

What's Bitsbox?

With Bitsbox, children learn to program by creating fun apps that work on computers and gadgets like iPads and Android tablets. The Bitsbox.com website provides each child with a virtual tablet and a place to type their code. The experience starts with lots of guidance, first showing learners exactly what to type, then quickly encouraging them to modify and expand their apps by typing in new commands.

What kinds of kids can use Bitsbox?

We developed Bitsbox for kids as young as seven. With younger kids (first through third grades), it's helpful to have an adult close by, but older kids can build apps entirely on their own. No previous programming experience is needed.

What do I need to use Bitsbox in my classroom?

Bitsbox is an online application that runs entirely in the computer browser. Here's what you need to use Bitsbox in your classroom:

- One computer for every one or two students (Younger kids do very well in "pair programming" setups where they can help each other out, while older students should probably have a machine all to themselves.)
- A modern web browser. (Google Chrome is best, but Firefox, IE11, and Safari work okay, too.)
- A reliable (though not necessarily lightning-fast) internet connection
- A physical keyboard and mouse for each computer
- A printed handout (included below) for each student, or one of our activity books

How do I get my students started with Bitsbox?

Here are the steps to get your students started with the Bitsbox Hour of Code activity:

1. Open Google Chrome, Firefox, Safari, or IE11+
2. Go to the following web address: bitsbox.com
3. Instruct each student to click the "Get Started" button on the screen.

Can I set up accounts for my students?

Yes! Just follow the instructions here: bitsbox.com/tools

When they're ready to start coding have them go to bitsbox.com, click "Get Started", then click "Kids Sign In"

Why can't my students do Bitsbox entirely on their iPads?

The apps that kids build with Bitsbox are able to be run on tablets, but the actual creation of the apps (the typing and testing part) is best done on a computer with a physical keyboard.

What programming language are students learning with Bitsbox?

Bitsbox uses a special set of commands that are written in JavaScript, one of the most popular programming languages in the world. Bitsbox's commands are short, easy to type, and easy to understand for beginners. As kids progress, they're introduced to more "raw" JavaScript syntax (vocabulary and grammar). We like to say it's a "Dick and Jane" approach to teaching language.

How can my students run their apps on a tablet?

Kids can easily run the apps they build on smartphones and tablets. Bitsbox apps are really just web pages; to run them on a device, you open the webpage in that device's web browser. The easiest way to "send" the web address to the device is by using a QR code reader to scan it right off the student's computer screen. Follow these steps:

1. Install a QR code scanner app on your tablet or phone. (Visit bitsbox.com/qrapps for help)
2. On a computer, open the student's app in Bitsbox.
3. Click the share icon in the upper right hand corner of the screen.



You should see a small, black and white QR code in the upper-right corner of the computer screen.

4. Use the QR code scanner app on your tablet or phone to scan the QR code on the student's computer screen.

What happens after kids finish the apps on the website?

The apps on the Bitsbox website are only a starting point: Print out copies of the extra apps included at the end of this Teacher's Guide. As kids finish the apps on the website, let them choose what to build next. You can also order classroom kits with lots of apps at bitsbox.com/teachers

Do you have any documentation available?

Yes! To see a list of all available fills, stamps, colors, and sounds as well as all the commands that we use, click the library icon in the upper right hand corner while you're coding an app.



Click on any item or the run button in the documentation to see examples appear on your tablet. Don't worry, once you close the documentation, your code is still there.

Who created Bitsbox?

Bitsbox was invented in 2014 by a pair of Boulder, Colorado-based ex-Google employees named Scott Lininger and Aidan Chopra. They both have kids who would like to learn to code. The Bitsbox website is actually only half of the idea; the other half is a subscription box which arrives at subscribers' homes every month. Learning from print? What an outrageous idea!

As a teacher, how can I stay in the loop about Bitsbox and education?

Go to bitsbox.com/teachers and sign up for our educator-specific mailing list.

How do I contact the Bitsbox team?

