

22 amazing apps to code

marvelous MISCHIEF



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

- 1 Find a computer with a physical keyboard.**
The coding part of Bitsbox isn't meant to be done on tablets just yet.
- 2 Open a web browser and go to bitsbox.com**
We recommend Chrome, Firefox, Safari, or Internet Explorer 11.
- 3 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 little QR code in the corner of the screen.
- 3** Scan the QR code with your gadget.

It looks like this!



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

marvelous MISCHIEF

There's a big difference between mischievousness and outright bad behavior. This month's theme is a celebration of all things roguish.

You'll shoot toilet plungers with Panda Kid. You'll kidnap cows. You'll build an app that plays the music from that fox song so many times that the older people in your house will threaten to take away your computer forever.

That's okay. Life requires you to color outside the lines, and we're here to help.

Yours in coding,

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 some of our trickier lines of code
- Suggestions for extending the apps to make them even more fun

mini apps to get you started

4917 regal eagle



```
1 fill('blue')
2 stamp('eagle')
```

Now can you change the fill color to gold?

2304 hello city

```
1 fill('city')
2 text('hello!')
```

How would you make the screen say your name?



5386 what does the fox say?



```
1 stamp('foxboy',500)
2 sound('fox')
```

Try changing the number on line 1 from 500 to 100. What happens?

Can you add a line of code that fills the screen with purple??

2628 birdy boogie



```
1 song('brothers')
2 stamp('bird').tap = dance
```



What happens when you tap the bird?

Can you change the bird into a ladybug?

5367

dress up ned



Start by typing in this code:

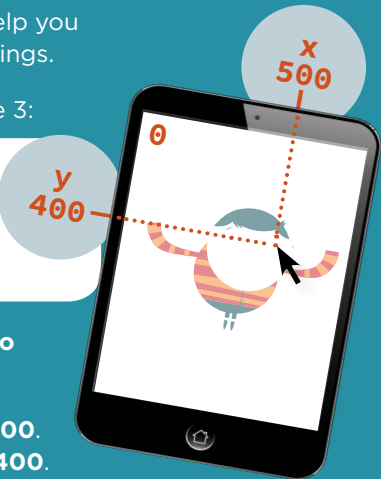
```
1 fill('dress ned')
2 stamp('eye',350,400)
```

Now move your cursor across the tablet on your screen.

Do you see the little orange numbers? They help you figure out where to place stamps and other things.

Draw Ned's other eye. Type more code on line 3:

```
1 fill('dress ned')
2 stamp('eye',350,400)
3 stamp('eye',
```



Next, put your cursor where his eye should go and look at the orange numbers.

The "x" number *above* the screen is close to 500. The "y" number *beside* the screen is close to 400.

Now add these numbers to your code:

```
1 fill('dress ned')
2 stamp('eye',350,400)
3 stamp('eye',500,400)
```

Can you add more stamps to finish dressing Ned?



'hat 3'



'nose'



'eye'



'glasses'

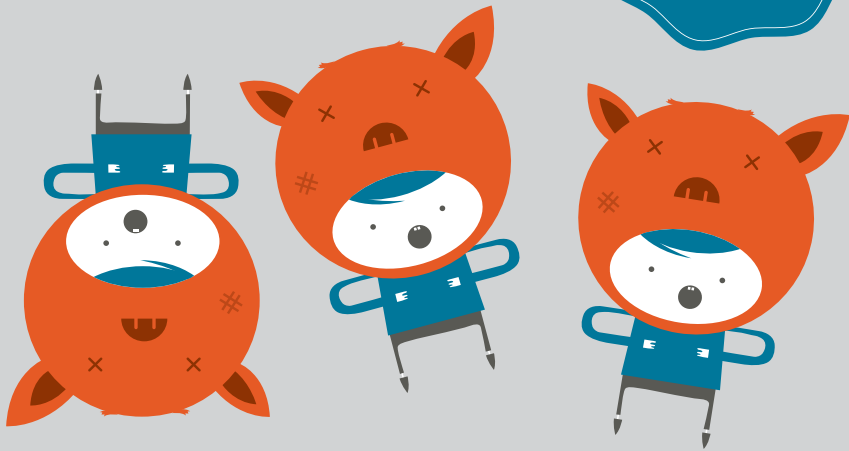


'mustache'

8361

terrible triplets

Bear Boy and his beastly brothers are bouncing all around.



```
1 boy = 'bear head'
2 stamp(boy, 100, 200)
3 stamp(boy, 200, 400)
4 stamp(boy, 300, 600)
```

Save yourself some typing!

