

Knowledge Organiser

Computing

Year 9 Term 3: Mobile Applications



Name/Class:

Enquiry Questions

- Can a computer host multiple applications?
- What is a user interface?
- How do you design a user interface?
- Do instructions given to a computer need to be more precise?
- What are mobile applications used for?
- What are variables?
- Do variables store more than one data item at a time?
- What is a subroutine?
- What is Selection in programming?
- What are logic operators?
- How is music and sound added to applications?

Terminology

Algorithm

An [algorithm](#) is a set of instructions that are followed to solve a problem. It's a computer's thought process.

Arithmetic operators

Arithmetic operators are essential in almost every application, especially in games. If a game character earns experience, it needs to be added to the total earned. If an arrow hits an enemy, the damage the enemy takes needs to be calculated.

Conditional statements

Conditional statements evaluate to true or false. Use them to print information or move programs forward in different situations.

Else statements

Else statements are used to do something else when the condition in the if statement isn't true.

For loops

For loops allow you to run a block of code repeatedly, just like while loops. However, for loops run a block of code a set number of times. (Remember, while loops run an unknown or unspecified amount of times; more on that below.)

Functions

A function is a block of code that can be referenced by name to run the code it contains.

If statements

An if statement runs a block of code based on whether or not a condition is true.

Engage task—Lesson 1

1. Think of 3 apps you use

2. Describe their purpose

3. Who is the target audience for this app?

4. Explain who you believe the intended audience should be

App Lab

Engage task—Lesson 2

App for that –

1: What is the purpose of decomposing a programming project?

- ☐ To break down the problem into smaller parts that are more manageable and easier to understand
- ☐ It is used to test your system at the end of the project
- ☐ To split the problem into two parts
- ☐ To make a hand-drawn design for the solution to the problem before you start writing code

2: Which one of the following success criteria would not be suitable?

- ☐ User screen must have a button to move to the next screen
- ☐ Must allow the user to input their name
- ☐ App must be easy to use
- ☐ App must have a score screen

3: Where would you change the id of a text box or button?

- ☐ In the properties tab
- ☐ In the events tab

Engage task—Lesson 3

Q1. Looking at the code below, what happens after the `confirm_button` has been clicked (Image source: Code.org)

```
onEvent(▼"confirm_button", ▼"click", function() {  
  var name = getText(▼"name_input_box");  
});
```

- ☐ The user is prompted to enter text into the input box
- ☐ A data entry box will appear
- ☐ Nothing, as the variable name hasn't been declared beforehand
- ☐ The text entered by the user into the input box is collected and linked to the name variable

Q2. There is an app that has two screens, `home_screen` and `game_screen`. On the home screen are two buttons. What sequence should the two events below be placed in? (Image source: Code.org)

```
1. onEvent(▼"home_button", ▼"click", function() {  
  setScreen(▼"home_screen");  
});  
  
2. onEvent(▼"play_game_button", ▼"click", function() {  
  setScreen(▼"game_screen");  
});
```

- ☐ The sequence of each `onEvent` is unimportant, as the flow of the program is controlled by when the events are triggered
- ☐ 1 then 2
- ☐ 2 then 1

Q3. Fixing an error in your code is known as:

- ☐ Error detection
- ☐ Error correction
- ☐ Debugging
- ☐ Detection

Engage task—Lesson 4

Q1. In the following image, what type of input option do you see at point A?

A screenshot of a login form titled "Login". It contains a "Username:" label followed by a text input box, and a "Password:" label followed by a password input box. Below the password box are links for "I forgot my password" and "Resend activation email". Further down are two checkboxes labeled "Remember me" and "Hide my online status this session". At the bottom is a red "Login" button. Three blue arrows point to specific elements: arrow A points to the text input box for the username, arrow B points to the "Remember me" checkbox, and arrow C points to the "Login" button.

- ☐ Button
- ☐ Check boxes
- ☐ Text boxes
- ☐ Labels

Q2. In the following image, what type of input option do you see at point B?

- ☐ Button
- ☐ Check boxes
- ☐ Text boxes
- ☐ Labels

Q3. Looking at the code below, what happens after the "login" button has been clicked?

```
onEvent (▼ "login", ▼ "click", function() {  
  var username = getText (▼ "username");  
});
```

- ☐ The user is prompted to enter text into the input box
- ☐ The text entered by the user into the input box is collected and linked to the username variable
- ☐ Nothing, as the variable name hasn't been declared beforehand
- ☐ A data entry box (username) will appear

Engage task—Lesson 5

Q1. What would be displayed in the console if the score variable held the value of 5? (Image source: Code.org)

```
if (score > 10) {  
  console.log("Great Work");  
} else {  
  console.log("Hard Luck");  
}
```

- ☐ Great Work would be outputted
- ☐ Hard Luck would be outputted
- ☐ Both Great Work and Hard Luck would be outputted
- ☐ Nothing would happen

Q2. What would be displayed in the console if the score variable held the value of 10? (Image source: Code.org)

```
if (score > 10) {  
  console.log("Great Work");  
} else {  
  console.log("Hard Luck");  
}
```

- ☐ Great Work would be outputted
- ☐ Hard Luck would be outputted
- ☐ Both Great Work and Hard Luck would be outputted
- ☐ Nothing would happen

Q3. What would be displayed in the console if the score variable held the value of 6? (Image source: Code.org)

```
if (score > 10) {  
  console.log("Great Work");  
} else if (score > 6) {  
  console.log("Not Bad");  
} else {  
  console.log("Hard Luck");  
}  
console.log("Try again");
```

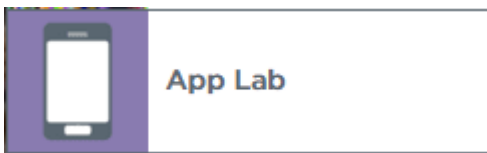
- ☐ Great Work would be outputted
- ☐ Not Bad would be outputted
- ☐ Hard Luck would be outputted
- ☐ Both Hard Luck and Try again would be outputted

The Basics - Creating an Account

Go onto www.code.org

Create ▼

Click create then App Lab.



You then click create an account.

Create an account

You can create the account using your school email address.

18AZmith@thedustonschool.org

Create a password you will remember.

Make sure to select 'student' from the Account Type drop-down on this page.

Think of a username that you will use to sign in with each time.

Fill in the rest of the information and sign up.

Have an account already? Sign in

Email or username

Password

[Forgot your password?](#)

Sign in

Haven't joined yet?

Create an account

Account Type

Email *

Password

Password confirmation

Display Name (e.g. Cool
Coder or Jane S.)

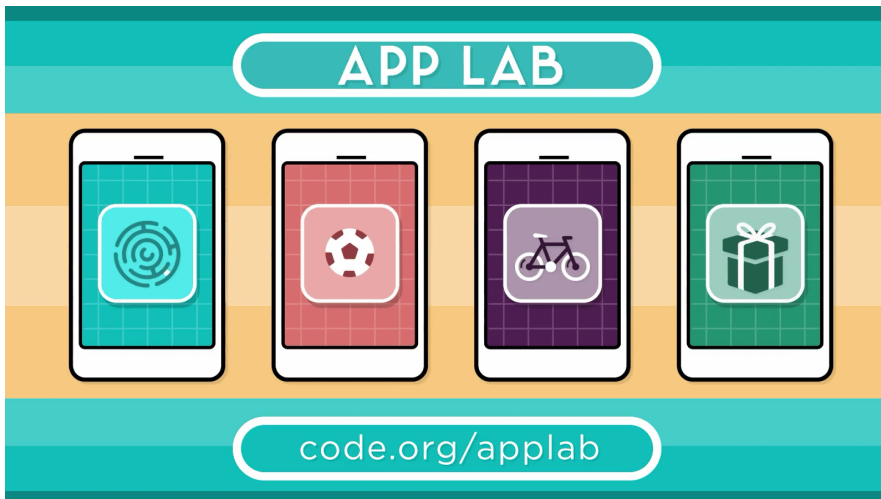
Age

Gender (optional)

☐ I agree to the [Terms of Service](#) and [Privacy Policy](#).