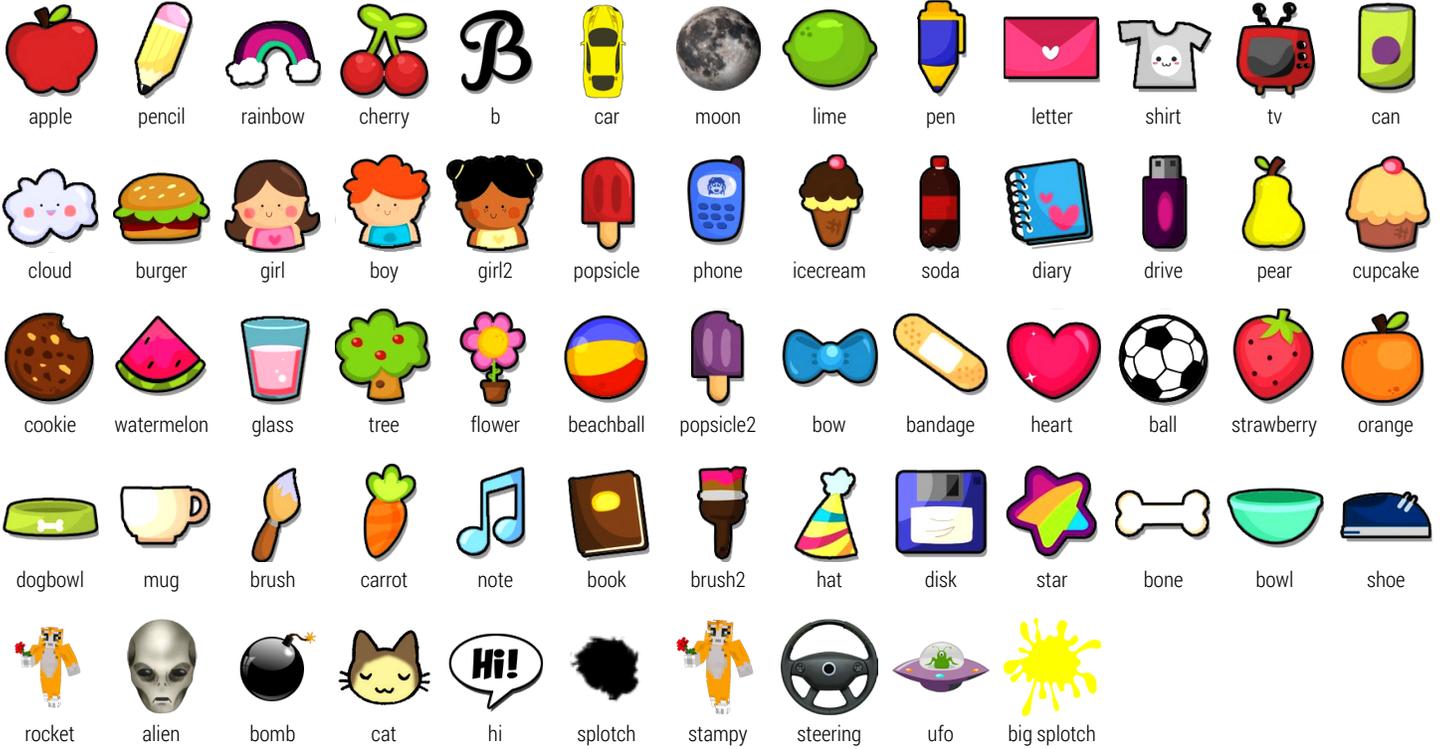
Thanks! If you have any questions or ideas, please email scott@bitsbox.com. We love feedback and bug reports. There's also a little feedback widget in the lower-right corner of bitsbox.com; if you prefer, you can use that instead.

THANKS FOR TRYING BITSBOX WITH YOUR KIDS!

MORE APPS!

The apps on the Bitsbox website are only a starting point: Print out copies of the extra apps on the following pages. As kids finish the apps on the website, let them choose which ones they'd like to build next. Be sure to give each kid a copy of the Pictures & Sounds page, too. Have fun!

STAMPS



FILLS



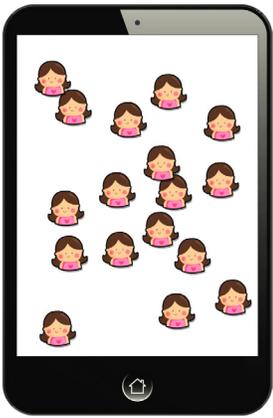
SOUNDS

bad
 blip
 coin
 heart
 hit
 hurt
 jump
 nuke
 power

SONGS

brothers
 forces
 dreaming
 grand

1 Tap the screen to stamp a picture. Tap, tap, tap to create a crowd!



Stamper Girl

```
1 function tap() {  
2   stamp('girl',x,y)  
3 }
```

Challenges

- Can you replace the girl with something else?
- Can you add a line of code that sets the background fill color to yellow?