See the code on line 1? It means that typing **boy** is the same thing as typing **'bear head'**. Whenever you want to use **'bear head'** in your app, you can just type **boy**. This makes typing lines 2, 3 and 4 much faster!

In this app, **boy** is a *variable*. Variables are super common in coding because they make it easy to make big changes very quickly.

Try changing **'bear head'** to **'panda kid'** on line 1. Don't forget to click the green button to run your app again. Aren't variables awesome? Take a look at the back of this book to see what else you can stamp!

4571

dream drawing

Mr. Fancy is a notorious birdbrain.



```
1 fill('dreaming')
2
3 function tap() {
4   stamp('bird2', x, y, 100)
5 }
```

Don't forget the curly bracket on line 5!

This code stamps a bird wherever you tap. What can you draw with only birds?



Tap Tap Drag

Try changing the word **tap()** to **drag()** on line 3. Click the green button to run your app, then drag your cursor across your screen instead of tapping. Whoa! **tap()** and **drag()** are basic Bitsbox commands. They let you tell your app what to do when someone taps or drags on the screen.



Panda Kid's parents said she couldn't shoot arrows, so she raided the bathroom and made the most of it.

2170

plumber's delight

```
1 fill('plunger fun')
2 toy = stamp('plunger',220,675)
3 kid = stamp('pandakid',100,600)
4
5 function tap() {
6   toy.move(x,y,200)
7   sound('plunger')
8 }
```

This number controls the speed of the plunger!



Beware of flying plungers

The `.move` command on line 6 makes the toy shoot across the screen. When you change the number on line 6 from 200 to 800, does the toy move faster or slower?

7080

peekabot

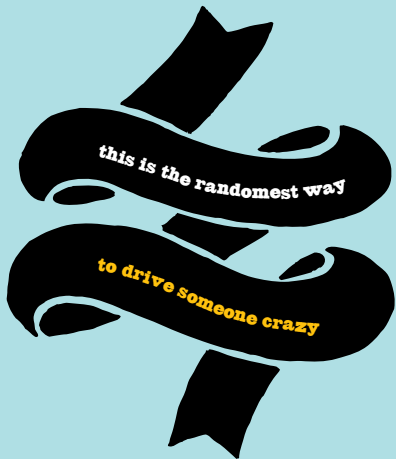
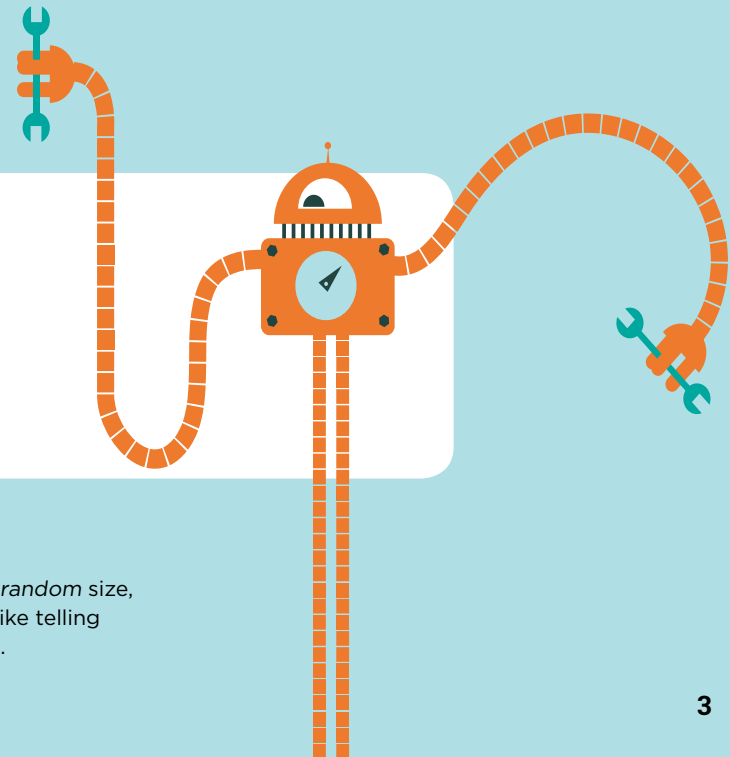
```
1 fill('city')
2 x = random(800)
3 y = random(1000)
4 size = random(100,500)
5 stamp('wrenchbot',x,y,size)
6 sound('peekaboo')
```

This line picks a random number between 1 and 800

This line picks a random number between 100 and 500

That's an awful noise.

