

WANT TO MAKE YOUR OWN APPS? YOU'VE COME TO THE RIGHT PLACE!

With Bitsbox, you can code real apps that work on gadgets like phones and tablets. All you need is a computer with internet and a working brain.

HOW TO BITSBOX

- Find a computer with a physical keyboard.

 The coding part of Bitsbox isn't meant to be done on tablets just yet.
- Open a web browser and go to bitsbox.com
 We recommend Chrome, Firefox, Safari, or Internet Explorer 11.
- Click Get Started.

 Have fun!

How can I run my apps on a phone or a tablet?

Before you do these steps, install a QR reader app on your gadget. Go here for a couple of suggestions: bitsbox.com/QRapps

- 1) On a computer, open the Bitsbox app you built.
- 2 Click the sharing icon < in the corner of the screen.
- 3 Scan the QR code with your gadget.





When you change the code on your computer, the app on your gadget changes, too!

ARTSY SMIARTSY

ere's something that might surprise you:

Both of the guys who invented Bitsbox studied ART in university! Scott was a painter and a graphic designer, and Aidan did a bunch of jewelry and metalsmithing, photography, and illustration.

Coding isn't just for hardcore computer programmers and scientists. If you love to draw, if you're a dancer or a musician, if you're a creative kid of any kind—the tech world needs you most of all! Technology can make life easier, but art makes life worth living.

To the studio!

Scott, Aidan, Anastasia & Jeff (the whole Bitsbox team)

GROWNUPS READ THIS!!

You'll find the *Grownup Guide* for this issue at bitsbox.com/grownups

In the Grownup Guide, you'll find:

- An FAQ section about Bitsbox in general (and this issue in particular)
- Descriptions of the coding concepts we're using
- Explanations of each app's code
- Suggestions for extending the apps to make them even more fun



MINI APPS TO GET YOU STARTED

2470 ACHOO!

- fill('yellow green')
 stamp('splat3',700)
- Gesundheit! Can you change the fill color to yellow?

7780 ART CRITIC

- 1 text('Was that gross?',100,500)
- 2 text('Or was it art?',400,600)

Try changing the code to make the text say something completely different.

5005 ART WILL BLOW YOUR MIND

stamp('brain').tap = explode

How about making a butterfly explode instead?

835D ROCKET LAUNCH

- fill('desert sky')
- 2 stamp('rocket5').move(UP,800,2000)
- 3 sound('rocket')

Check the back of this book for other stamps you can launch!

1773

NAME GAME

Paint your name!

Start by typing in this code and running it:

- 1 fill('paper2')
- stamp('a paint',100,200)
- 3 stamp('l paint',200,200)
- 4 stamp('i paint',300,200)

Can you spell out your name instead?

You may have to add more lines of code.

Now move your cursor across the tablet on your screen.

Do you see the little numbers?

They help you figure out where to place stamps.

Add another word below your name. Start typing another line of code:

- 1 fill('paper2')
- 2 stamp('a paint',100,200)
- 3 stamp('l paint',200,200)
- 4 stamp('i paint',300,200)
- 5 stamp('r paint',

Next, put your cursor where the next letter should go and look at the numbers.

The 'x' number above the screen is around **200**. The 'v' number beside the screen is around **400**.

Now add these numbers to your code:

- 1 fill('paper2')
- 2 stamp('a paint',100,200)
- 3 stamp('l paint',200,200)
- 4 stamp('i paint',300,200)
- 5 stamp('r paint',200,400)



Fun even if your name isn't Ali.



Can you finish spelling out the word "ROCKS" by adding four more lines of code?

3524

GOODBYE PICASSO

Flush poor Picasso down the commode.



Not all ideas are good ideas

```
fill('flush')
man = stamp('artist',400)
time = 2000
man.size(0,time)
man.rotate(720,time)
```

ᢔ YOU CONTROL THE SPEED.