2 Stamp a picture in the middle of the screen.

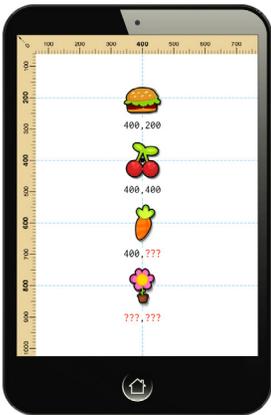


Big Apple

```
1 fill('green')  
2 stamp('apple',500)
```

Challenges

- Can you replace the apple with something else?
- Can you change the fill color of the background?
- Can you make the apple smaller? *Hint: The apple is 500 pixels wide.*

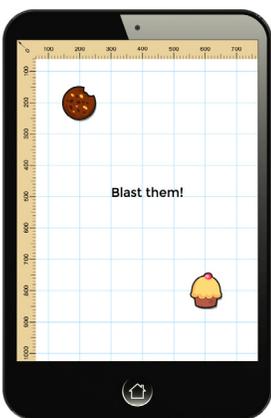


Corner Store

- 1 `fill('ruler11')`
- 2 `stamp('burger', 400, 200)`
- 3 `stamp('cherry', 400, 400)`
- 4 `stamp('carrot', 400, ???)`
- 5 `stamp('flower', ???, ???)`

Challenges

- Can you figure out how to draw the carrot by finishing line 4 of the code?
- Can you figure out how to draw the flower?



Dessert Blaster

- 1 `fill('ruler 3')`
- 2 `text('Blast them!', 300, 500)`
- 3 `stamp('cookie', 200, 200).tap = explode`
- 4 `stamp('cupcake', 600, 800).tap = explode`
- 5
- 6

Challenges

- Can you change the text to say something else?
- Can you change the background to a color?
- Can you add more exploding objects to the screen?



♥ Looking for Love

```
1 fill('park')
2 girl = stamp('girl 2')
3 boy = stamp('boy')
4
5 function tap() {
6   girl.move(x,y,250)
7   boy.move(x,y,2000)
8 }
```

Challenges

- Can you make the boy chase something else?
- Can you make the girl move faster? *Hint: Her speed is 250.*

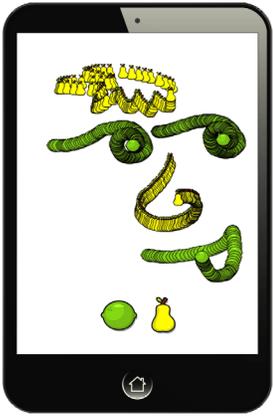


⚽ Soccer Draw

```
1 function drag() {
2   stamp('ball',x,y,80)
3 }
4
5 text('Drag to draw!',50,100,'orange')
```

Challenges

- Can you make the ball bigger? *Hint: The ball is 80 pixels wide.*
- Can you draw with something besides a soccer ball?
- Can you change the color of the text from orange to green?



Fruit Painter

```
1 ink = 'lime'  
2 stamp('lime',325,900).tap = swapInk  
3 stamp('pear',475,900).tap = swapInk  
4  
5 function swapInk() {  
6   ink = this.name  
7 }  
8  
9 function drag() {  
10  stamp(ink,x,y,50)  
11 }
```

Challenges

- Can you draw with different stamps?
- Can you change the size of the paintbrush? *Hint: The paintbrush is 50 pixels wide.*



321 Blastoff!

```
1 fill('desert sky')  
2 ship = stamp('rocket',384,900)  
3  
4 function tap() {  
5   ship.move(384,-100,1000)  
6   sound('rocket')  
7 }
```

Challenges

- Can you make something else blast off instead?
- Can you change the background to stars?
- Can you make the rocket move faster? *Hint: The speed is 1000 milliseconds.*



Galactyroids

```
1 song('forces',50)
2 fill('stars')
3 points = 0
4 score = text(0, 375, 60, 'white')
5 moon = stamp('moon')
6
7 moon.touch = boom
8
9 function boom() {
10   points = points + 1
11   moon.explode(newMoon)
12   sound('nuke')
13 }
14
15 function newMoon() {
16   score.change(points)
17   moon.move(random(700), random(1000), 1000)
18   moon.unhide()
19 }
```

Challenges

- Can you change the moon to something else?
- Can you change the background to something else?
- Can you make the music louder? *Hint: The volume is 50.*
- Can you figure out how to get more points for every moon you blast? *Hint: Try changing line 10.*

Here are some of the commands you can use inside Bitsbox.