Every time you run this app, Peekabot appears at a *random* size, at a *random* spot on the screen. Using `random()` is like telling your app to pick a number by rolling imaginary dice.



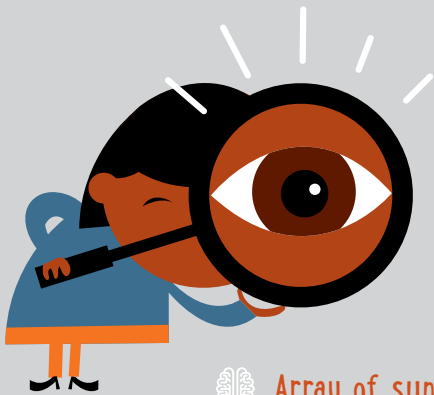
1 9 3 4

Who's my BFF?

Because best friends are forever

Make a list of all your friends, then let this app figure out which one is your BFF.

```
1 friends = []
2 friends[1] = 'Mary'
3 friends[2] = 'Thad'
4 friends[3] = 'Shri'
5
6 pick = friends[random(3)]
7
8 text('My BFF is ' + pick)
```



Can you change this app to include your own friends' names?

 Array of sunshine on a cloudy day.

When you're coding, sometimes it's handy to make lists of things. These lists are called arrays. `friends` is an array that contains the words 'Mary', 'Thad', and 'Shri'. Line 6 picks one of these names at random, and line 8 writes the complete sentence on the screen.

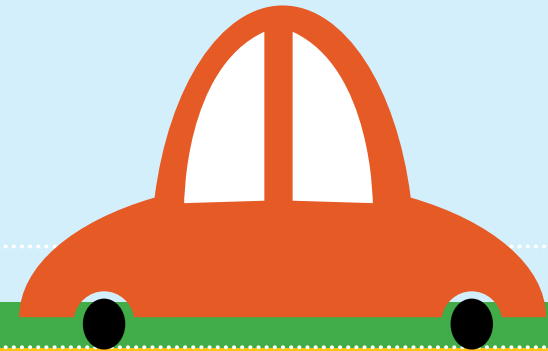
3 6 1 7

road racer

peel out!

Watch (and hear) the car race across the screen.

```
1 fill('road')
2 car = stamp('car2',1,800)
3 speed = 5
4
5 function loop() {
6   car.move(RIGHT,speed)
7   sound('putt putt')
8 }
```



 Feeling loopy?

Use a *loop* whenever you want something to keep happening over and over and over. This car looks like it drives across the screen once, but it actually goes forever. The `loop()` is what makes it keep moving to the right, 5 pixels at a time, again and again. The sound effect is looping, too.

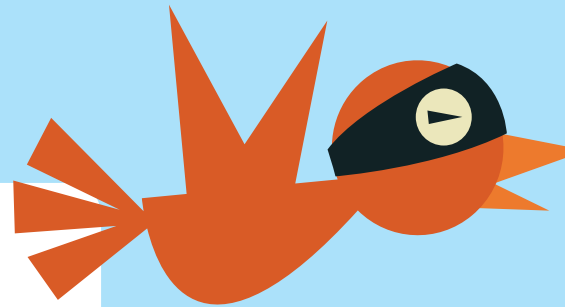
3019

the pirate code

```

1 fill('treasure map')
2 password = prompt('What be the secret word?')
3 if (password == 'please') {
4   stamp('pirate x',290,630)
5   text('X marks the spot!',115,720)
6 } else {
7   text('Wrong word, ye scallywag!',150,420)
8 }

```

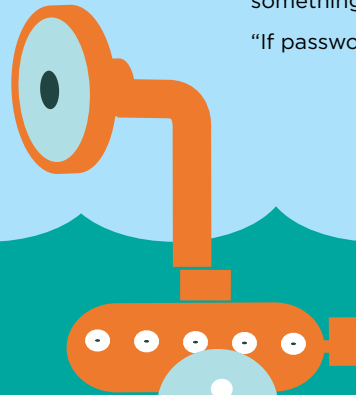


What be the secret word?