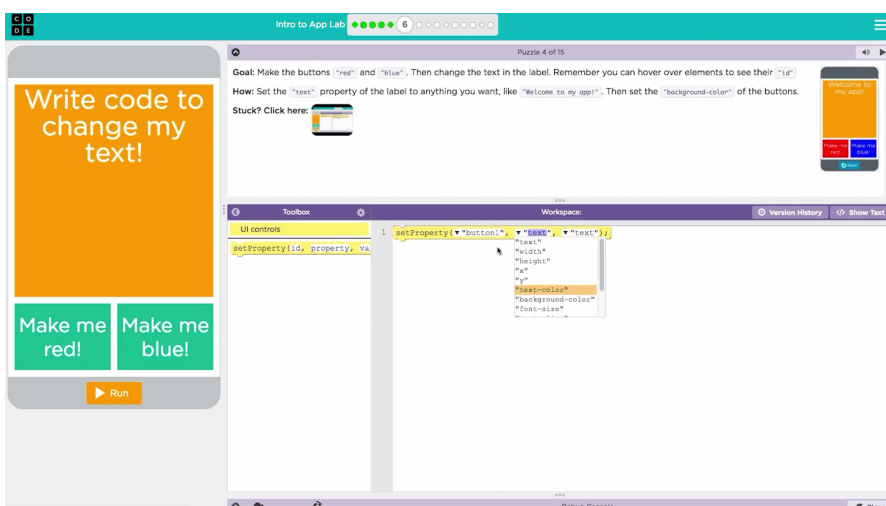
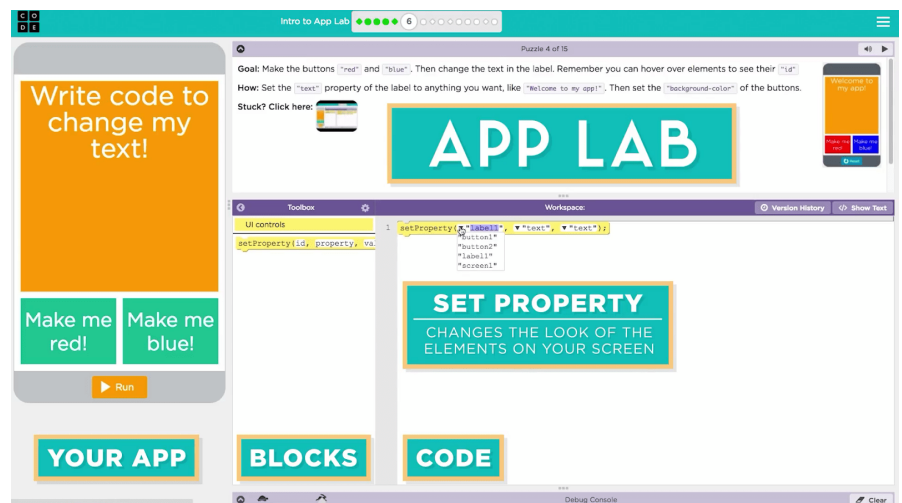
Sign up

Setting Properties



Making your own apps is easy with App Lab! Whether you're new to coding or have some experience. App Lab is a great tool for building new apps and sharing them with your friends.

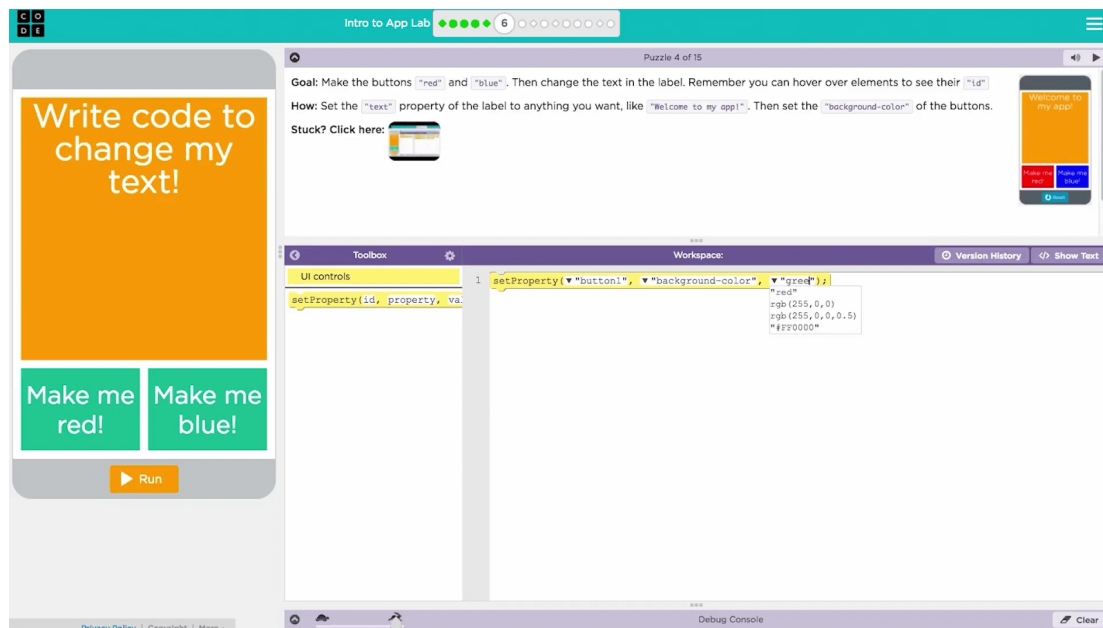
The set property block changes the look of the elements on your screen. Like the buttons, labels, or even the screen itself. First you need to decide which element you want to change. If you hover over an element in your app you can see the name or ID here. Then go select that ID from the first drop down.



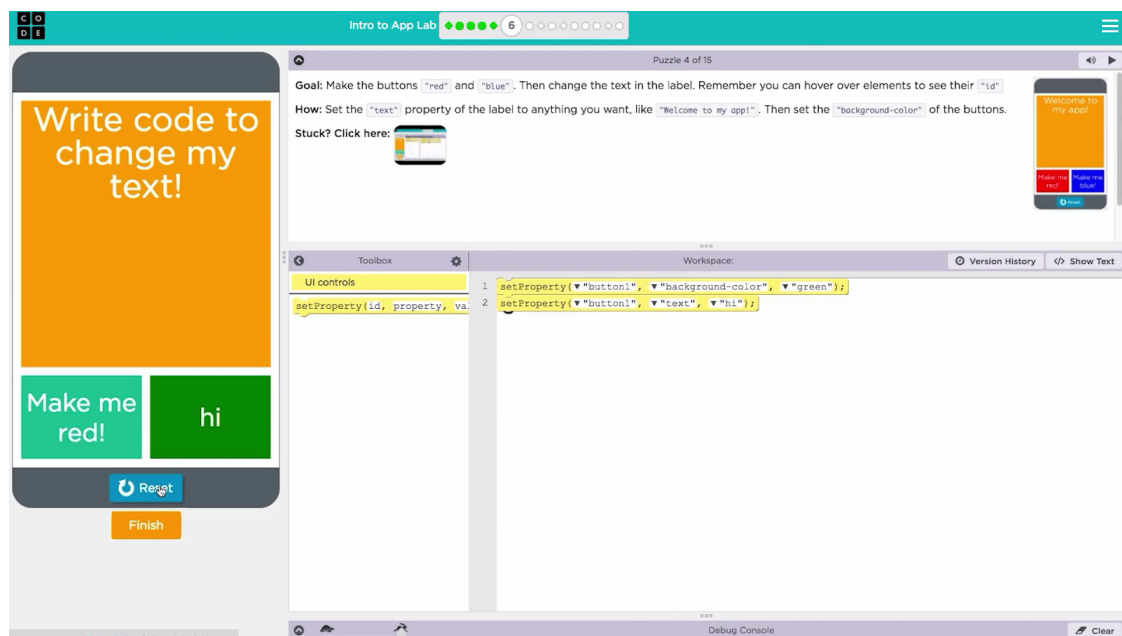
Elements have lots of properties you can change like their text colour, background colour or font size. You can see the full list and choose which property you want to change in the second drop down.