<code>box(10,10, 20,30)</code>	<i>draws a box with its top, left corner at 10,10 sized 20x30 pixels</i>
<code>box(1,1,9,9, 'red', 'tan')</code>	<i>draws a small box that's red with tan lines</i>
<code>fill('yellow')</code>	<i>fills the background with a named color</i>
<code>fill('stars')</code>	<i>fills the background with an image</i>
<code>fill(100,25,0)</code>	<i>fills the background with an r,g,b color</i>
<code>line(100,100)</code>	<i>draws a line to position 100,100</i>
<code>line(40,10, 500,10)</code>	<i>draws a horizontal line from 40,10 to 500,10</i>
<code>line(40,400, 'blue')</code>	<i>draws a blue line to 40,400</i>
<code>line(60,10,5)</code>	<i>line to 60,10 that's 5 pixels thick</i>
<code>line(60,10,5, 'red')</code>	<i>line to 60,10 that's 5 pixels thick and red</i>
<code>roll = random(6)</code>	<i>rolls a random number between 1 and 2</i>
<code>roll = random(50,100)</code>	<i>rolls a random number between 50 and 100</i>
<code>silence()</code>	<i>turns off all sounds and songs</i>
<code>song('brothers')</code>	<i>plays the song called 'brothers' at full volume</i>
<code>song('brothers', 50)</code>	<i>plays the song called 'brothers' at half volume</i>
<code>sound('coin')</code>	<i>plays the coin sound</i>
<code>stamp('cat')</code>	<i>stamps a cat</i>
<code>stamp('cat', 100)</code>	<i>stamps a cat that's 100 pixels across</i>
<code>stamp('cat', 200, 400)</code>	<i>stamps a cat at screen location 200,400</i>
<code>stamp('cat', 200, 400, 500)</code>	<i>stamps a 500 pixel cat at 200,400</i>
<code>text('hello')</code>	<i>draws the text 'hello' on the screen</i>
<code>text('hello', 50)</code>	<i>draws the text 'hello' at 50 pixels tall</i>
<code>text('hello', 100, 'red')</code>	<i>draws the text 'hello' at 100 pixels tall in red</i>
<code>sue = stamp('cat')</code>	<i>stamps a cat with the name "sue" so we can use the following...</i>
<code>sue.move(10,10)</code>	<i>move sue to screen position 10,10</i>
<code>sue.move(20,20,500)</code>	<i>move sue to position 20,20 over 500 milliseconds</i>
<code>sue.move(LEFT, 50)</code>	<i>move sue LEFT 50 pixels (try UP, DOWN, NORTH, etc.)</i>
<code>sue.rotate(90)</code>	<i>rotate sue to 90 degrees</i>
<code>sue.rotate(45,500)</code>	<i>rotate sue to 45 degrees over 500 milliseconds</i>
<code>sue.rotate(LEFT, 10)</code>	<i>rotate sue left (counter clockwise) by 10 degrees</i>
<code>sue.size(500)</code>	<i>resize sue to 500 pixels</i>
<code>sue.size(500,1000)</code>	<i>resize sue to 500 pixels over 1000 milliseconds</i>
<code>sue.hide()</code>	<i>hide sue</i>
<code>sue.unhide()</code>	<i>unhide sue</i>
<code>sue.explode()</code>	<i>explode poor sue, then hide her</i>
<code>sue.dance()</code>	<i>make sue dance</i>
<code>sue.tap = function() {</code>	<i>every time we tap sue, she moves upward</i>
<code>this.move(UP, 50)</code>	
<code>}</code>	
<code>function tap() {</code>	<i>every time we tap the screen, draw a stamp there</i>
<code>stamp('cat', x, y)</code>	<i>(instead of tap, try drag, touching, touch, untouch, and loop)</i>
<code>}</code>	