Change the text to make it funnier!

🧠 If this be true...

Use **if** when you want some of your code to run only if something is true. In plain English, Line 3 reads: "If password is exactly the word please, do the following".



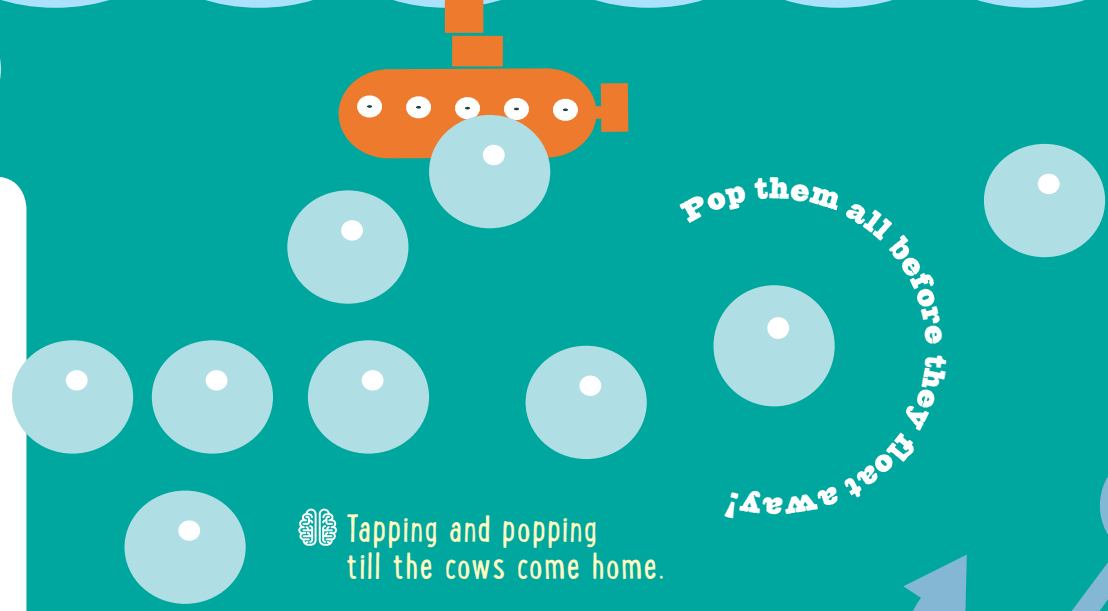
8161

bubble pop

```

1 fill('underwater')
2 size = 70
3
4 function create() {
5   x = random(800)
6   y = random(200,1000)
7   bubble = stamp('bubble',x,y,size)
8   bubble.move(x,0,9000)
9   bubble.tap = pop
10 }
11
12 repeat(create,20)

```



What happens when you change **pop** to **explode**?

🧠 Tapping and popping till the cows come home.

Line 7 creates a new bubble called **bubble**. Line 9 tells that bubble to **pop** when you **.tap** it.



6488

quick paint

Code yourself a creative app that lets you paint anything you want!

```
1 fill('white')
2 size = 10
3 circle(200,990,20)
4 palette = stamp('palette',100,930)
5
6 function drag() {
7   circle(x,y,size/2)
8   line(x,y,size)
9   palette.front()
10 }
11
12 function touch() {
13   line(x,y,0)
14 }
15
16 function change() {
17   color(look(x,y))
18   circle(200,990,20)
19 }
20
21 palette.tap = change
```

This makes
sure your
palette is
always in
front of the
picture you're
painting

SUPER ADVANCED CODE CHALLENGE

Can you add code that lets the artist make their brush bigger and smaller?

If a picture's worth a thousand words, an app that lets you paint a picture is worth even more.



Look Ma! No code!

See the **look** command on line 17? It “looks” at the color where you’ve just tapped and makes that the color for every line and shape you draw afterwards.

7886

slinky sculptor

```
1 fill('boardwalk')
2 size = 150
3
4 function drag(info) {
5   ring = stamp('ring',x,y,10)
6   ring.rotate(info.angle)
7   ring.size(size)
8 }
```



You can change line 5 to stamp anything you want. We think pigs make good art, but that's just a suggestion.

A real Slinky® is fun, but what if you had one that was infinitely long?

Short and sweet, but tricky.

