

Today you will learn to program using MIT's Scratch platform. We will walk them through creating their own applications. The core concepts that will be taught are: events, operators & math, and conditional logic. To teach these concepts, you will write your own application.

Getting Started - Installing Scratch

If you have not already installed Scratch, we have provided USB drives with the software. The USB drives include **Scratch 2** and **Adobe Air** for Windows, Mac OS X, and Linux.

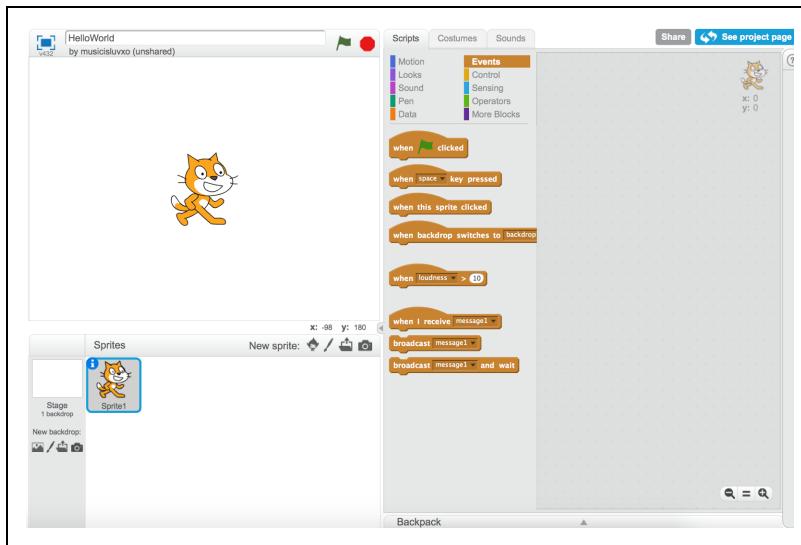
Unfortunately, Adobe has stopped support for Air on Linux. They offer documentation on installing for Linux on their website (<https://helpx.adobe.com/air/kb/install-32-bit-air-linux.html> & <https://helpx.adobe.com/air/kb/install-air-2-64-bit.html>).

- *Windows users:* Install “*AdobeAIRInstaller.exe*”. Upon completion, install *Scratch-432.exe*
- *OS X users:* Mount and Install “*AdobeAIR.dmg*”. Upon completion, mount and install *Scratch-432.dmg*

What you will know how to do after this class:

- Learn to program using **MIT's Scratch**
- **Events:** you will write a hello world application.
- **Logic:** you will write a decision making game.
- **Operators and Math:** you will write a Point of Sale system.
- Finally, will close the class with resources and where to go next.

ASSIGNMENT 1



The screenshot shows the Scratch 2.0 interface. A cat sprite is on the stage. The scripts editor contains the following script:

```
when green flag clicked
when space key pressed
when this sprite clicked
when backdrop switches to backdrop1
when touchness > 10
when I receive [message1]
broadcast [message1 v1]
broadcast [message1 v2] and wait
```

The stage properties show 1 backdrop. The script editor tabs are Scripts, Costumes, and Sounds. The Scripts tab is selected.

Welcome to Scratch!