Setting Properties

The last drop down is where you'll write the value you want to use value you want to use. The block will make a suggestion for you. But you can always type in different colours or numbers yourself.

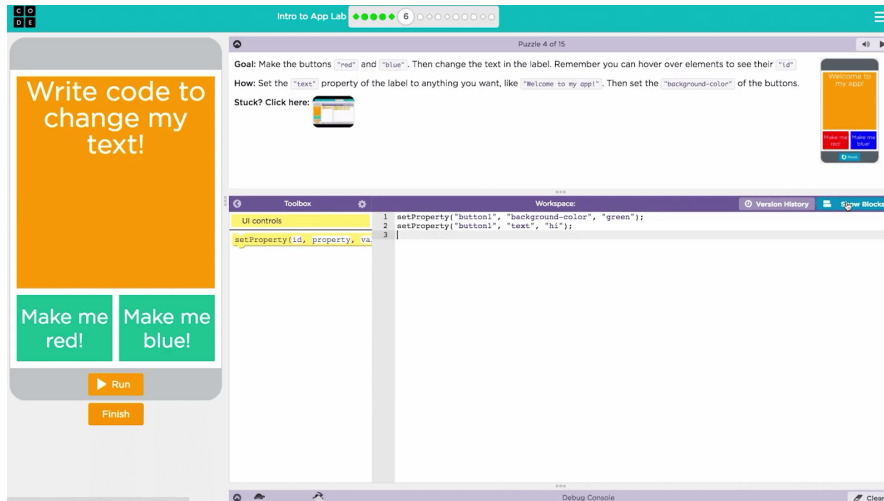


Once you're done, the block reads a bit like a sentence: 'Set button1's background colour to green.' Hit run to see the changes you're building in code. If you reset you can drag in more blocks to change other things about your app. Like the text!



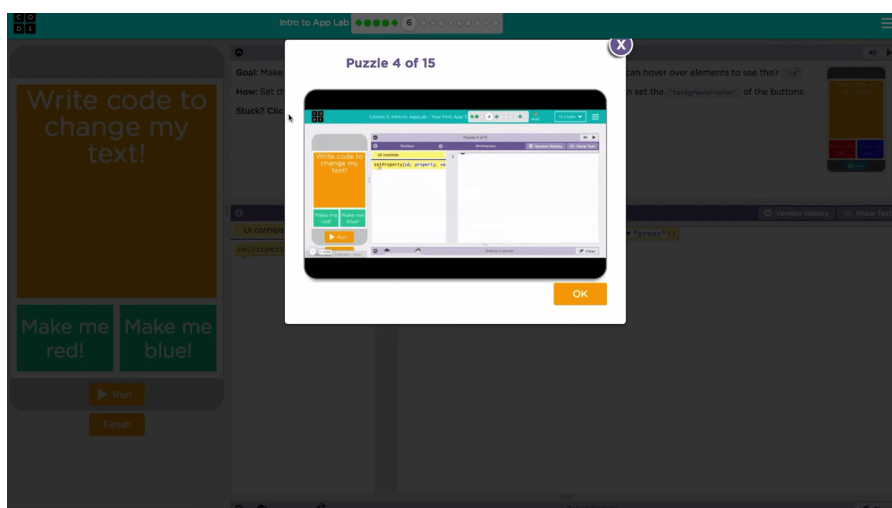
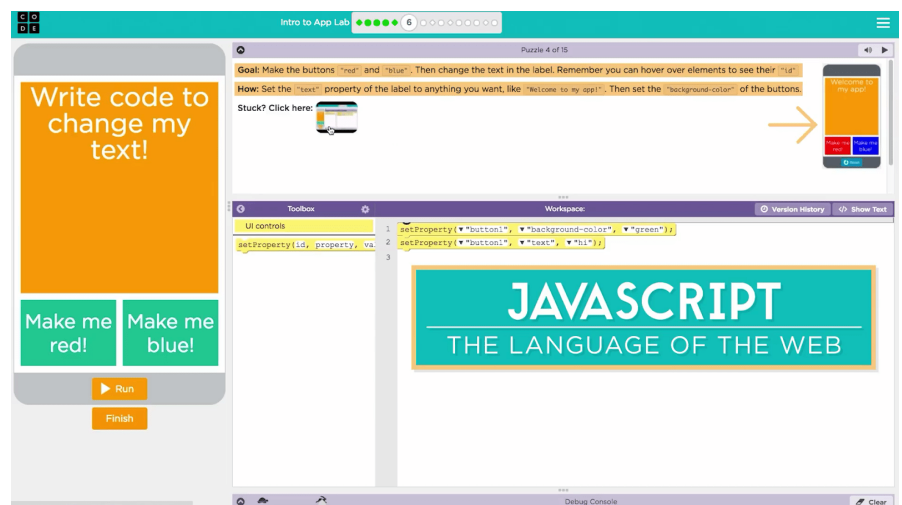
App Lab

Setting Properties



For this tutorial we'll be working in block mode but App Lab also supports working in text.

Either way you'll be programming in JavaScript, the language of the web. Each level has a goal, instructions and an image of what you're aiming to create.



If you're ever stuck, click on the picture and it will show you exactly how to complete the level. But see if you can do it on your own first.

Setting Properties

Goal: Make the screen green.

How: Drag in a `setProperty()` block. Use the dropdowns to set the screen's `"background-color"` to `"green"`. Then hit "Run".

Goal: Make the text bigger and blue.

How: You'll need two lines of code. Give `"label1"` a `"font-size"` of `80` and `"text-color"` of `"blue"`.

Goal: Make the buttons `"red"` and `"blue"`. Then change the text in the large orange label.

How:

- Remember you can hover over elements to see their `"id"`.
- Set the `"background-color"` of the buttons to `"red"` and `"blue"`.
- Set the `"text"` property of the label to anything you want, like `"Welcome to my app!"`.