This app is only 8 lines long, but there's a lot going on. Since we have some room on this page, let's take a closer look at the code:

See the word **info** between the parentheses on line 4? It's keeping track of the direction that you're dragging when you're using the app. It's stored as an *angle*.

Now look at line 6. This code rotates the **'ring'** you just stamped using—you guessed it—the angle it sees between the parentheses on line 4.

a brief history

OF COMPUTING

1830s



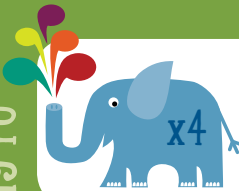
Charles Babbage designs the world's first computer. He calls it the Analytical Engine, but it isn't built in his lifetime.

Ada Lovelace becomes the world's first computer programmer by designing an algorithm (a program) for Babbage's computer.



1843

1946



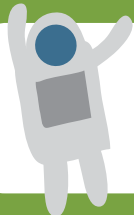
ENIAC is the world's first electronic computer. It weighs more than four adult elephants and is longer than a tractor trailer.

Grace Hopper is a critical member of the team that invents one of the first human-readable coding languages: COBOL. Years earlier, she invented the first compiler for a coding language.



1959

1969



Computers at **NASA** help to land people on the moon for the first time. These machines had less computing power than some of today's toasters.

Researchers at **Xerox PARC** develop the first GUI (graphical user interface) for a computer. We can thank them for the mouse!



1973

The **Apple Watch** hits stores in April. Its guts are still top secret, but it's millions (maybe even billions) of times faster than ENIAC.



2015



Larry Page and Sergey Brin invent Google. Now it's possible to find anything you're looking for on the web—in less than a second.

1996

Tim Berners-Lee invents the World Wide Web. Before this, the internet was all text—no pictures, no videos, and certainly no Bitsbox.



1991

1981



The first **Apple Macintosh** computer goes on sale. A current iMac has 10,000 times more memory and is less than half the price.

1977

The **TRS-80** goes on sale as one of the first personal computers. One of the founders of Bitsbox learned to code on a TRS-80 when he was 7 years old!



Need a break from coding?

COLOR ME!

Programming Puzzler

Find the following coding words hidden in the matrix below. They can be printed frontwards, backwards, or even diagonally. The unused letters spell out a hidden message. What is it?

r	t	n	r	m	c	t	x	y	e	t	v	l	h	i
o	a	s	o	i	e	o	e	t	s	a	s	a	e	c
t	r	l	g	i	b	t	a	x	r	e	t	n	t	s
a	p	o	g	s	t	n	h	i	t	p	r	o	m	a
r	l	y	t	o	i	c	a	o	l	a	i	i	o	r
e	u	i	t	d	r	b	n	o	d	r	n	t	d	r
p	b	b	r	h	l	i	o	u	t	a	g	i	n	a
o	a	o	y	e	o	p	t	m	f	m	p	d	a	y
f	o	r	t	r	a	n	e	h	t	e	s	n	r	e
c	t	n	a	t	s	n	o	c	m	t	t	o	t	u
n	r	u	t	e	r	r	e	g	a	e	h	c	a	l
s	y	n	t	a	x	j	a	m	o	r	e	a	p	a
f	i	l	p	b	r	p	d	n	u	o	s	s	v	
h	t	m	l	o	d	h	y	b	a	s	i	c	l	h
j	a	v	a	s	c	r	i	p	t	g	n	o	s	e

algorithm
array
basic
bitsbox
conditional
constant
coordinate
drag
else
fill
fortran
function
hash
html
javascript
logic
loop
method
object
operator
parameter
python
random
return
ruby
song
sound
stamp
string
syntax
tap
text
then
value
variable

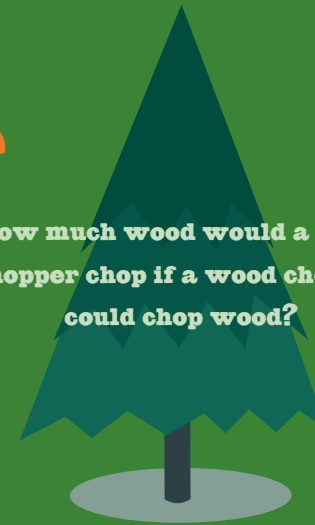
5113

paul bunyan

Chop down all the trees to win, but plan your path carefully—if you end up too far from the next tree, you lose.



