# Teacher’s Notes

The students will be working online from the LiveCode Learning Management System (LMS). In the LMS, the hour of code will be split into steps that the students can follow. Each step will contain a short video and a set of slides.

It might be advisable to run through the slides and videos in advance of the Hour of Code session.

We recommend the students work directly from the LMS - they will be asked to download resources from the Resources tab in the LMS. At any point that the students are asked to download resources, they need to do this from within the LMS. This button can be found to the right above the slide display.

We suggest that the students follow the steps in the slides and use the videos as an additional resource. If pair-programming, the students should swap over at the end of each step. The slides will remind students to switch over if pair-programming at the appropriate times.

The videos have been filmed on a Mac, and the images that accompany the slides will be from a Windows machine - this is so that both Mac and Windows users can get visual clues.

Within the hour, we expect the students to be able to complete at least one of the apps. If the students struggle for time, or wish to complete the additional apps and challenges outside of the hour allotted, then the LMS will be available to them for the following week.

As this is aimed at 14 - 17 year olds, we expect a certain level of competency when using computers. However, we have provided additional notes below in reference to places where the students may get stuck.

# Downloading LiveCode

Before working in these lessons the students will need to download and install LiveCode.

There are links to the installers in Slide 4 of the Introduction step, or the installers can be downloaded directly [here](“http://downloads.livecode.com/livecode/“).

We are using the latest Stable version of LiveCode 7.1.0.

# How LiveCode Works

The LiveCode platform allows you to author apps for all platforms from either a Windows, Mac or Linux machine. (The only exception is deployment to iOS, you can create for iOS on any platform but due to third party licensing restrictions you can only build iOS on a Mac at this time and you will need a Apple developer program membership to submit to the store.)

At the start of the app creation process is a drag-drop user interface builder. If you’ve ever used a paint, drawing or presentation program you should find this process familiar. You can simply drag out the user interface elements you want to use. You’ll see your app take shape in front of your eyes. Objects can be customized to each platform allowing you to create native-themed apps. (With LiveCode 8 this process is automatic.)

Once you’ve created your user experience its time to add some code to tell all the objects what to do. To make it as simple as possible, LiveCode sends something called an event to every object whenever a user does something with that object. Think of it like a postman delivering letters to the objects in your app whenever the user interacts with them. For example, when a user taps a button a touchStart message is sent to that button. If you want the button to do something, you add code to handle (or respond to) this event. You can choose which events you want your program to respond to.

You’ll find that the LiveCode language is based not on complicated programing jargon but rather on English. You can express what it is you want your app to do using straightforward commands. There is still some learning to do – you can think of it more like learning a new language, a simplified and logical form of English. You won't need very much by way of maths (unless you're writing a program that involves complex processing algorithms). And as you're learning you'll discover that you can describe many actions in just the same way you think about them.

## Event Driven Programming

A LiveCode application is driven by user actions. LiveCode constantly watches the computer for common actions, such as clicking on a button, typing into a field, sending data across a network, or quitting an application.

Whenever an event occurs, LiveCode sends a message. When writing your program, you decide what messages you want your program to respond to. LiveCode automatically sends each message to the most relevant object. For example, if a user clicks on a button, LiveCode sends a message to the button. You place code within the button that tells it how to respond to being clicked on.

There are a wide range of possible events. When a user clicks on a button, a series of events are sent to the button. For example, when the mouse first moves within the border of the button a mouseEnter message is sent. Then a series of mouseMove messages are sent as the mouse moves over the button. When the mouse button is depressed a mouseDown message is sent. When the mouse is released a mouseUp message is sent. You don't have to respond to all of these events. You simply place code within an object to handle the events you want your application to respond to.

Events that are not handled by individual objects can be handled in a number of ways at different levels of your application, in libraries, or they can be ignored.

## Object-Based Programming

Any graphical application you build using LiveCode will be based on objects. With LiveCode you typically create the objects of your application before writing any code. You can start by drawing the buttons, text fields, and other controls that make up your application. LiveCode operates like other layout, drawing or application development environment. You can select controls by clicking them, move them by dragging them around, resize them, and change their 'layer' to move them closer or further from the 'top' of the interface.

