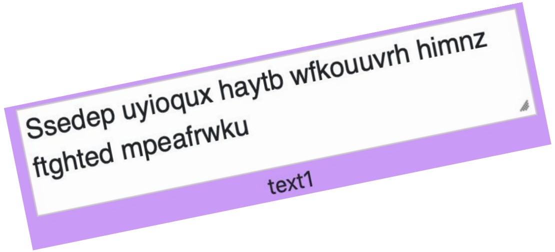
CODE to LEARN | PROGRAMMER pour APPRENDRE

How to create Secret Codes!

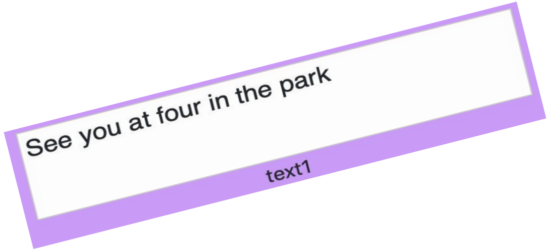
with

LYNX™

lynxcoding.club



Ssedep uyioqux haytb wfkouuvrh himnz ftghted mpeafrwku



See you at four in the park

With funding from

Canada

CODEtoLEARN.ca

TakingITGlobal
INSPIRE INFORM INVOLVE
INSPIRER INFORMER ENGAGER

DESCRIPTION

Overview and Learning Goals

Important note:

You can find a longer version of this project *Secret codes* at Lynxcoding.club > Help > User Guides.
Comprehensive “walkthrough” instructions are available in that guide.

DESCRIPTION

Creating Secret Codes

Students (grades 3-10) will create a few algorithms to encode and decode messages.

There are many ways to encode messages so only designated recipients can read them. Secret codes have been used for thousand of years by government, spies, crooks and good guys to send messages across "enemy" lines.

This activity is based on text manipulation - because this is what encoding is all about.

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

Students will code the computer to:

- Create a text box which will contain the clear (readable) message.
- Create an algorithm (a method) for encoding (scrambling) the message.
- The algorithm will "do something special" to every character of the clear message, in order to generate the encrypted message.
- The student will also create a "decode" algorithm in order to process the encoded message and retrieve the original "clear" message.

Success Criteria

Co-construct success criteria with your students.

LEARNING GOALS

Students will learn, and use, these...

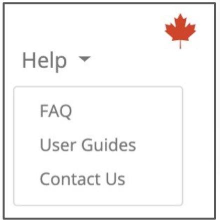
BIG IDEAS IN CODING		
	MAIN IDEAS	
CODE & CONCEPTS	Clear the text, insert, print or delete text in a text box ct, print, insert, delete	Count the number of characters in a text box show count text1
	Commands for moving the cursor in a text box top, cb, cf	Think in pseudo-code: Move the cursor and insert an “x” then transform your idea into real code: cf insert 'x'
	A numerical code for each letter show ascii ‘a’ show char 88	