How much wood would a wood chopper chop if a wood chopper could chop wood?



Type in this code first!

```

1 fill('maine')
2 chopped = 0
3 goal = 10
4 repeat(tree,goal)
5 paul = stamp('bunyan',200)
6 info = 'Try to chop down all ' + goal
7 note = text(info,175,150,'cornsilk')
8
9 function walk() {}
10
11 function tree(i) {
12   x = random(700)
13   y = 110 + i * 80
14   size = 100 + i * 15
15   pine = stamp('pine',x,y,size)
16   pine.tap = chop
17 }
18
19 function chop() {
20   if (distance(this, paul) > 340) {
21     note.change('Too far away!')
22     return
23   }
24   this.move(x,-200,1000)
25   walk()
26 }

```

Type in this code next!

```

27
28 function walk() {
29   paul.move(x,y,500)
30   paul.change('bunyan chop')
31   chopped = chopped + 1
32   note.change(chopped)
33 }
34
35 function loop() {
36   if (chopped >= goal) {
37     note.change('You win! The Earth weeps.')
38   }
39 }

```

 Make it even better with sound!

Add a line of code `sound('woodchop')` after line 23. Now can you figure out how to make a noise when he's walking?

This app has a lot of code, so we broke it up into two parts. You can type in the first 26 lines and click the green run button, then type in the next 12 to make your game even better.



5170

invasion of the ox snatchers

Move the alien over Babe the Blue Ox
to trap him in your tractor beam.



When alien meets bovine.

How does this app know that it's supposed to pick up Babe when the alien gets close enough? Check out line 14. The `.hits` command checks to see if one object is touching another. In this case, it's checking to see if `ufo2` is touching `ox`. If it is, it runs the code on lines 15 through 21. Uplifting!

Beware
of tractor
beams,
Babe.

```
1 fill('night sky')
2 ox = stamp('ox',100,930)
3 ufo = stamp('ufo2',370,425)
4 direction = RIGHT
5
6 function loop() {
7   if (ox.x > 750) {
8     direction = LEFT
9   } else if (ox.x < 50) {
10    direction = RIGHT
11  }
12  ox.move(direction, 20)
13
14  if (ufo.hits(ox)) {
15    sound('ox')
16    beam = stamp('beam', ox.x, ox.y, 350)
17    ox.move(375,-200,2000)
18    ufo.move(375,-400,2000)
19    beam.move(375,-200,2000)
20    drag = null
21    loop = null
22  }
23 }
24
25 function drag() {
26   ufo.move(x,y)
27 }
```



Play around with the
stamps and sounds in this
app to make your
own version. Check out the
back of this book for ideas.

2725

one man band

Tap the instruments to play them.

```
1 fill('world of music')
2 x = 100
3
4 function add(name) {
5   instrument = stamp(name,x,300,150)
6   instrument.tap = sing
7   x = x + 180
8 }
9
10 add('guitar')
11 add('piano')
12 add('conga drum')
13 add('maracas')
```

Try replacing the instruments with animals to create a Farm Orchestra. Hint: pig, sheep, cow, and chicken are four animals that we like a lot.

Make beautiful music.
Or just music.



Did you know that some stamps can sing? If you look carefully at line 6, you'll notice the command `sing`. The stamps in this app all have their own sound effects.

When you tell `guitar` to `sing`, the sound associated with that stamp plays automatically.

tuba or not tuba

```

1  fill('road')
2  bird = stamp('bird',50)
3  bird.hide()
4  stamp('tubaman2',150,800,250)
5  output = text('0',105,975)
6  robot = stamp('wrenchbot',700,random(1000),150)
7
8  across = 0
9  fall = 0
10 gravity = 4
11 power = 0
12
13 function loop() {
14   bird.move(RIGHT,across)
15   bird.move(DOWN,fall)
16   fall = fall + gravity
17   if (bird.hits(robot)) {
18     robot.explode()
19     sound('hurt')
20   }
21 }
22
23 function touching() {
24   power = power + 1
25   bird.move(180,770)
26   output.change(power)
27 }
28
29 function untouch() {
30   bird.unhide()
31   sound('bloop',power*2)
32   across = power
33   fall = power * -2
34   power = 0
35 }

```

Remake this game using completely different pictures and sounds.

Hold down on the screen to build up your power, then let go to fire a bird at the robot.