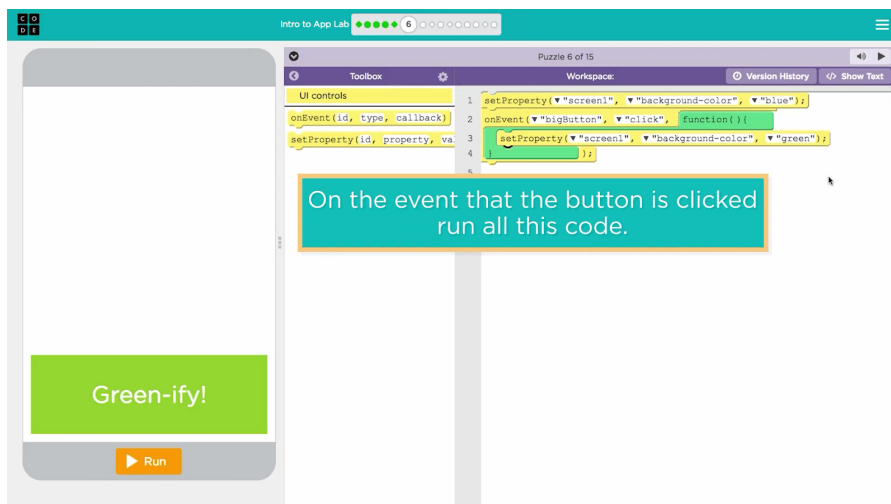
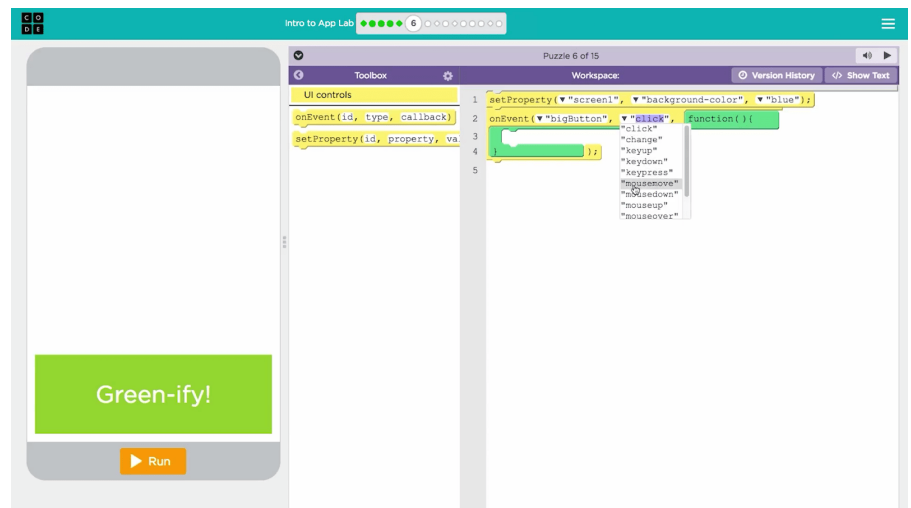


App Lab

Make It Interactive

Now that you can change the look of your app let's make it interactive by responding to events! Events are user actions like clicking a button, scrolling through a menu, or hovering over a picture. Interactive apps need ways to respond to events, like playing a sound when I click this button. To do this in App Lab you need to use a new block called 'onEvent'.

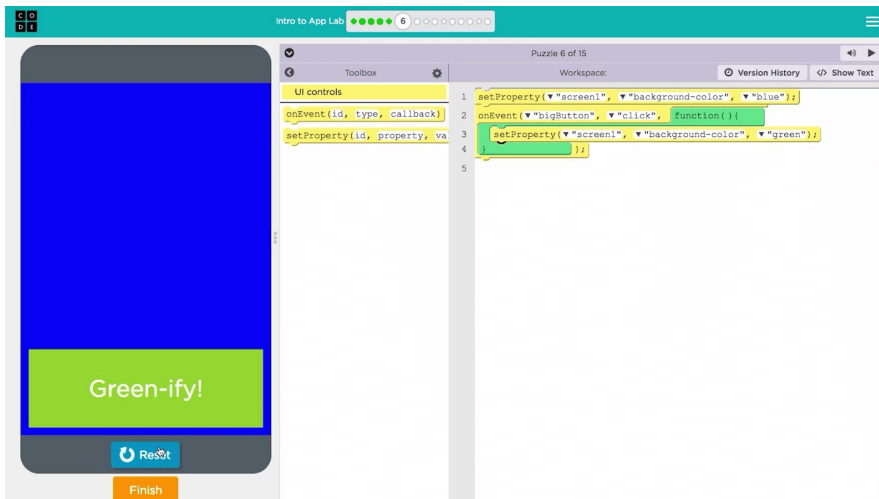
This program sets the screen to blue. I want the screen to turn green when I click this button. First I'll need to drag in an onEvent block. By hovering I know the button's ID is "bigButton" so I'll select that in the first drop-down.



Next I'll choose the type of event. There's lots of options you can pick from but right now I'll leave it as a regular old click.

App Lab

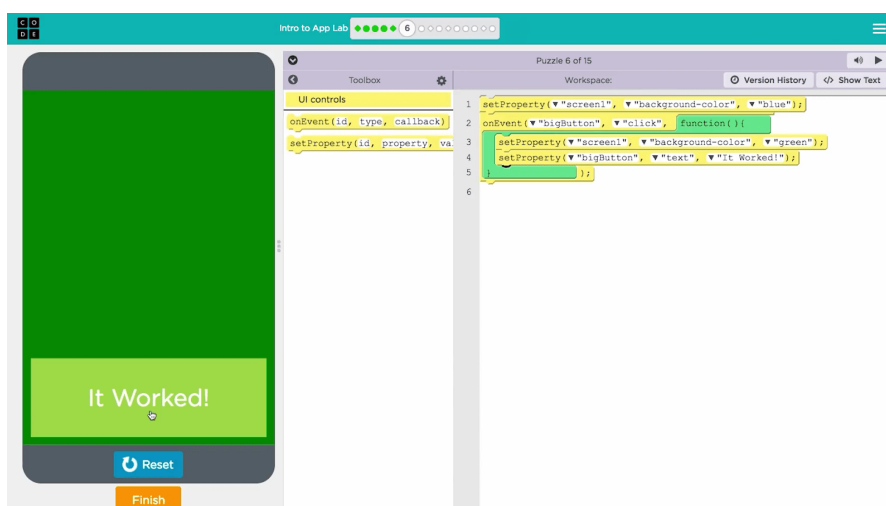
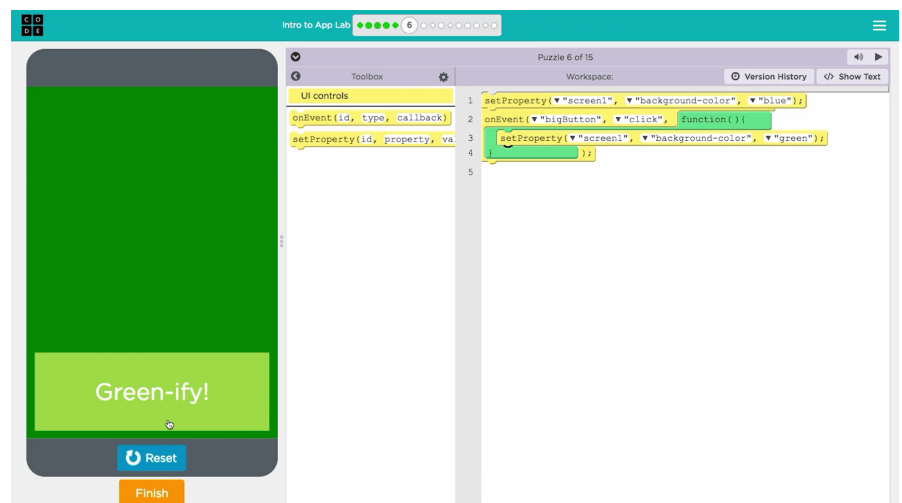
Make It Interactive



Finally I'll add code inside the on event that will change the background colour of the screen. You can read this block like a sentence: "On the event that the button is clicked run all this code."