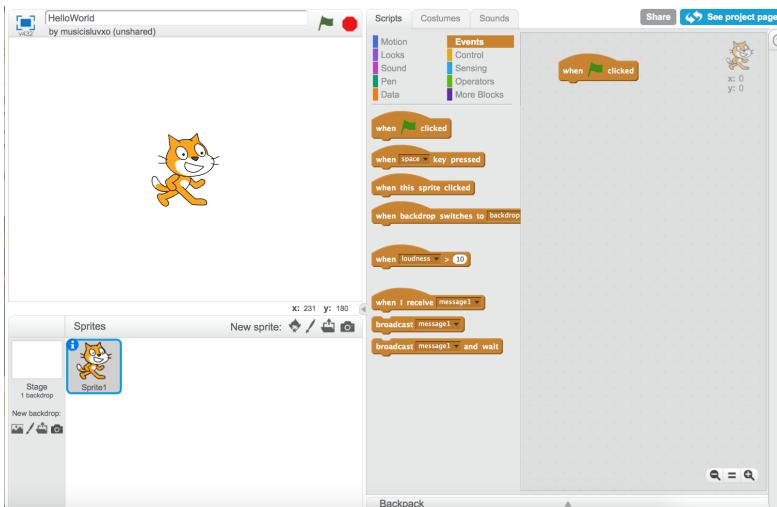
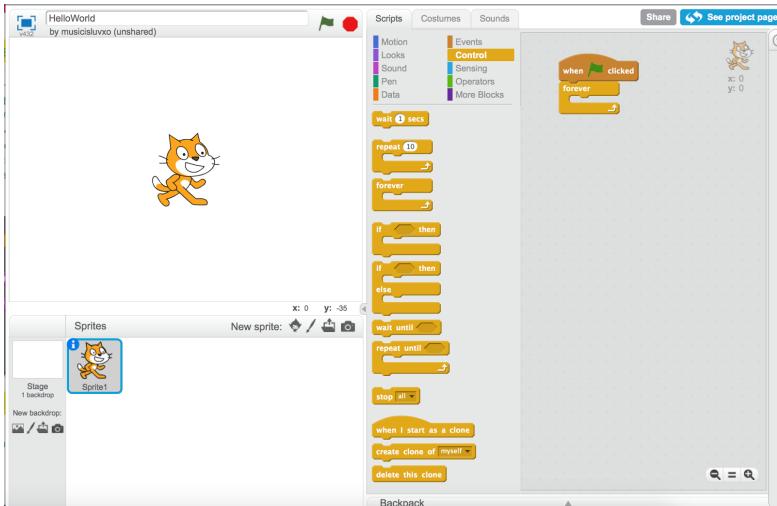
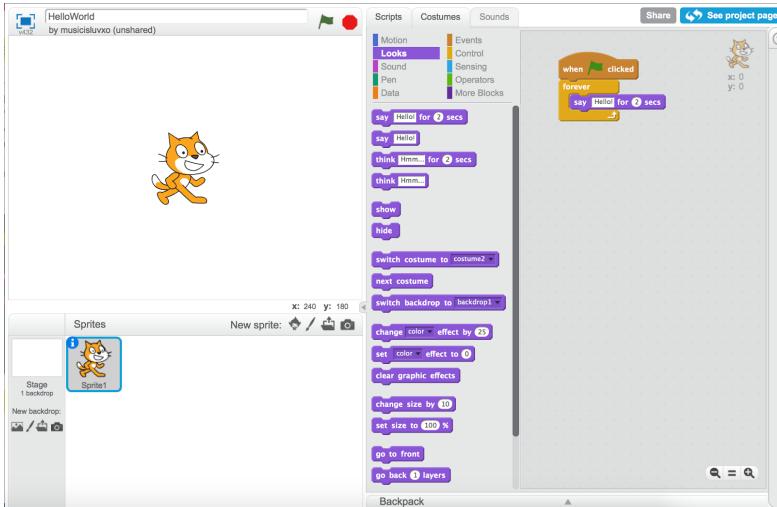
Sprites:

Events:

Intro to Programming

Pie Coding Collaborative (<https://pie-coding.github.io>)