GETTING STARTED

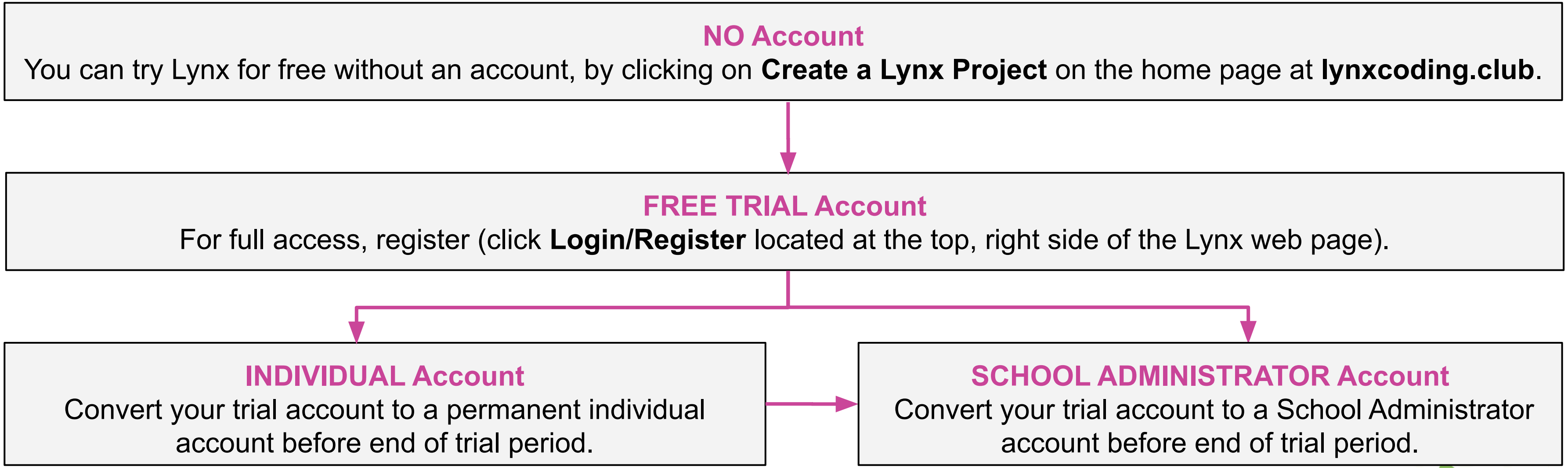
Get a LYNX Account and Understand the Layout

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**.

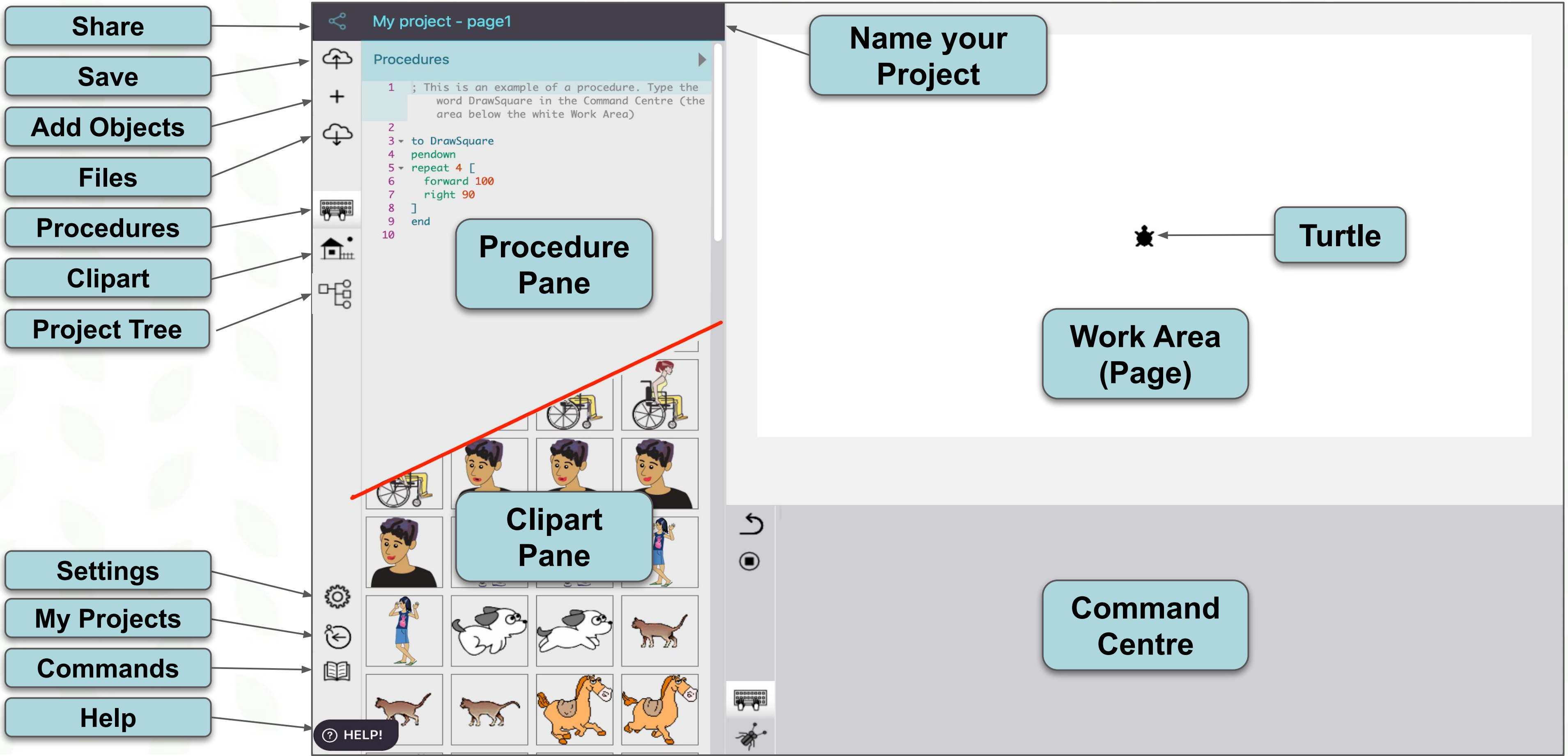
FREE TRIAL Account
For full access, register (click **Login/Register** located at the top, right side of the Lynx web page).