This sousaphone is a blast to play.



Touch and untouch.

In Bitsbox, **touching()** happens when someone's holding down their cursor on the screen. At the instant they stop touching, **untouch()** happens.

While we're on the subject, **touch()** happens the instant something is touched, and **tap()** happens when something is touched and then untouches. Whew.

7560

time machine

Welcome to the fourth
dimension, Ned



Travel through time in your very own
interactive time machine.

This app is really long, so we cut it up into three chunks.
Type in one chunk at a time and click the green run button after each.

Type in this code first!

```
1 pilot = stamp('ned',375,600)
2 stamp('time machine2')
3 year = text('2015',225,394,'orange')
4 time = text('0',468,390,'black',30)
5
6 function tap() {
7   date = x + y + 1000
8   year.change(date)
9   time.change(date - 2015)
10 }
```

These lines stamp the basic graphics and change the date when you tap anywhere on the screen.

Type in this code second!

```
11
12 flag = stamp('flag',380,240)
13 flag.tap = dance
14 pilot.tap = sing
15
16 function zap() {
17   sound('blip')
18   rings = stamp('rings',270,240)
19   rings.move(270,-100,1000)
20 }
21 stamp('antenna',270,240).tap = zap
```

This code adds a flag and an antenna, and tells each what to do when they're tapped.

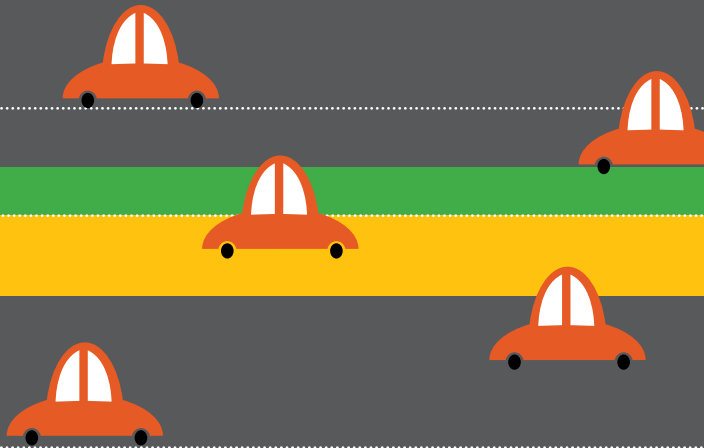
Type in this code last!

```
22
23 function spin() {
24   t = random(360,720)
25   clock.rotate(RIGHT,t,2000)
26   minutes.rotate(RIGHT,t*2,2000)
27   sound('coin')
28 }
29
30 clock = stamp('antenna',568,750,60)
31 minutes = stamp('antenna',568,750,70)
32 clock.tap = spin
```

This section stamps the hands of the clock and makes them spin.

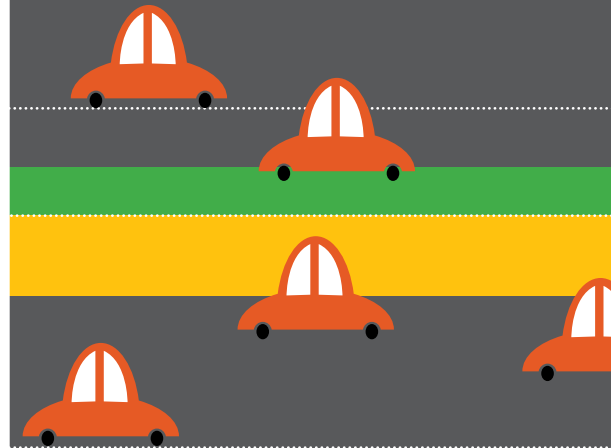
1068

look out, sam!



```
1 fill('buslane')
2 sam = stamp('cyclist',100,900,150)
3
4 function drag() {
5   sam.move(x,y)
6 }
7
8 function spawn(name) {
9   x = 900
10  y = random(1000)
11  racer = stamp(name,x,y)
12  speed = 2000 + random(2000)
13  racer.move(-150,y,speed)
14 }
15
16 function loop() {
17   if (random(7) == 1) {
18     spawn('racer')
19   }
20   if (sam.hits('racer')) {
21     sam.pop()
22   }
23 }
```

Help Sam the cyclist ride into oncoming traffic without getting hit. Seriously, Sam?



The cars just keep on coming.