Let's test it. Blocks outside an onEvent still run right away so the screen starts blue.

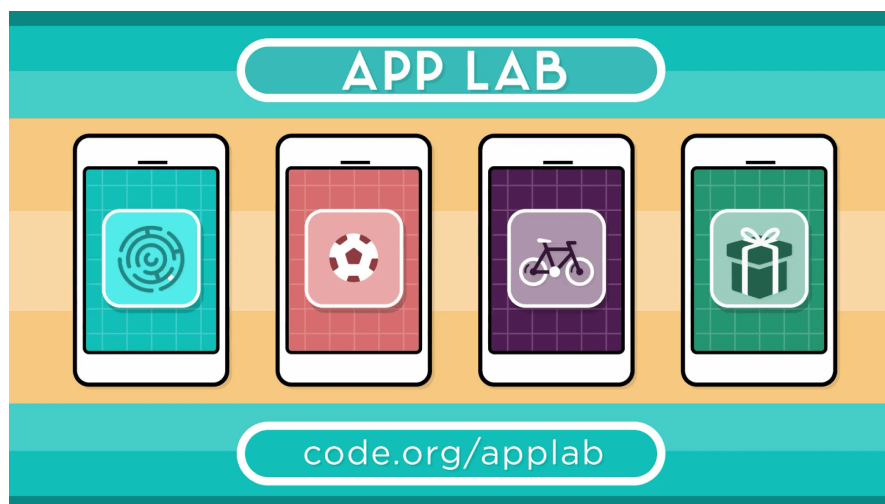
And when I click the big button the code inside my onEvent runs and my screen turns green.



If you want to change more things after the event, like the text on the screen just add more code to the onEvent

App Lab

Make It Interactive



Goal: The screen will start out blue. Add code so it turns green when you click the "Green-ify" button.

How: Add code inside the `onEvent()` that sets the screen's `"background-color"` to `"green"`.

Goal: Help finish this flashlight app. The "On" button already turns the screen white. Run the app and try it out. Then write code so the "Off" button makes the screen black again.

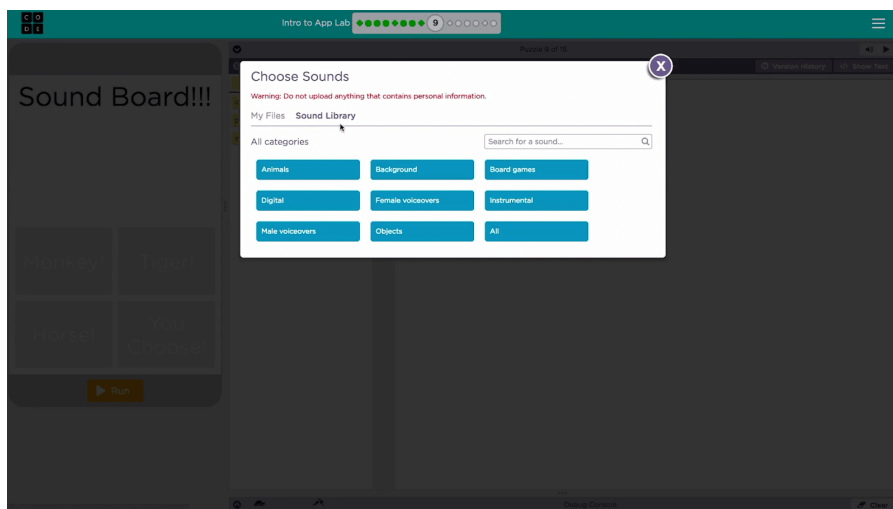
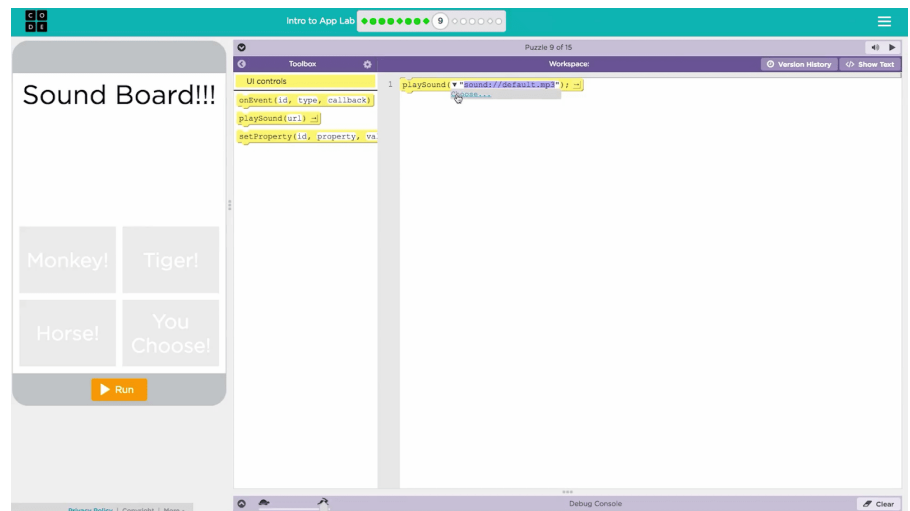
How: Add code inside the empty `onEvent()` that sets the `"background-color"` of the screen to `"black"` and `"text-color"` of the label to `"white"`.

App Lab

Images and Sounds

Next you'll learn about adding images and sounds to your app. Let's start by checking out the sound options.

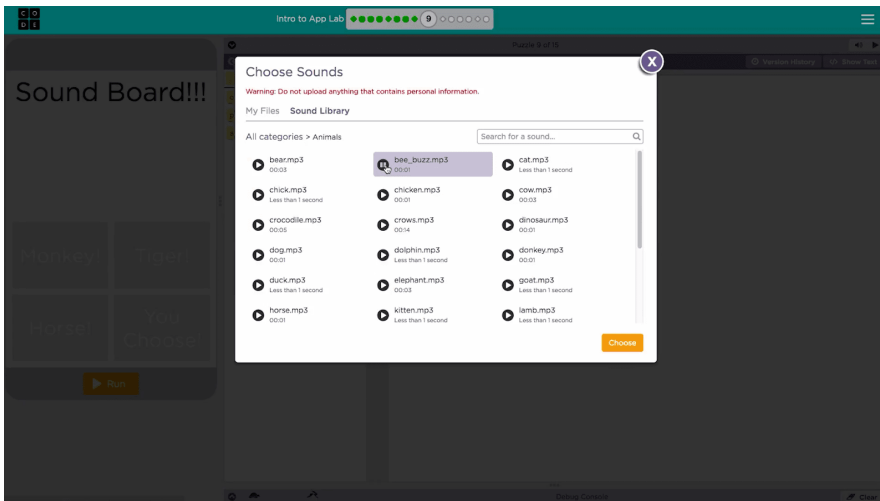
In the tool box you'll find a new block called "playSound". Drag it into the workspace. You can pick a sound to play by clicking the drop-down then clicking "Choose".



From here you can either upload a sound file from your computer, or search for a sound from the sound library. The sound library has lots of different categories like instruments, background music or animals.