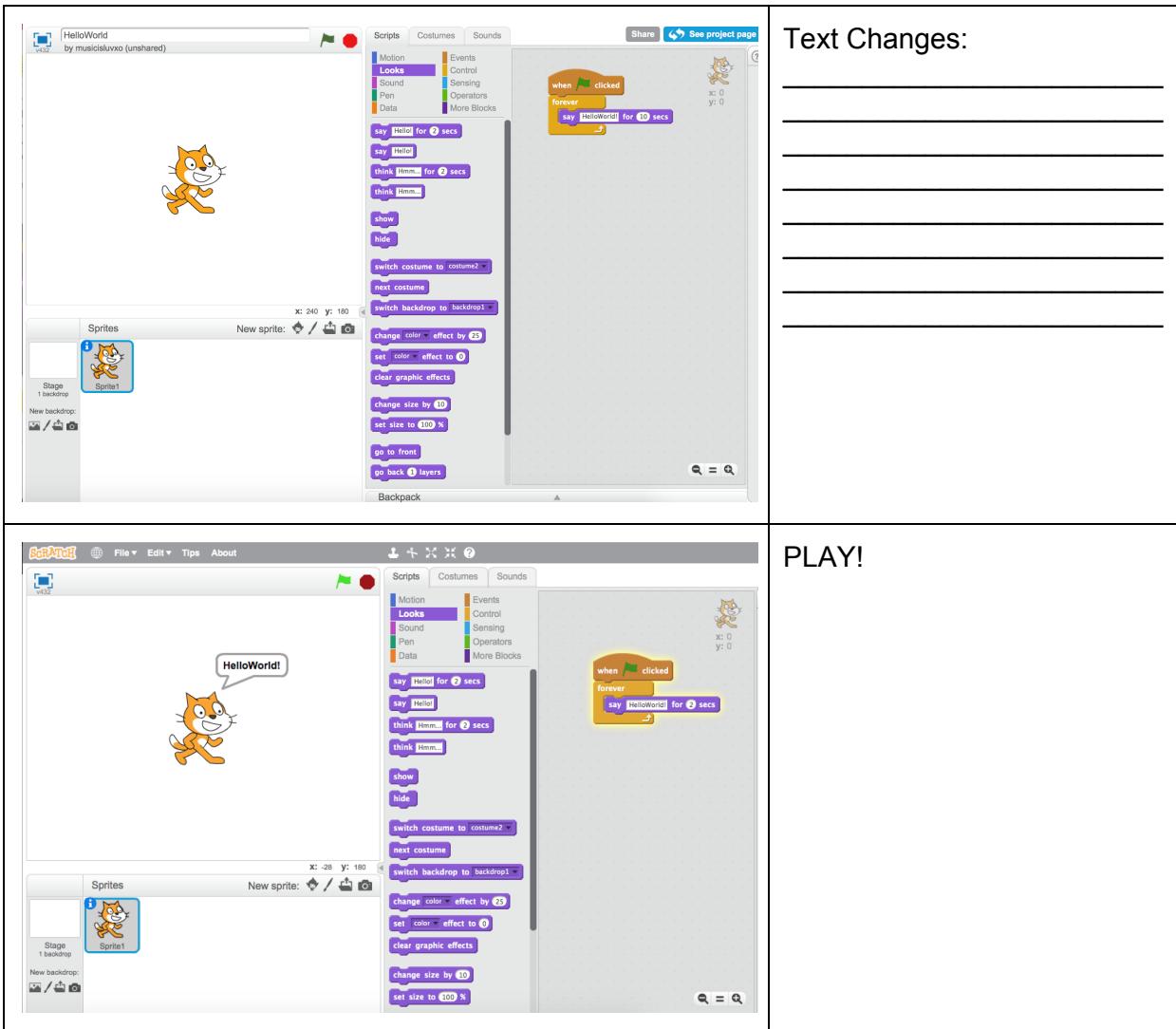
February 25th, 2015

| | |
|---|--|
|  <p>The Scratch stage shows a single orange cat sprite. The Scripts tab displays a script starting with "when green flag clicked". It includes several other scripts like "when space key pressed", "when this sprite clicked", and "when backdrop switches to [backdrop v]". The Backpack contains scripts such as "when I receive message []", "broadcast message1 v", and "broadcast message1 - and wait".</p> | <p>Start Flag:</p> <hr/> <hr/> <hr/> |
|  <p>The Scratch stage shows the same orange cat sprite. The Scripts tab now features a "when green flag clicked" script with a "forever" loop attached. Inside the loop are blocks like "repeat (10)", "forever", "if <random> then", "if <random> then", "else", "wait until", "repeat until", and "stop all". The Backpack contains scripts for cloning.</p> | <p>Stop Flag:</p> <hr/> <hr/> <hr/> |
|  <p>The Scratch stage shows the same orange cat sprite. The Scripts tab shows a "when green flag clicked" script with a "forever" loop. Inside the loop is a "say [Hello! v] for (1 sec)" block. The Backpack contains various look and sound blocks.</p> | <p>Time Flag:</p> <hr/> <hr/> <hr/> <p>Looks:</p> <hr/> <hr/> <hr/> <p>Say:</p> <hr/> <hr/> <hr/> |