Once you have the objects in place, you can proceed to attach code to each object to respond to the events you want. LiveCode includes a complete graphical development environment that makes it easy to create and edit any kind of user interface. It includes objects for all the basic operating system elements, including buttons, checkboxes, text fields, menus, graphics, and many more. In addition you can create and customize your own objects that look and behave however you want.

## The ‘Edit’ and ‘Run’ Mode

Unlike most other development systems, a LiveCode application can be created, edited, debugged and run live.

When in run mode, objects receive all the normal messages that drive a LiveCode application. For example, clicking on a button in run mode will cause a mouseUp message to be sent to it, causing the button’s script to run if you’ve designed it to respond to the mouseUp message.

When in edit mode, objects do not receive messages when you click on them, and you can move, resize or edit the properties for objects.

There few other differences between the two tool modes. You can view and edit properties and code in either mode. Your application does not stop running while you make changes to it. Only mouse interaction with objects is suspended in edit mode to allow you to edit them more easily.

Because LiveCode is constantly live, you can easily make simple changes and watch each change take effect as you make it. This allows you to design and experiment using an iterative process, resulting in a more productive and satisfying development experience.

# General Terms used in LiveCode

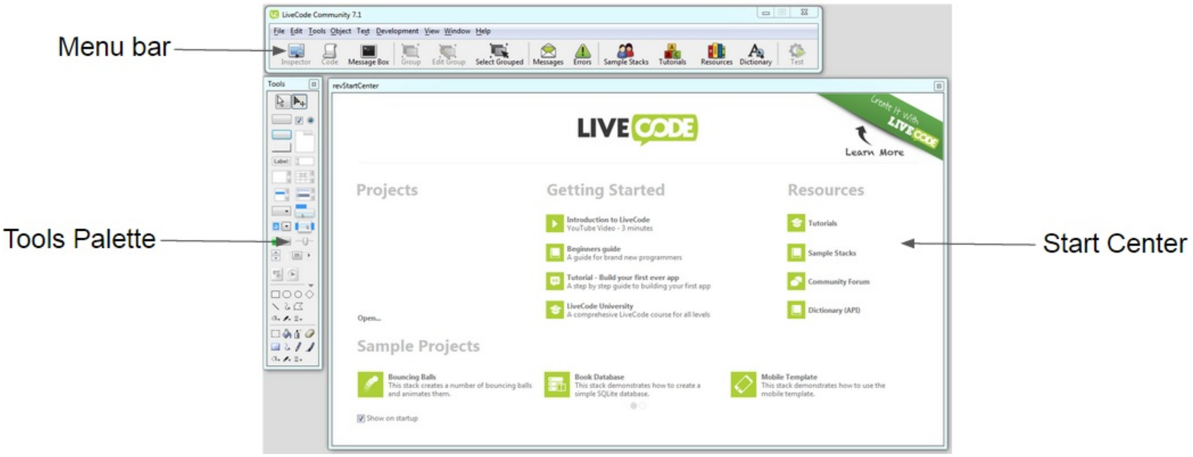
- Stack: The first step in creating a LiveCode application is creating a window, which in LiveCode is called a stack. Each window you see in LiveCode is a stack.
- Card: Each stack contains one or more sets of information called cards. Each card can have a different appearance or all the cards in a stack can look the same. By going from card to card in a stack, you change what’s being displayed in that stack’s window. You can think of a LiveCode stack as a stack of playing cards (hence the name), where you can flip through the cards, but only one card at a time is visible. A stack can have a single card or many cards.
- Object: Every control in LiveCode is an object, this can be a button, field, scrollbar,

graphic etc. Objects are created by dragging them from the Tools Pal

- Edit mode: When in edit mode, objects do not respond to clicks so you can move, resize or edit the properties for objects.
- Run mode: When in run mode, you can interact with the app as a user. For example, clicking on a button in run mode will cause a mouseUp message to be sent to it and the script will run.
- Property: Every object has a set of associated properties. Properties control how an object looks and some aspects of an object's behavior. Some examples are name, visible, disabled, text color etc. Changing the properties of an object changes how it looks and behaves.
- Message: When a user event occurs LiveCode sends a message to the most relevant control telling it what kind of event occured. You add code to your app to respond to these messages.