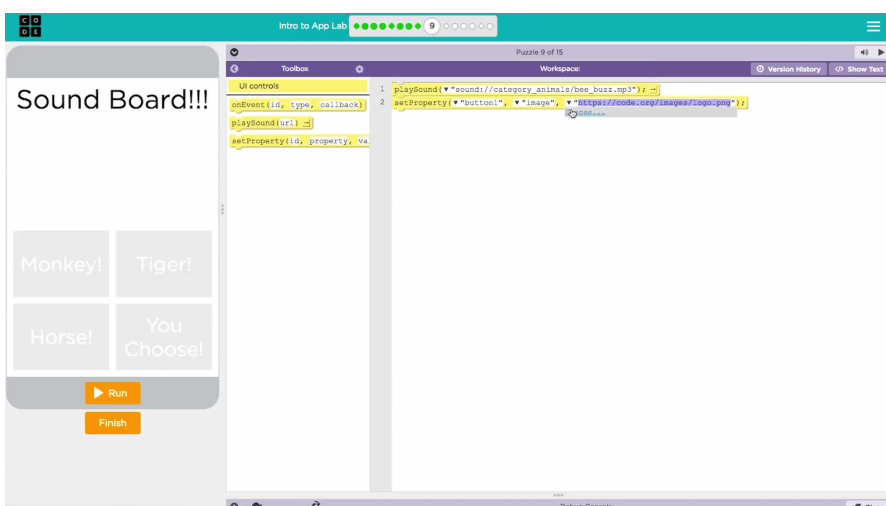
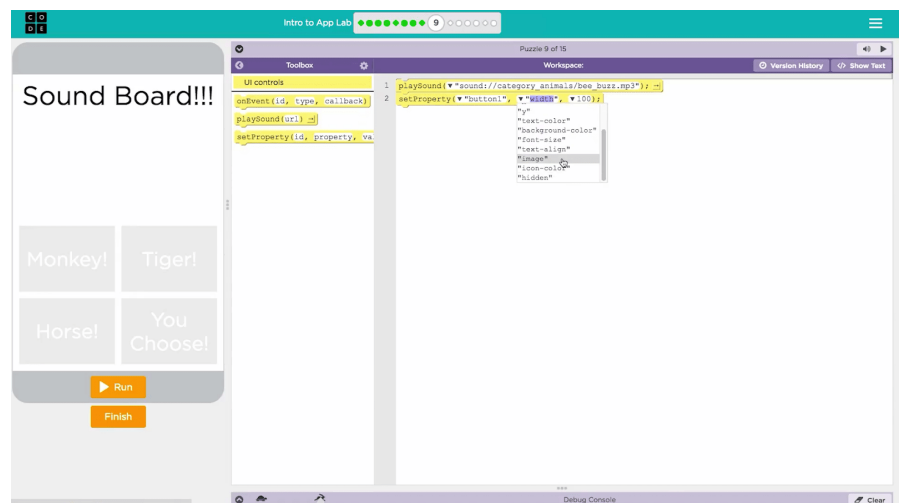
App Lab

Images and Sounds



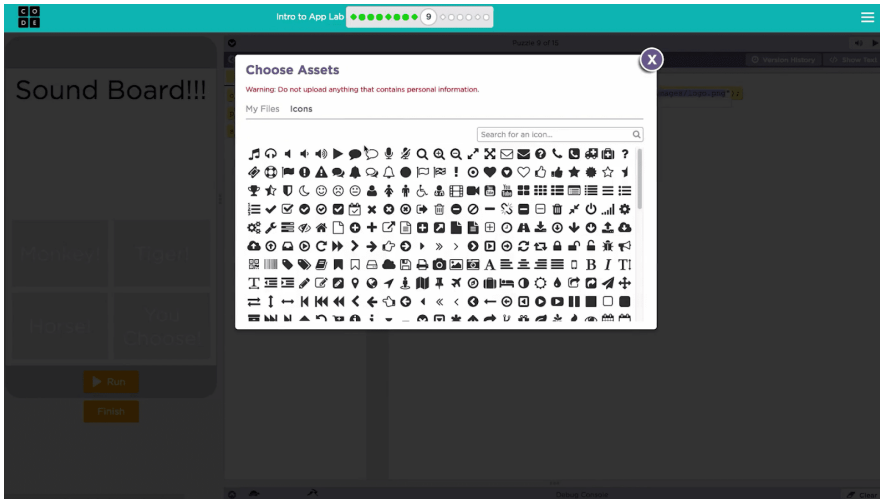
Once you've got the sound you want, click "Choose". When this block runs, it will play the sound you chose.

To add images to your elements you can just use the `setProperty` block. Select the image property in the second drop-down



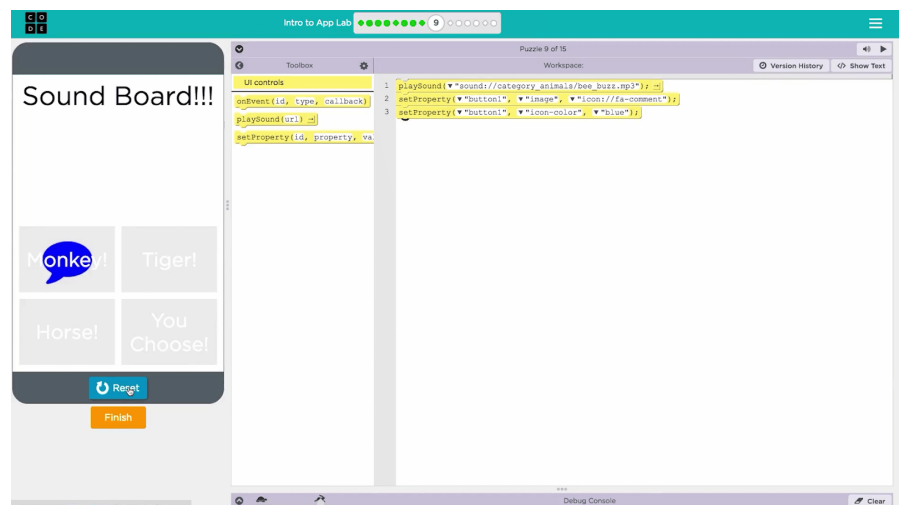
Then select "Choose" from the third drop down.

Images and Sounds



From here you can upload an image from your computer or you can look through a huge library of icons in the icon library.

Back in code mode you can use the "setProperty" block to change the icon color of your icon. Once you've picked what image your icon to use click "Run" to see how it looks.



Goal: This sound board is almost done! Run it to see how it works. Then add an image, text, and sound to the last button.

How:

- Add an `onEvent()` to the program and set the "id" to `"button4"`. Drag a `playSound()` into the `onEvent()` and choose a sound from the library.
- Add a `setProperty()` (outside your `onEvent()`) to change the button's `"image"` property. Choose an image using the third dropdown.
- Add one more `setProperty()` to change the button's `"text"`.