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

Name your Project

Procedure Pane

Clipart Pane

Turtle

Work Area (Page)

Command Centre

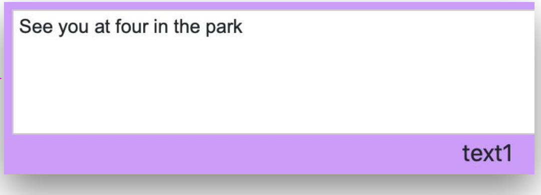
```
; This is an example of a procedure. Type the
    word DrawSquare in the Command Centre (the
    area below the white Work Area)

to DrawSquare
pendown
repeat 4 [
  forward 100
  right 90
]
end
```

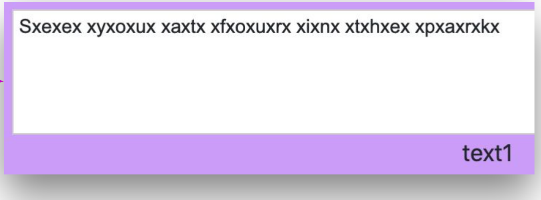
STEPS FOR STUDENTS

Table of Contents (the big picture). Soon you will learn how to do this.

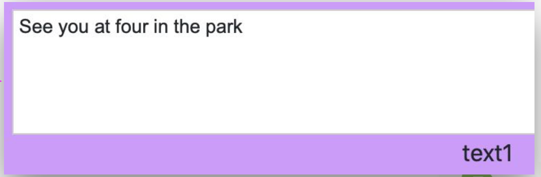
1. ***Log in and create a new project.***
2. ***Give a name to your project and save it. There is NO autosave.***
3. ***Create a text box for your "readable" message.***
4. ***Think (pseudo-code) about a way to "scramble" your text.***
5. ***Create a procedure that manipulates the text, in a way that corresponds to your pseudo-code.***
6. ***Create a procedure that does the exact opposite, to "decode" the scrambled text.***
7. ***Then, redo steps 4, 5 and 6 with a more complex code!***



See you at four in the park



Sxexex xyxoxux xaxtx xfxoxuxrx xixnx xtxhxex xpxaxrxkx



See you at four in the park

Your first secret code

Log in, start a new project, and create your first secret code (code and decode procedures)