# The LiveCode IDE

When you start up LiveCode you will see the Integrated Development Environment(IDE). The IDE is where we create the app.



## The Menubar

The menubar provides quick and easy access to the tools we will use the most.

## The Tools palette

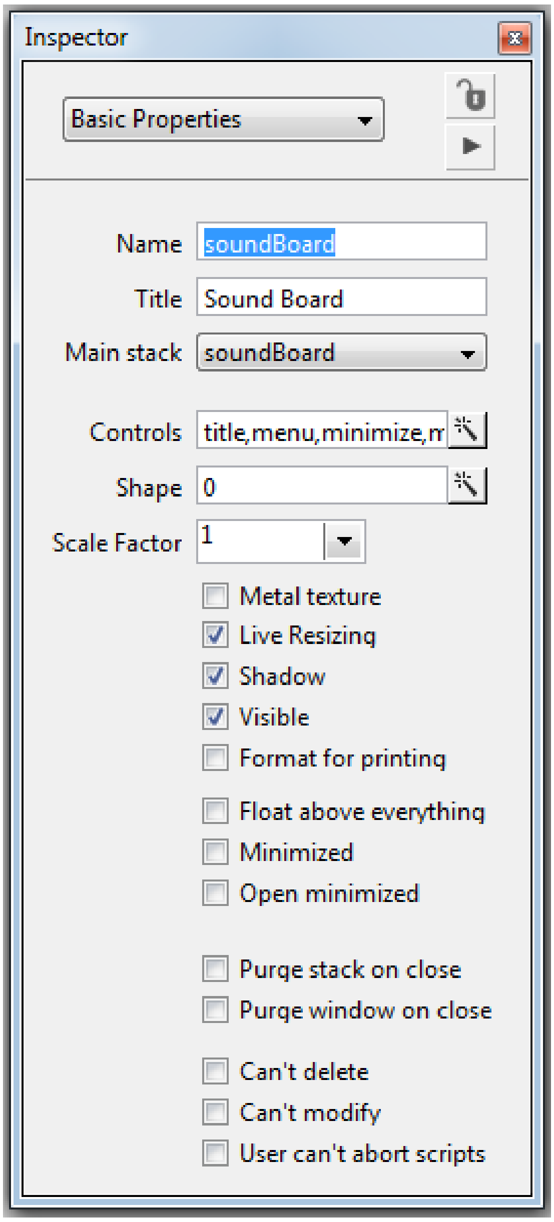
The tools palette lets us create the layout of an app. We drag out controls from here straight on to our app.

The Tools Palette can be reopened from the Tools menu if it gets closed, either intentionally or unintentionally.

## The Start Center

This window has links to your recent projects, the LiveCode forum, learning resources, example stacks, and more. We will not be using this, so we can close the window now.