Intro to Programming

Pie Coding Collaborative (<https://pie-coding.github.io>)

February 25th, 2015



ASSIGNMENT 2

Intro to Programming

Pie Coding Collaborative (<https://pie-coding.github.io>)

February 25th, 2015

The image shows the Scratch interface. On the left, the 'Sprite Library' window is open, displaying various categories like 'All', 'Animals', 'Fantasy', etc., with a grid of sprites including a person, airplane, acrobatic man, Anna, apple, arrow, ball, ballerina, balloon, banana, baseball, basketball, bat1, bat2, and a palette. On the right, the main stage shows a cat sprite and a bat sprite. The script editor shows a script for the bat sprite with blocks like 'say Hello for [2 secs]', 'think Hm... for [2 secs]', 'switch costume to [bat2-b]', and 'next costume'. The bottom Sprites panel lists 'Sprite1' and 'Bat2'. A second Scratch interface is shown below, with a different set of sprites including a girl, a dragon, and a Giga walk sprite. The script for the Giga walk sprite includes 'switch costume to [Giga walk4.]', 'next costume', and 'switch backdrop to [backdrop1]'. The bottom Sprites panel lists 'Sprite1', 'Bat2', 'Girl3', 'Dinosaur3', and 'Giga walk.'.

Now lets add 5 more sprites!

Pick a sprite!

Intro to Programming

Pie Coding Collaborative (<https://pie-coding.github.io>)

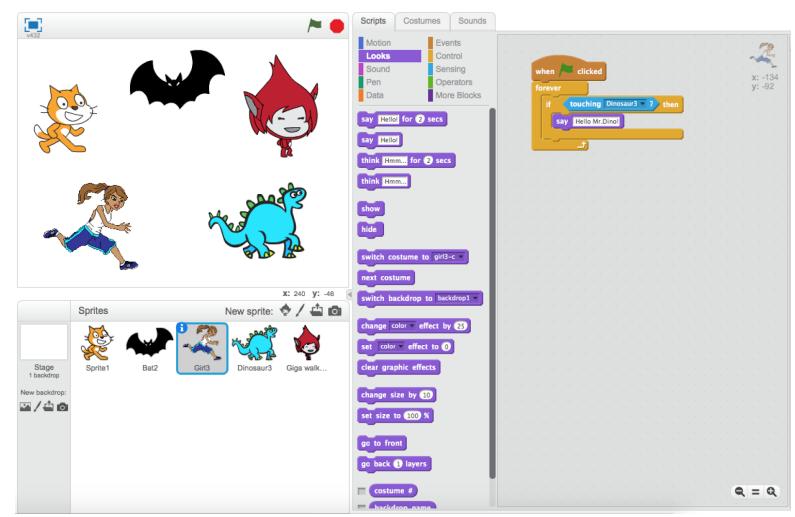
February 25th, 2015



The Scratch stage features five sprites: a cat, a bat, a girl, a dinosaur, and a giga-walkie. The script for the cat sprite is:

```
when green flag clicked
forever
  [wait (1) sec]
end
```

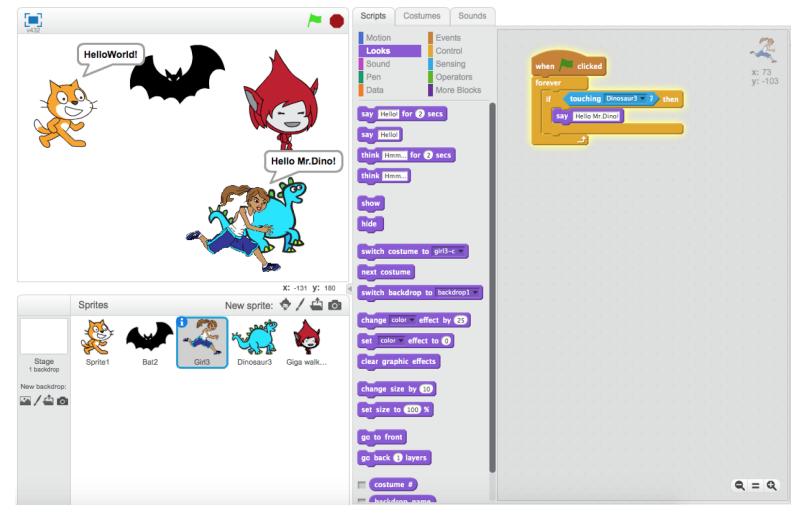
Take one sprite, and lets get going!