This app uses **loop** to create an endless stream of cars. On line 17, try changing the 7 to a 14. Does this make the game easier or harder? Here's what's happening:

The code inside the **loop()** command runs 20 times per second—all loops do, in fact. If a new car were created that frequently, Sam would have a wall of cars coming at him. The game would be impossible.

Line 17 reduces the number of cars that are created. Twenty times per second, the code generates a random number between 1 and 7. It *only* creates a new car when the number is exactly equal to 1. That's 1 out of every 7 loops. Changing the 7 to a 14 only creates a new car out of every 14 loops, which is fewer cars.



rocketgirl saves jetkid

Hurry!
There's no
time to
waste!

Help Rocketgirl free Jetkid by lobbing a tiny spaceman at evil Dr. Dastardly's brick wall.

Type in one section of code at a time, then click the green run button after each. The game gets better with every new section you add.



1st

```

1 fill('night sky')
2 dx = RIGHT
3 dy = UP
4 speed = 10
5 ball = stamp('spaceman doll',50,950,80)
6
7 function bounce() {
8   if (ball.x > 750) { dx = LEFT }
9   if (ball.x < 0)   { dx = RIGHT }
10  if (ball.y < 0)   { dy = DOWN }
11 }
12
13 function victory() {}
14 function collide() {}
15
16 function loop() {
17   ball.move(dx,speed)
18   ball.move(dy,speed)
19   bounce()
20   collide()
21   victory()
22 }

```

Make the spaceman bounce
around the screen.
Poor lonely spaceman.

2nd

```
23
24 paddle = stamp('astronaut',50,900,150)
25 kid = stamp('jetkid',390,90,200)
26 words = text('Help me!',495,50,'white')
27
28 function drag() {
29   paddle.move(x,y)
30 }
```

Add Rocketgirl and Jetkid to the app. Characters!

3rd

```
31
32 y = 200
33 repeat(block,15)
34 y = 250
35 repeat(block,15)
36 y = 300
37 repeat(block,15)
38
39 function block(i) {
40   stamp('block',i*48,y,48)
41 }
```

Build Dr. Dastardly's impenetrable brick wall.



Coding onward...

Can you figure out how to make this game even harder?

How would you add a sound effect when the spaceman hits a block?

Hint: 'hurt'



4th

```
42
43 function collide() {
44   if (ball.hits(paddle)) dy = UP
45
46   collision = ball.hits('block')
47   if (collision) {
48     collision[0].hide()
49     dy = DOWN
50     speed = speed + 1
51   }
52 }
```

Let the spaceman bounce off Rocketgirl and break bricks!



5th

```
53
54 function victory() {
55   if (ball.hits(kid)) {
56     loop = null
57     words.change('You saved me!')
58     paddle.dance()
59     kid.dance()
60   }
61   if(ball.y > 1000) {
62     loop = null
63     paddle.explode()
64     words.change('We lost. :(')
65   }
66 }
```

This code controls what happens when you win or lose.

STAMPS

Use these stamps (& fills & songs & sounds) to make any app your own!
Just don't forget to put single quotes around them in your code, like this:

```
stamp('unicorn') fill('park 2') song('forces') sound('roar')
```



astronaut



axe



bird 5



book 2



brush 3



car 2



crossbones



cyclist



eagle



foxboy



hat 3



jet kid



ladybug



lander



lumberjack



mr fancy



mustache 4



ned



note 2



octopus



ox



planet 2



radio



rocket 2



spygirl



submarine



time machine



tree 3



tuba man



ufo 2

SONGS

1812

baby blue

brothers

cafe

dreaming

matrix

obstacles

panda

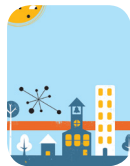
silly



FILLS



castleroad



city



dreaming



maine



mottledsky



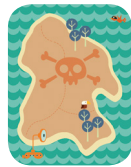
oldwest



plungerfun



road



treasuremap

SOUNDS

alien

bad

bat

bear

bleep

blip

bloop

blp

bird

boing

bubble

chaching

chomp

coin

conga drum

dog

fox

frog

guitar

heart

hit

hurt

jet

jump

maraca

nuke

ox

peekaboo

piano

plunger

pop

power

putt putt

rocket

select

slap

violin



More stuff online!

These are just the tip of the iceberg. Look for more when you're coding at bitsbox.com!