## The Property Inspector



Inspector

Basic Properties

Name: soundBoard

Title: Sound Board

Main stack: soundBoard

Controls: title,menu,minimize,m

Shape: 0

Scale Factor: 1

□ Metal texture
☑ Live Resizing
☑ Shadow
☑ Visible
□ Format for printing

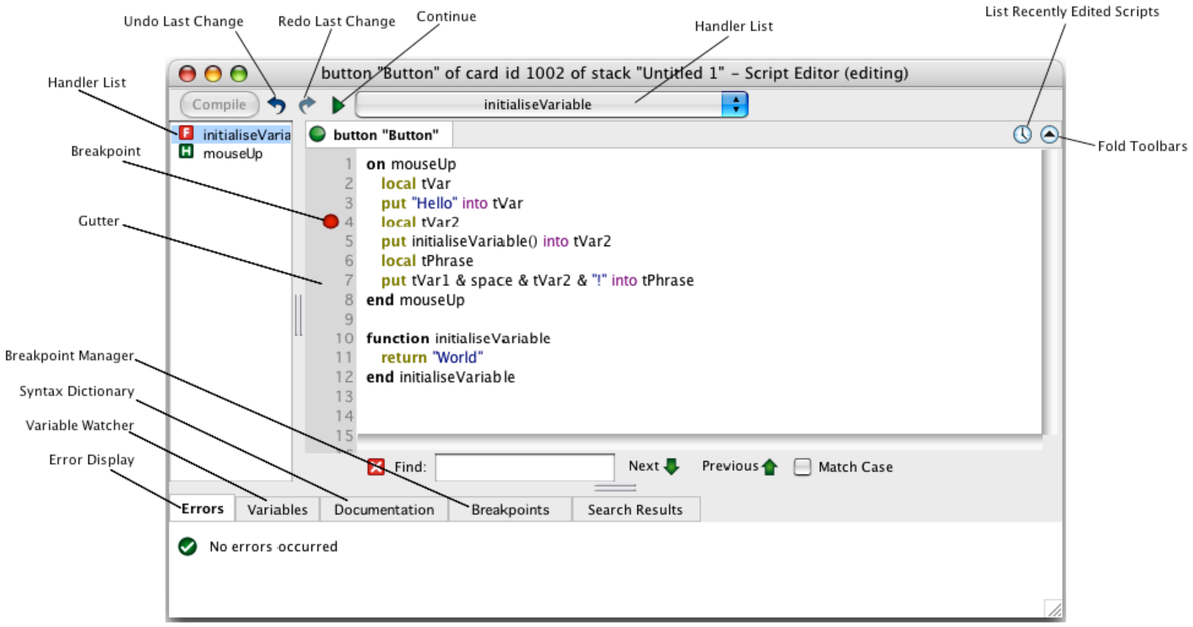
□ Float above everything
□ Minimized
□ Open minimized

□ Purge stack on close
□ Purge window on close

□ Can't delete
□ Can't modify
□ User can't abort scripts

The Properties Inspector allows you to view and edit the properties for any selected object. Properties control how an object looks and some aspects of an object's behavior. The Inspector can be accessed by double clicking on a selected object, from the toolbar, from the Object menu and from context sensitive menus.

## The Script Editor



```
on mouseUp
   local tVar
   put "Hello" into tVar
   local tVar2
   put initialiseVariable() into tVar2
   local tPhrase
   put tVar1 & space & tVar2 & "!" into tPhrase
end mouseUp

function initialiseVariable
   return "World"
end initialiseVariable
```

No errors occurred

The Script Editor within LiveCode has been designed specifically for LiveCode coding. It includes features to help make code more understandable. These include code indentation and color coded syntax highlighting, as well as other integrated tools such as a Debugger and syntax Dictionary. You can access the Code Editor for an object by selecting the object then choosing Script from the Tool bar. The Code Editor is also available from the Object menu, and from a number of other context sensitive menus detailed elsewhere in this guide. Each of the Code Editor's components are outlined below.

# Possible Causes of Confusion

## Jungle Soundboard

### Introduction Slides - Slide 4

Students working on a Mac should download the Mac installer.
Students working on a Windows machine should download the Windows installer.

Once downloaded, the installer will be found in the downloads folder. Double clicking on the installer will open it. The instructions will guide the students through the installation process. Once installed, it is a good idea to move the LiveCode application to the Applications folder on their machine.

## Introduction Slides - Slide 5

If there are any additional windows open when the student opens LiveCode that are not shown/mentioned on the slide, these can be ignored and closed as they will not be using them.

All of the steps that the students will be following in the slides are to be done in the LiveCode IDE (that is to say within the LiveCode application) so they will need to keep LiveCode open.

## Introduction Slides - Slide 6

The students do not need to have the Start Center open, they will not be using it and it takes up space on the screen so it is strongly recommended that they close it. Once closed, they can open it again by going to the "Help" menu and selecting "Start Center" from there if they wish to open it again.