Create a New Project

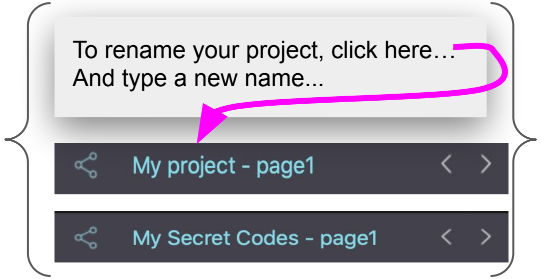
Add a Wide Text Box

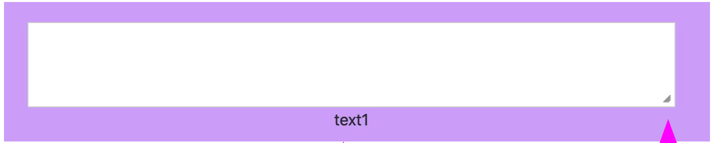
1. *Log in and create a new project.*

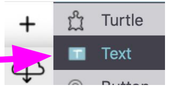
2. *Give a meaningful name to your project, and save it.*

3. *Set a background colour you like.*
   setbg 112
   You can find all the colour numbers in the online Help.
   *Click on 📖, choose **Turtle and Graphics** and scroll down to **Setcolour**. Also on the Lynx Home Page> Help> User Guides has a **Lynx Colour Chart**.*

4. *On the left side, look for the + sign and move to the right and select **Text.***
5. *Use the bottom right corner to stretch the text box across the page*
6. *Drag by the name of the text box (text1) if you want to relocate the text box on the page.*


To rename your project, click here…
And type a new name...





First secret code

Think in pseudo-code

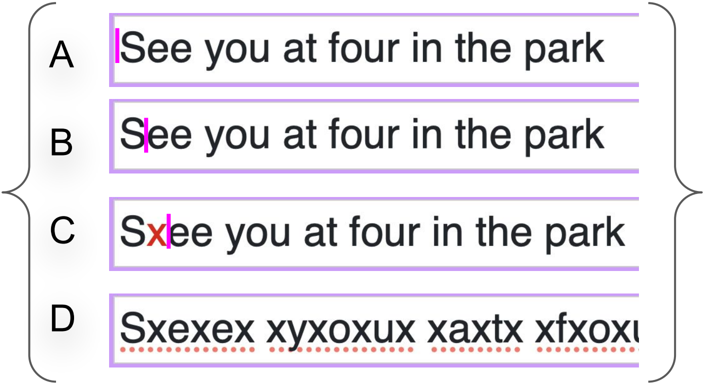
Save your work!

1. *First, type a message in the text box. This is called a "clear" message, a message anyone can read.*

Think of a way to scramble the message. In this first example, you will insert an "x" after each character (letter).

Now thinking in pseudo-code means: make the program in your head (or on paper), using your own words, instead of real Lynx instructions.

- A. *Bring the cursor to the beginning of the message.*
- B. *Move the cursor "1 character" to the right.*
- C. *Insert an "x".*
- D. *Repeat steps B and C up to the end of the message (I should figure out how many characters I have in my message).*



A See you at four in the park

B See you at four in the park

C Sxee you at four in the park

D Sxexex xyxoxux xaxtx xfxoxu

First secret code

Now put that into real Lynx code!

Save your work!

In Lynx, you can give commands to the cursor (the insertion point) that's inside a text box, just like you can give commands to the turtle.
Here are the commands you will need for this secret code:

- cf (stands for Cursor Forward) moves the insertion point one character to the right.
- top moves the insertion point to the top of the text box.
- insert inserts whatever you want, exactly where the insertion point is.

The other important trick is that the **name** of the text box (text1) returns the entire contents of the text box, as a long, long word. In our example, text1 reports See you at four in the park as one word.
You can use the primitive count to figure out how many characters there are in that word, which is to say, how many characters there are in the text box:

```
show count text1
27 (this is what Lynx returns for our example)
```

Now do you see how you can turn your pseudo code into real Lynx code?

```
A. Bring the cursor to the beginning of the message.
B. Move the cursor “1 character” to the right.
C. Insert an “x”.
D. Repeat steps B and C up to the end of the message (I should figure out how many characters I have in my message - count text1 does that).
```

becomes

```
to code1
text1, top
repeat count text1 [cf insert 'x']
end
```

Note: in Lynx, make sure you use straight quotes and double quotes (not curly), as you type or as you copy and paste instructions:
'a' and "a, not ‘a’ and “a.