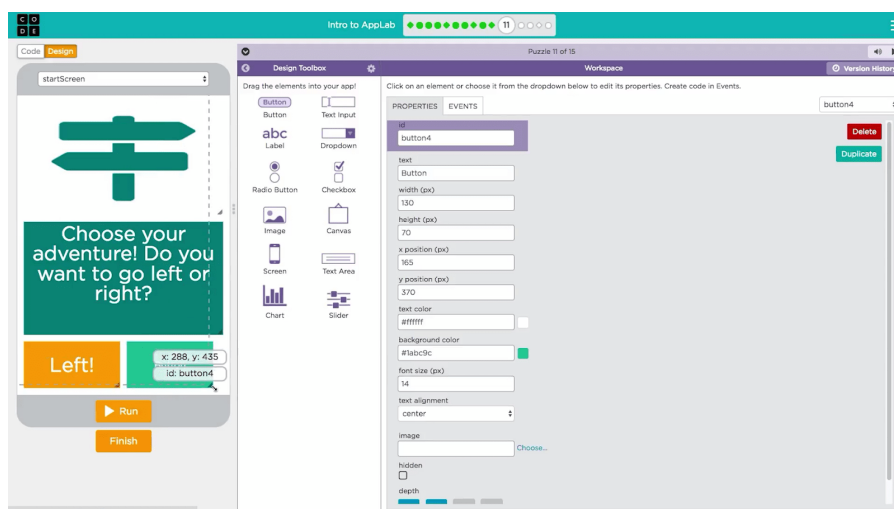
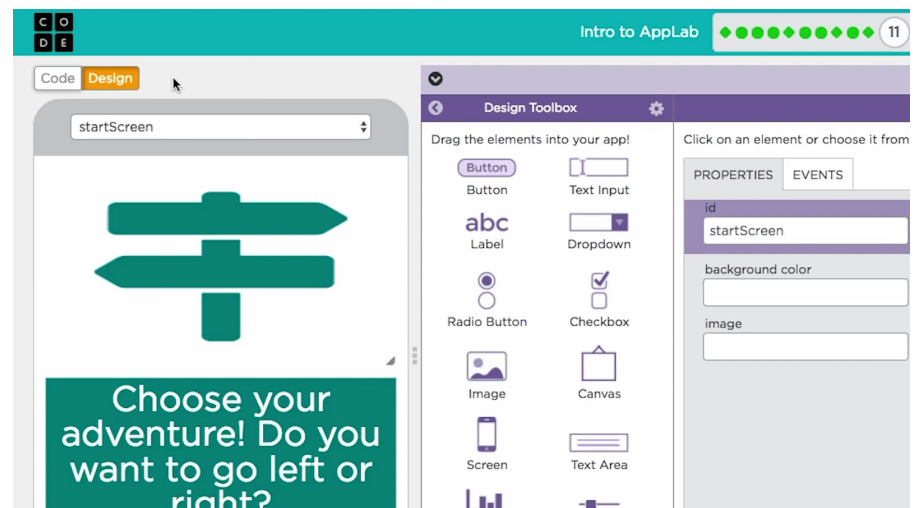
App Lab

Design Mode

To build your own apps you're going to need to start designing screens and elements from scratch. App Lab makes this easy to do with Design Mode.



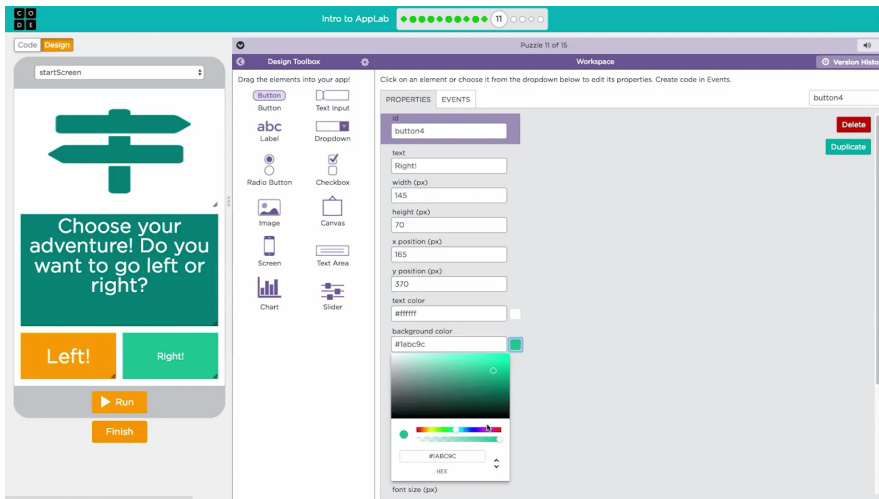
Use the switch on top of your app to go into Design Mode.



You can add new elements by dragging them onto the screen. You can move them around to different locations and change their size by dragging the bottom right corner.

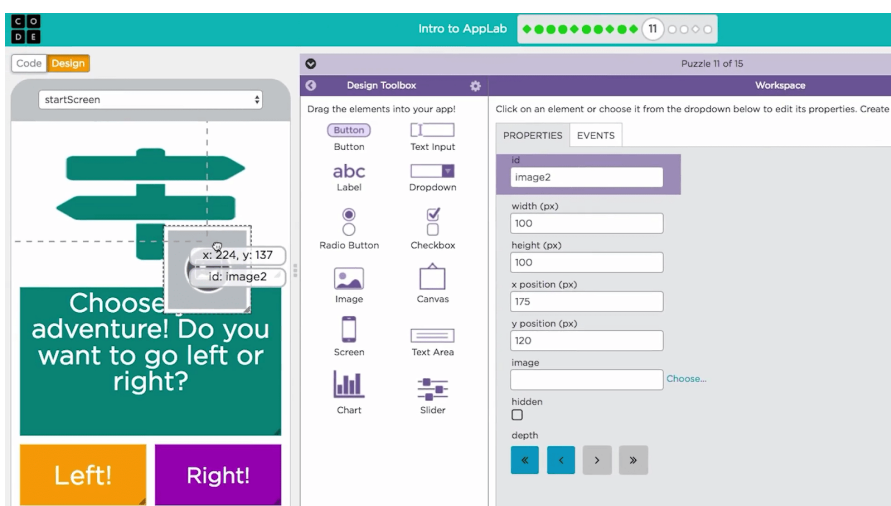
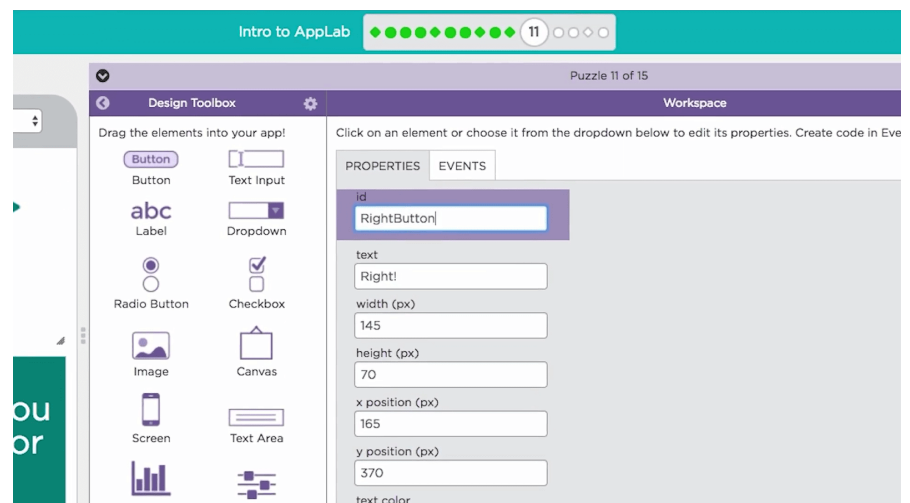
App Lab

Design Mode



To change an element's properties use the controls on the right. For example, it's really easy to change this button's text, colour, and font size.