The code commands on lines 5 and 6 both use the variable **speed** to control how fast the toilet flushes. When **speed** = 2000, it takes 2000 milliseconds (2 seconds) for the size of Picasso to decrease to 0, and for him to rotate 720 degrees (2 full turns).

Using a variable makes it easier to change this app's behavior. Changing the value of **speed** on line 3 automatically affects the code on lines 5 and 6. That's the power of using variables!

What else can you flush?

Tap to add a numbered dot, then drag to draw lines that connect them.

5052

Make your own puzzles!

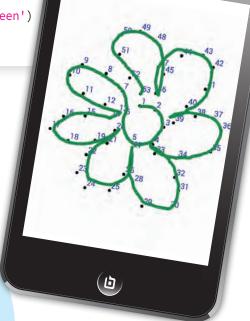
CONNECT THE DOTS

```
1  number = 1
2  function tap() {
3    circle(x,y,5,'onyx')
4    text(number,x+10,y,30,'blue')
5    number = number + 1
6  }
7
8  function drag() {
9    circle(x,y,5,'sea green')
10  }
```

Experiment with the code on line 9 to change the thickness and color of your drawing tool.

TAP TO ADD. DRAG TO DRAW.

tap() and drag() are fun Bitsbox commands that you can use to build apps that respond when users touch the screen in different ways.



Open the sharing menu on your screen to download screenshots of your app! **Anything worth** saying is worth saying big.

Write messages in the sky.



```
fill('sky blue')
   text('marry me?',100,545,50,'white')
   plane = stamp('skywriter',150,540)
   function loop() {
     plane.move(RIGHT, 10)
6
     plane.wrap()
8
```

& THAT'S A WRAP

Line 7 features the .wrap command. When the centerpoint of the airplane stamp moves offscreen, the app automatically "wraps" it around to the other side. Neat. huh?

Can you change the message that the airplane writes?

7 0 1 9

11 }

THIS LINE PICKS A

RANDOM

NUMBER

FROM I TO

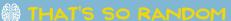
255.

LOTS O' DOTS

function loop() { r = random(255)g = random(255)b = random(255)x = random(150,600)v = random(1024)size = random(10, 30, 50)8 9 color(r,g,b) circle(x,y,size,color) 10

THIS LINE PICKS A RANDOM NUMBER FROM 150 TO 600.

Watch this app draw colorful circles forever. And ever.



Try changing the numbers on lines 5 and 6 to restrict the circles to different parts of the screen.











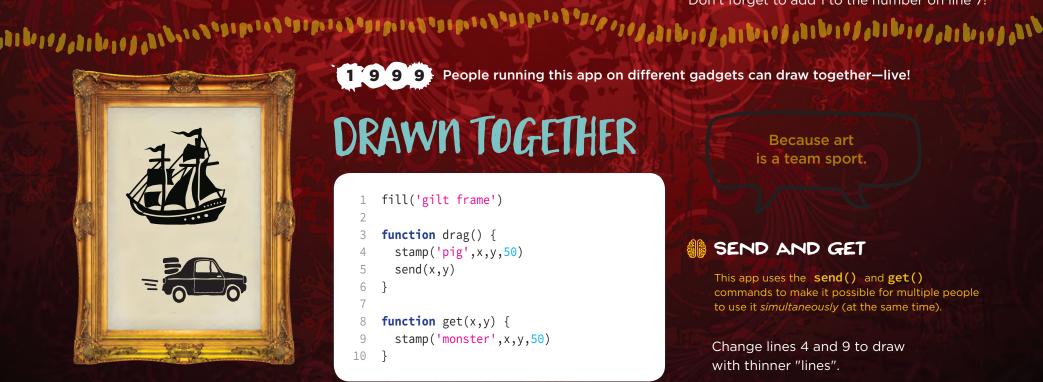
Fate is my curator.

ARRAYS ARE NUMBERED LISTS.