First secret code

Enough talking. Let’s do it!

Save your work!

1. *If you don’t have a “clear” message in your text box, create one now.*

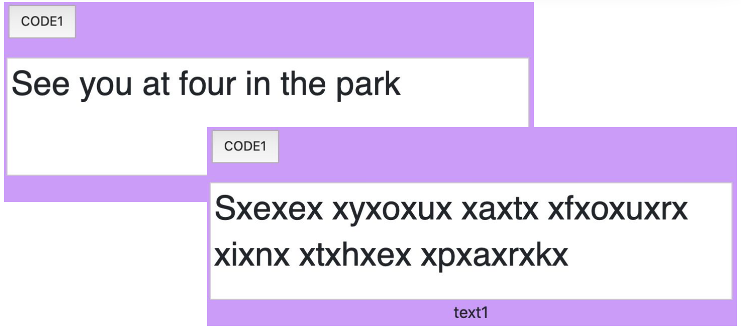
2. *Create this procedure in the Procedures Panel. In this example, I use code1 as the name of the procedure, because you are going to make more than one code procedure.*

```
to code1
text1, top
repeat count text1 [cf insert 'x']
end
```

3. *You can run this procedure from the Command Centre, but a button is so cool. On the left side, look for the + sign and move to the right and select Button.*

4. *Right-click on the button to open its dialog box. Type whatever you want as the label, and choose your procedure code1 in the On click menu.*

5. *Test your CODE1 button. Enlarge the text box if you have to.*



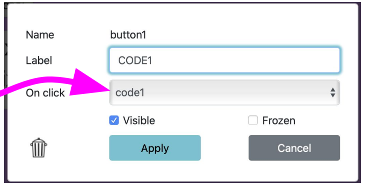
CODE1

See you at four in the park

CODE1

Sxexex xyxoxux xaxtx xfxoxuxrx xixnx xtxhxex xpxaxrxkx

text1



Name: button1

Label: CODE1

On click: code1

☑ Visible □ Frozen

First secret code

Save your work!

Nxexexdx xtxox xdxexcxoxdxex xnxoxwx!x

First some pseudo-code: Look at your encoded message. Think about what you DID to the message to encode it, and figure out a way to "undo" this.

A. *Bring the cursor to the beginning of the message.*
B. *Move the cursor "1 character" to the right.*
C. *Delete one character (the "x").*
D. *Repeat steps B and C up to the end of the message (CAREFUL HERE. The encoded message is TWICE as long as the original message, because of all the "Xs" you added!)*

You will need one more command here:

- delete   deletes one character at the right of the insertion point, much like the **Delete** key on your keyboard.

1. *Create this procedure in the Procedures Panel.*

```
to decode1
text1, top
repeat (count text1) / 2 [cf delete]
end
```

See the first input of repeat?
The original message had **27** characters. You added 27 "Xs" to scramble the text, so now count text1 will say you have **54** characters in the text box. Divide that by 2 in order to delete just the 27 "Xs".

2. *Create a button, type the label, and choose decode1 in the On click menu.*
3. *Test it on your encoded message.*
4. *If all goes well, type a new message in the text box and test your code1 and decode1 buttons.*

Your second secret code

The first secret code was not too hard to “crack”. A good spy would figure out that all you have to do is “remove the “Xs”.

Let’s make it harder to decode!

Second secret code

Each letter has a numerical value: its ASCII code

Save your work!

In this second secret code, you will add random characters (instead of always "Xs").