## Introduction Slides - Slide 7

To switch over to Edit mode, click on the edit/pointer tool in the tools palette.
To switch over to Run mode, click on the run/browse tool in the tools palette.

## Creating An App - Slide 2

To save a stack, students can either use the keyboard shortcuts (cmd & s on Mac and ctrl & s on Windows) or select "Save" or "Save As" from the "File" menu.

## Creating An App - Slide 4

To change the name and title of the stack, the students need to click in the text box to the right of the property names and type the new name/title into the text box.

## Creating An App - Slide 6

To get to the "Size & Position" pane, students need to open the drop down menu (as instructed in the previous slide) and select "Size & Position" from the drop down menu. This will take them to the right pane.

## Creating An App - Slide 8

If the students are unsure of what card they are on, they can make sure that they are on the second card in the stack (the one they have just created) by opening the “View” menu and selecting “Go Last”.

## Creating An App - Slide 9

To download the resources, the students simply need to click on the “Resources” button on the right of the bar above the slide viewer. This will open the resources pane on the right hand side of the LMS. Then simply click on the button named “Resources”. The download will automatically start and the students will find the folder in the downloads folder on their machine.

To change the pane on the right hand side of the LMS to display the topics rather than the resources, simply click on the button with the menu icon on the left of the top bar in the right hand pane.

## Creating An App - Slide 10

The folder that the students just downloaded contains two further folders - one named images and one named sounds. They need to make sure they select the folder “images” so that all of the images in that folder are added to the stack.

## Creating An App - Slide 11

See slide 5 if they have forgotten how to change to a new pane in the property inspector.

A further check to make sure they are on the correct card: when the students open the Card Inspector they can check the value of the name property. At this step, the students should have the “jungleSounds” card open, so when they open the card inspector the name property should be “jungleSounds”. If they are still on card “resources” then they can navigate to the the correct card by opening the “View” menu and selecting “Go First”.

When the students click on the background pattern button (the left hand box out of the two boxes to the right of the word “Background” - hovering over the box will bring up a tooltip that will say “backgroundPattern”) a new window will open. At the top of the new window there is a drop down menu - clicking on the blue arrow button will open the drop down menu. From here, they need to select “This Stack”.

## Adding Buttons - Slide 2

If they have forgotten how to switch to edit mode - need to select the top right button in the tools palette: arrow with crosshair (see slide 5 in the Introduction).

They will know that the button is selected as its border will be outlined by grey boxes.

## Adding Buttons - Slide 4

To copy and paste from the “Edit” menu: Open the “Edit” menu, select “Copy”, then open the “Edit” menu again and select “paste”.

Keyboard shortcuts → Mac: cmd & c for copy and cmd & v for past. Windows: ctrl & c for copy and ctrl & v for paste.

## Adding Buttons - Slide 7

See notes Creating an App - Slide 11 above for further information if the students are stuck on step 5.

## Starting to Add Code - Slide 1

When the students downloaded the Resources folder they were asked to move it to the folder that their stack is saved in. If they cannot find the Resources folder, they should check their downloads folder. If it is still in the downloads folder, they should move it to the folder their stack is saved in now. The Resources folder should contain two further folders: “images” and “sounds”.

## Starting to Add Code - Slide 2

When the students hit “Apply”, if the code has been typed in correctly, then in the bottom of the script editor window a white tick in a green circle with a message “No errors occurred” will appear. If there appears a white cross in a red circle, then the code has not been entered correctly. Check for typos. The error message that also appears also offers information on what the error is.

## Starting to Add Code - Slide 4

Close the stack by closing the window. Once it is closed, open the stack again by going to the file menu and selecting open, then choose the stack file.

## Starting to Add Code - Slide 6

To switch to edit mode, click on the arrow with crosshair in the top right of the tools palette.

When the students open the script editor for the “tiger” button there will be two lines of code there already: on mouseUp and end mouseUp, the students do not to repeat these two lines of code.

