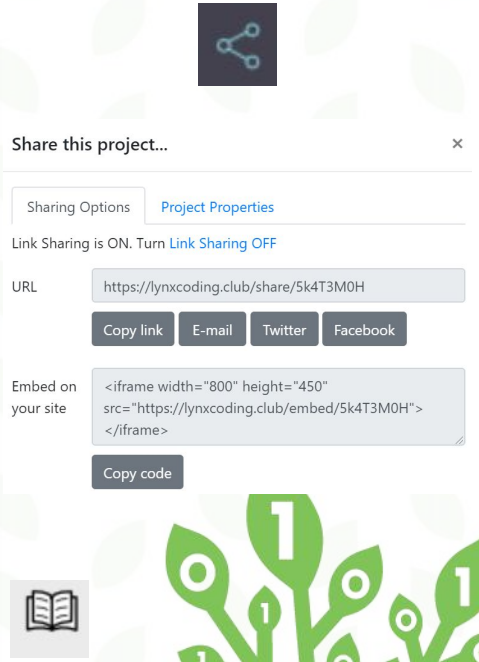
Spruce up your Work Area by adding colour, try using Shapes for Buttons.

Apply Your New Learning - Challenge Yourself to Make…

- ➔ *a cash register application*
- ➔ *a unit or currency conversion application (e.g., Canadian to US dollars, kilograms to pounds).*

The possibilities are endless!

*Help is available! Click on the **book** icon or Help Widget in the bottom left corner of Lynx, or select **Help** on the homepage and look at the User Guides*



HINTS FOR ADDITIONAL BUTTONS AND OPERATIONS

Here is some sample code for the multiply procedure

```
21 to multiplyNumbers
22   storeNumbers
23   make "product  :num1 * :num2
24   animate
25   txtStatement, print (sentence 'The product is:')
26   txtAnswer, print :product
27 end
```

Tip - *The asterisk (*) means multiply and the forward slash (/) means divide in computer programming.*

Tip - *When you hover over the green primitives, Lynx provides an explanation and required inputs to assist you.*

Tip - *For more math operators, check out the **Book** icon, bottom left corner, go to **Other Stuff**, under **The Big list**, go to **Complete list of Other commands** and check out the **Math** operators and primitives.*

CODE to LEARN

Credits

Principal Writer..... *Lisa Anne Floyd*
Contributors..... *Cohen Floyd*
Michael Quinn

Partners

LCSI

fair chance learning
what every student deserves

CISCO

21

Deloitte.

A program of
TakingITGlobal
INSPIRE INFORM INVOLVE

Connected North

With funding from
Canada