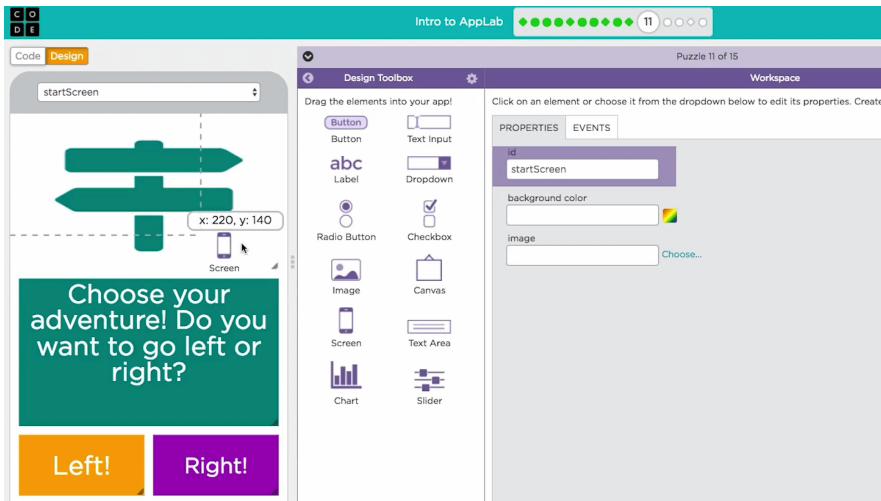
When you add a new element to your screen it'll get a generic ID like button1. It's a good idea to change this button's ID to something more meaningful like "rightButton". So that you'll know which one it is when you go to the program.



If you add an element to your app by mistake, just drag it out or hit the delete.

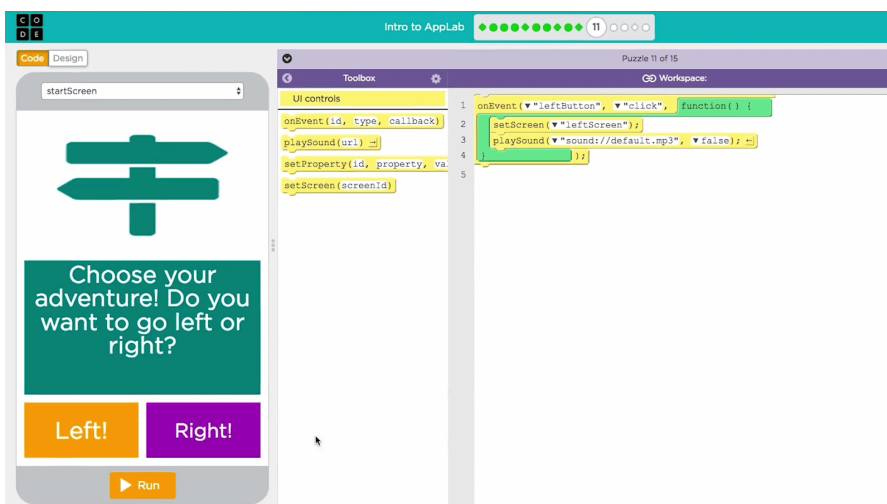
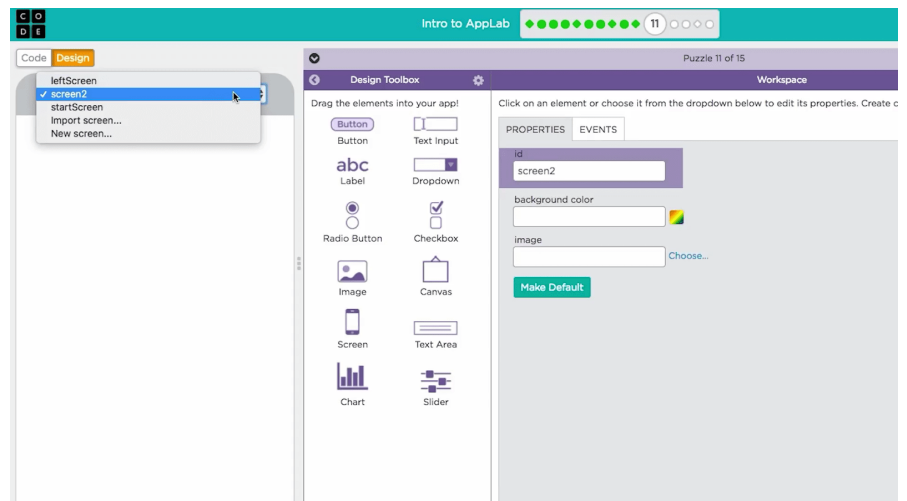
App Lab

Design Mode



You can add entirely new screens to your app by dragging in a screen element

From the drop-down at the top you can quickly switch back and forth between the screens you create.



Inside your app, you'll need a way to switch between all of these screens, so the setScreen block has been added to the toolbox. Use "setScreen" inside the "onEvent" block to change screens at the click of a button.

Design Mode

Goal: Add a second button to the screen that says "Right!" with an `"id"` of `"rightButton"`.

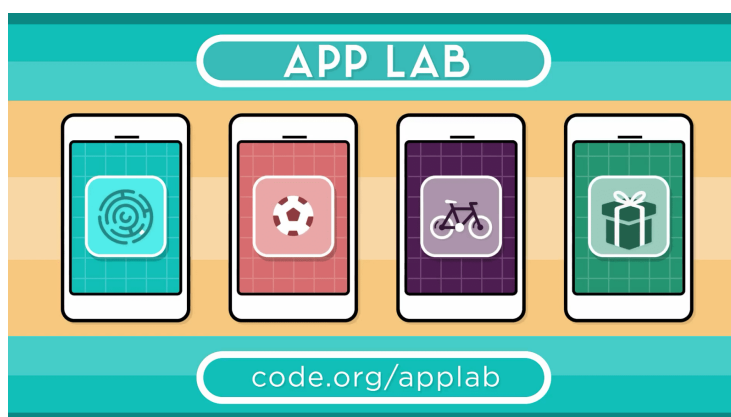
How: In Design Mode drag in a new button. Position and resize it. Then set its `"id"`, `"text"`, `"background color"`, and `"font size"`.

Goal: Create a new screen called `"rightScreen"`. Add a text label that says "You went right!".

How: In Design Mode drag in a new screen element. Change its `"id"` to `"rightScreen"`. Then drag a label onto the screen and change the text to "You went right!".

Goal: The screen switches to `"rightScreen"` when the user clicks the `"rightButton"`.

How: In Code Mode add a new `onEvent()` to your program for your "Right!" button. Add a `setScreen()` block inside of it that switches screens to `"rightScreen"`.



Homework 1 Due Date:

Homework 1 **Due:** _____

Question: How many mobile applications can you find that have made a huge success?

You are to fill out the table below using the internet to find the relevant information.

	Name of application	Highest number of users	Year of release in UK
X	Snapchat	101 Million Users	2011
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			



Homework 2 Due Date:

Homework 2 **Due:** _____

Task: Develop an app of your own. You will need to design it first before actually creating it in AppLab. Complete the following:

1. Write a short description of your app.
2. Break down the problem, like you did in lesson 1.
3. Draw a design to plan how your app will look.
4. Create it in AppLab.

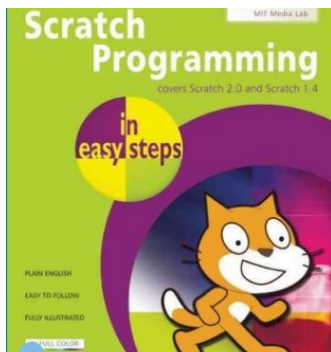
Take screenshots of your app and when you are done, publish it and paste the link in your work before uploading to the Teams assignment.



Wider Reading List

Wider Reading

- Scratch Programming in Easy Steps Paperback edition by Sean McManus
- The Everything Kids' Scratch Coding Book! By Jason Rukman
- Coding for kids by Zafer Demirkol



- ScratchX.org
- Scratch Tutorials Home Page - Scratch Game Video Tutorials (learnlearn.uk/Scratch)