### Starting to Add Code - Slide 8

Troubleshooting if the tiger sound doesn’t play (these steps can also be taken later if any of the other sounds don’t play): Try step 1 then press the button, if the sound still doesn’t play then move on to step 2 etc.

first check the sound files can be played on the machine the student is working on - this shouldn’t be a problem as the sounds files are mp3 files

check for any typos in the script for the “tiger” button (select the tiger button in edit mode and then select object script from the object menu) make sure the code in the script editor exactly matches the code on the slide - there should be no additional lines of code.

check that the file path is correct - In the folder where the stack has been saved there should be the resources folder that the students downloaded. In this folder there should be a sounds folder, the tiger sound should be in this folder. If the sound files are not in the correct place then the file path “Resources/sounds/tiger.mp3” given in the code on slide 6 will be incorrect

check that the default folder has been set - click on “Message Box” in the toolbar; type “put the defaultFolder” and press return; the name of the defaultFolder should be shown below the line you typed; this should be the name of the folder that the stack is saved in

check that there are no typos in the stack script (select stack script from the object menu)

open the Message Box as in step 3 and type in “preOpenStack” then press return, repeat step 3 to check the default folder has been set correctly

# Piano Soundboard

### Creating the Piano App - Slide 4

Don’t worry too much about the size, this app can be any size and can be resized later.

## Creating the Piano App - Slide 6 and 7

It is important that the sound files are in the right place.

The unzippe “Resources” folder should be in the same folder as the stack file, the “Resources” folder should contain a folder called “sounds”, which contains 13 mp3 files.

## Adding buttons - Slide 2

Make sure you are in **Edit mode** to add the button.

## Adding buttons - Slide 3

Remember that the button has to be selected before opening the Object Inspector.

## Adding buttons - Slide 11

The 0,0 point in LiveCode is in the topLeft corner of the card.

Every object has 4 points associated that can be set in the Property Inspector. The **top,left,right and bottom**. Setting any of these properties will move an object, not change its size. So setting the **top** and **left** to 0,0 will put the button in the topLeft corner of the stack.

## Adding buttons - Slide 12

To resize and move an object ensure you are in **Edit mode** and that the object is selected.

## Adding buttons - Slide 13

Copy and pasting can be done in **Edit mode** using the Edit menu or the standard keyboard shortcuts. The selected object will be copied.

The copy will be placed directly on top of the original control so don’t worry if you think the copy hasn’t been made, just move it slighly using the arrow keys and you should be able to see that you now have 2 buttons.

## Adding code - Slide 3

Double check that you have opened the Stack Script.

Remember to click the Apply button in the Script Editor and check that the indicator goes green.

### Adding code - Slide 4

Ensure you are in **Edit mode** so that you get the context menu when you right click on the stack.

### Adding code - Slide 9

The **Group** button is found in the Menubar, it will group together all the selected controls.

It is important to check that LiveCode is not in **Select Grouped** mode, ensure the **Select Grouped** icon in the menubar is grey, meaning off, and not green, meaning on.

### Adding code - Slide 10

The mouseDown message is sent when the mouse button is pressed down on an object.

### Adding code - Slide 13

Variables are containers that hold information the app might need later.

LiveCode variables don't need to be defined first, they are created when you use them for the first time, by putting a value into the variable.

LiveCode variables don't have a type.

# Further Challenges

## Slide 3

If the students want to create their own soundboard apps, then they will need to think about sourcing their own images and sounds. Images can be taken from google images and royalty free sounds for both public and private use can be found on websites such as soundbible.com, freesound.com and audiojungle.net.

They are given no further guidance, students should be able to refer to how they made the piano and jungle soundboards for additional help.

## Slide 10

Tip 1: To find the location of the monkey/elephant button
select the button (make sure they are in edit mode in the tools palette to do this)
open the “Object” menu and select the “Object Inspector”
In the Property Inspector, navigate to the “Size & Position” pane (open the drop down menu by clicking on the arrows to the right of where it says “Basic Properties”)
The location of the monkey button is 300,200
The location of the elephant button is 481,200

Tip 2: Open the sound files and play them - the duration will be shown in the media player. The time for both the monkey and elephant sounds is 2 seconds.