In coding terms, lists are called arrays. In this app, art is an array that contains three elements which happen to be famous paintings. Line 7 picks one of these at random, and line 8 stamps it on the screen.

```
1 fill('gilt frame')
   art = []
  art[1] = 'american gothic'
  art[2] = 'the scream'
   art[3] = 'mona lisa'
   choice = art[random(3)]
   stamp(choice)
```

Add another element to the array. Don't forget to add 1 to the number on line 7!



People running this app on different gadgets can draw together—live!

DRAWN TOGETHER

```
fill('gilt frame')
    function drag() {
      stamp('pig',x,y,50)
      send(x,y)
    function get(x,y) {
      stamp('monster',x,y,50)
10
```

Because art is a team sport.



SEND AND GET

This app uses the **send()** and **get()** commands to make it possible for multiple people to use it simultaneously (at the same time).

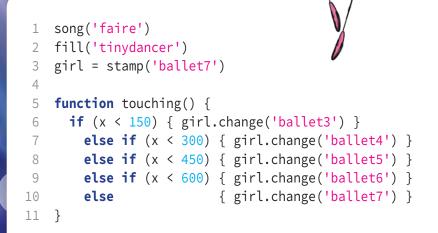
Change lines 4 and 9 to draw with thinner "lines".

2 6 8 1

With apologies to Sir Elton.

TINY DANCER

Drag left and right to make your ballerina boogie.



Pick a different song to play from the list on the back of this book.



This app uses **if** and **else if** statements to display a different stamp depending on where you're touching the screen. The **x** in lines 6 through 9 refers to the horizontal (side-to-side) position of your cursor (or finger) on the screen: 0 is all the way left, and 768 is all the way right.











Keep it clean, folks.

MAD LIBS

Tap the words and replace them with your own. Tap **Show!** to read the story. Once upor there was

```
1 function ask() {
      word = this.name
      reply = prompt('I need a ' + word + '!')
      text(reply,this.x,this.y,65,'blue')
      this.hide()
 6
    stamp('feeling',200,892).tap = ask
    stamp('noun',100,258)
                             .tap = ask
10
    stamp('place',325,350)
                             .tap = ask
11
    stamp('name',100,530)
                             .tap = ask
    stamp('animal', 100, 710) .tap = ask
12
13
    stamp('verb',380,800)
                             .tap = ask
14
15
    function show() {
16
      fill('madlib')
17
      this.hide()
18
19
    stamp('show', 670, 970).tap = show
```

name '

ed in

e day, a

ng and

TAP TO REPLACE.

Bitsbox's **.tap** command is on full display in this app.
It lets you tell a stamp what to do when someone taps it.

So _____ The end.

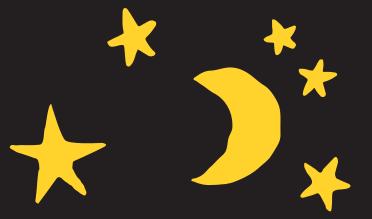
5 9 6 4

NANO PIANO

Build your very own musical instrument.

```
fill('nano piano')

stamp('piano c',160,720).touch = sing
stamp('piano d',235,720).touch = sing
stamp('piano e',310,720).touch = sing
stamp('piano f',385,720).touch = sing
stamp('piano g',460,720).touch = sing
stamp('piano a',535,720).touch = sing
stamp('piano b',610,720).touch = sing
stamp('piano b',610,720).touch = sing
```



SOME STAMPS CAN SING.



Each key on this piano is a different stamp, and each of these has its own, embedded sound effect. When you tell one of these stamps to **sing**, it plays its sound. Each of these keys plays a different musical note.

Another way to make a stamp play a sound when you tap it is to use code like this:

```
stamp('chicken').tap = cluck
function cluck() {
   sound('chicken')
}
```

This code is a lot longer, but it lets you associate *any* sound with *any* stamp, which can be very rewarding. And distracting.



Check out the free Sketch-a-Song app! bitsbox.com/sketchasong (our friends made it)