All the letters you have seen have a corresponding **number**: its ASCII code. You can use the random primitive to pick a **number**, and insert the random character that correspond to that **number**. You will need two more Lynx primitives to run this secret code:

- ascii returns the **numerical value** of a character. Try this in the Command Centre:

  ```
  show ascii 'a'
  ```

  97 Lynx returns the ascii code of 'a'

- char returns the **character** that corresponds to a number. Try this:

  ```
  show char 98
  ```

  b Lynx returns the letter that has the code 98

ASCII TABLE

This is the list of all the ASCII values for the letters a to z.

a: 97	*h: 104*	*o: 111*	*v: 117*
b: 98	*i: 105*	*p: 112*	*w: 118*
c: 99	*j: 106*	*q: 113*	*x: 119*
d: 100	*k: 107*	*r: 114*	*y: 120*
e: 101	*l: 108*	*s: 115*	*z: 121*
f: 102	*m: 109*	*t: 116*	
g: 103	*n: 110*	*u: 117*	

Second Secret Code

Save your work!

Simple random and modified random.

Random 26 *will give you a random number between **0** and **25**... But that’s not what you need to get random letters… based on the ASCII table, you need **random numbers** between **97** and **121** to get **random letters** from **a** to **z**.*

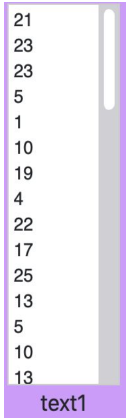
Here are the commands you will need for this secret code:

- cleartext clears the text in the current text box.
- print prints the text, then brings the insertion point to the next line (unlike insert, which prints the text, and leaves the insertion point in place.

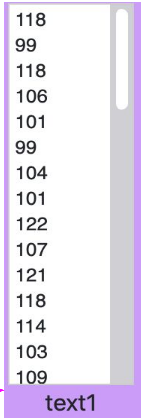
1. *Try this in the Command Centre:*
   cleartext repeat 50 [print random 26]
   *This prints 50 random numbers between **0** and **25**.*

2. *Let’s ADD to these numbers. Try this:*
   *cleartext repeat 50 [print **97 +** random 26]*
   *97 + random 26 can be **as low as 97** (if random 26 gives you a **0**)*
   *or **as high as 121** (if random 26 gives you a **25**).*
   *Therefore, the instruction above prints 50 numbers between **97 and 121**.*

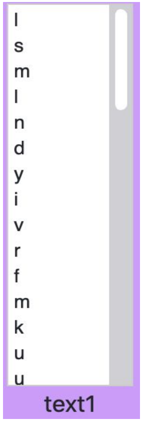
3. *Just what you need to cover all the codes in the ASCII table from the previous page! Now try this:*
   *cleartext repeat 50 [print **char** 97 + random 26]*
   *Thanks to char, you will get **letters** of the alphabet instead of **numbers**.*



```
21
23
23
5
1
10
19
4
22
17
25
13
5
10
13
```



```
118
99
118
106
101
99
104
101
122
107
121
118
114
103
109
```



```
l
s
m
l
n
d
y
i
v
r
f
m
k
u
u
```

Second Secret Code

Make a new code procedure

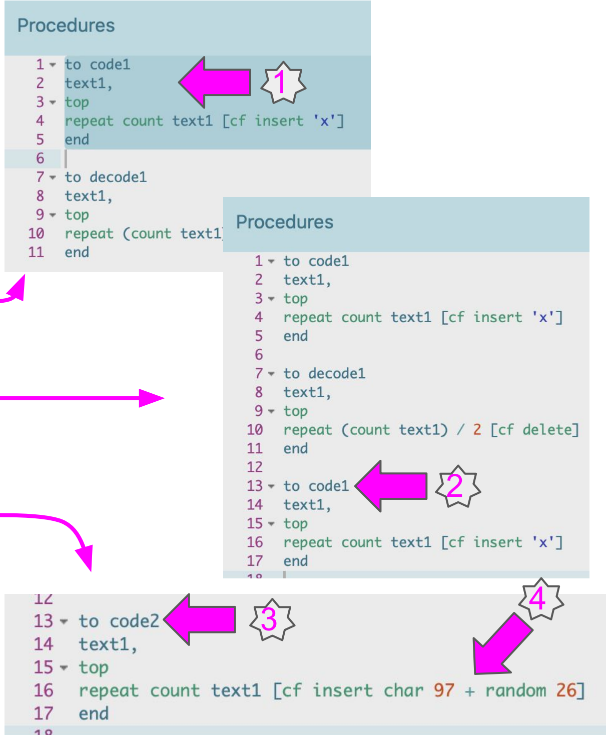
Save your work!