## Slide 13

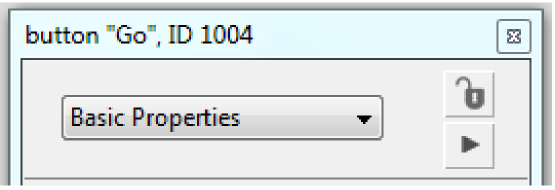
If the app is not working correctly, the best thing for the students to do is to double check the extra lines of code they added to each button do not contain any typos and the code is identical to that given in the slides.

# Things to look out for

These are a few of the things you might want to double check if your app is not behaving as expected

## You can’t find the property you want to set in the Property Inspector

The Property Inspector displays the name of the object you are inspecting in the title bar so you can always check with set of properties you are looking at.

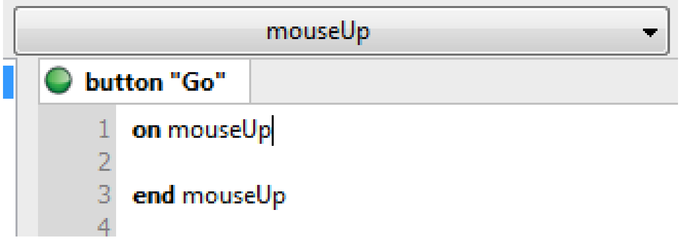


In particular make sure you are not looking at the card or stack properties.

## You have changed your code and it hasn't made any difference

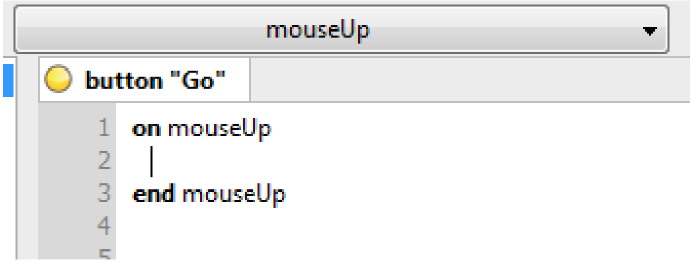
Always make sure to apply your scripts. There is an indicator next to the "Apply" button in the Script Editor, you want to make sure it is green.

- Green: applied with no Script Errors



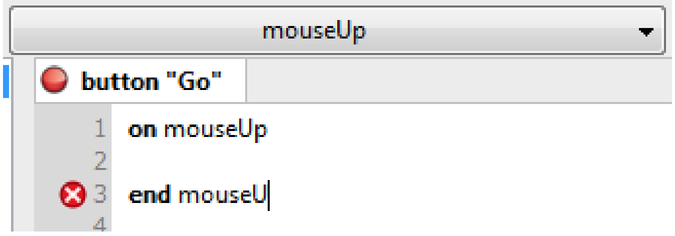
```
on mouseUp

end mouseUp
```

- Yellow: there are unapplied changes to the script



```
on mouseUp

end mouseUp
```

-
- Red: the script has errors



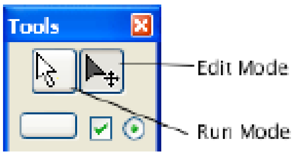
```
on mouseUp

end mouseU
```

-

## I can't test my app/make changes to my app

Check what mode you are in in the Tools Palette

-

- 

- Edit mode: When in edit mode, objects do not respond to clicks so you can move, resize or edit the properties for objects.
- Run mode: When in run mode, you can interact with the app as a user. For example, clicking on a button in run mode will cause a mouseUp message to be sent to it and the script will run.

# Help and Resources

There is additional help and resources available

- The LiveCode User Guide: available from the Help meu within LiveCode
- The LiveCode Dictionary: all the LiveCode keywords with examples and explanations. Available by clicking the **Dictionary** button in the Menubar
- The LiveCode Lessons Portal: Step by Step Tutorials in LiveCode