Your new “improved” code procedure is very similar to the one you made earlier. First make a copy of it, give the copy a different name (all the procedures MUST have unique names), and modify the copy.

1. *Select the procedure code1 in the Procedures panel, just like you would select text in any text editor. Then choose* ***Copy*** *in the browser’s menu. You can also right-click on the selected text and select* ***Copy***

2. *Click on an empty line, below the existing procedures, then choose* ***Paste*** *in the browser’s menu or right-click below your procedures and select* ***Paste.***

3. *You now have two copies of the code1 procedure. Edit the name of the one you just pasted and call it code2.*
4. *Change the instruction insert 'x' to insert char 97 + random 26. As explained on the previous page, this will insert a random letter instead of 'x'.*



```
to code1
text1,
top
repeat count text1 [cf insert 'x']
end

to decode1
text1,
top
repeat (count text1
end
```

```
to code1
text1,
top
repeat count text1 [cf insert 'x']
end

to decode1
text1,
top
repeat (count text1) / 2 [cf delete]
end

to code1
text1,
top
repeat count text1 [cf insert 'x']
end
```

```
to code2
text1,
top
repeat count text1 [cf insert char 97 + random 26]
end
```

Second Secret Code

Test your new code2 procedure

Save your work!

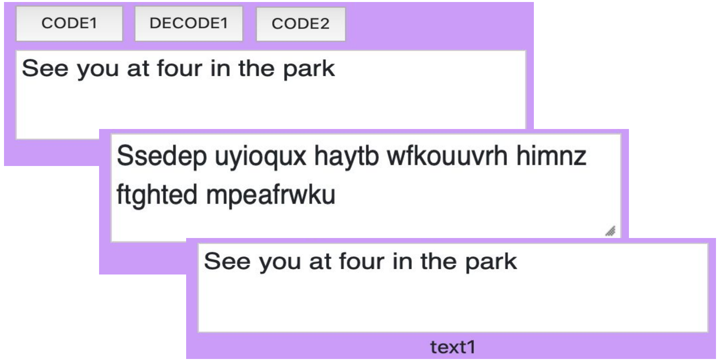
Let's test all this. The new secret message should be a bit harder to guess.

1. *Just like you did on **Page 13**, create a button for you new code2 procedure. You don't need a new decode procedure, because the one you have will work fine (it will delete the random characters instead of the "Xs").*

2. *Clear the text in Text1 by using cleartext and try new messages. What happens if you run code1 THEN code2? How do you "descramble" that? What if you run code2 and code2 again?*

CHALLENGES

3. *Try to insert TWO random characters between each good character, to make it even more confusing. You will need a new code and a new decode procedure if you do that.*
4. *Try to include capital letters and numbers in your algorithm. You can find on the web the ASCII table for letters A to Z (uppercase) and numbers. You will have to change your starting point (96) and your random number (26).*



See you at four in the park

Ssedep uyioqux haytb wfkouuvrh himnz ftghted mpeafrwku

See you at four in the park

CODE to LEARN

Credits

Principal Writer............ *Alain Tougas*
Contributors................ *Michael Quinn*

Partners

LCSI
fair chance learning what every student deserves
CISCO
C21
Deloitte.

A program of
TakingITGlobal INSPIRE INFORM INVOLVE
Connected North
With funding from
Canada