The Scratch stage features the same five sprites. The script for the cat sprite is:

```
when green flag clicked
forever
  if touching [Dinosaur3 v] then
    say [Hello Mr. Dino! v]
  end
end
```

Conditionals (if)



The Scratch stage now shows speech bubbles with "HelloWorld!" and "Hello Mr. Dino!". The cat's script remains the same as the previous stage.

ACTION!

Intro to Programming

Pie Coding Collaborative (<https://pie-coding.github.io>)

February 25th, 2015

The image shows a Scratch script editor with two Scratch stages. The left stage features a cat, a bat, a girl, and a dragon. The right stage features a dinosaur. Both stages have speech bubbles with text like "HelloWorld!", "Hello Gigi!", "Hello Mr.Dino!", and "Boop!". The scripts are as follows:

Stage 1 Script (Dinosaur):

```
when green flag clicked
forever
  if touching Giga walking? then
    say Hello Gigi!
  end
  if touching Bat2? then
    say Hello Batman!
  end
  if touching Girl3? then
    say Hello Running Girl!
  end
  if touching Sprite1? then
    say Hello Kity!
```

Stage 2 Script (Dinosaur):

```
when green flag clicked
forever
  say Boop!
  wait until touching Bat2?
  say Hello Batman!
```

Now, lets make things a little more complicated.

Give the dinosaur a little more to do!

if/else:

ASSIGNMENT 3

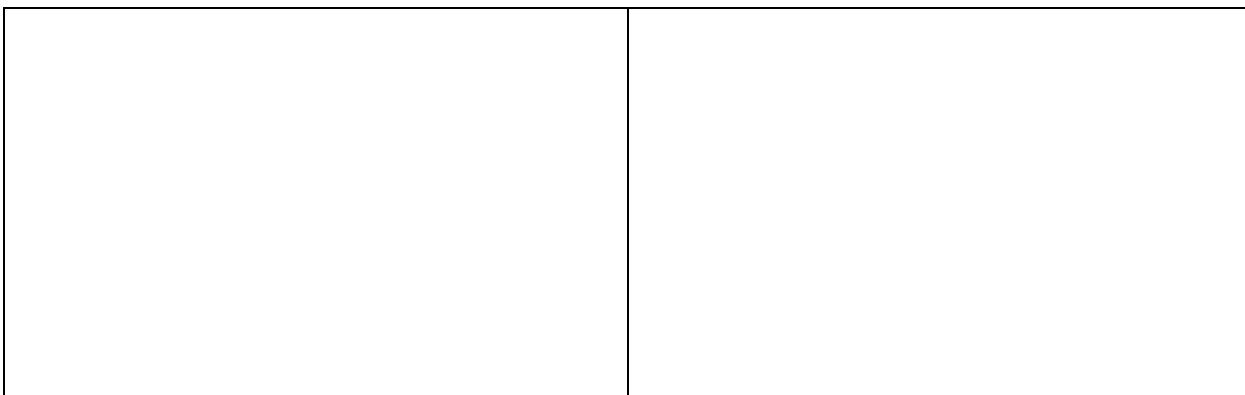
For your final assignment, you will create a point of sale system that could be used. We're going to create one for your favourite snack shop! Our primary goal is to teach you how to use various mathematical operators and to add some math to your application. However, you will also learn to identify user interface elements, how to prototype a user interface using those elements, and how to make it all work together.

First, we need to identify what we're selling. Our snack shop sells the following **fruits**: **apples** (\$1.90), **bananas** (\$2.15), **oranges** (\$1.00), and **watermelons** (\$3.25).

Finally, when the order is entered in, you must be able to **complete** it and ring it up or **cancel** it at any point in time and reset the point of sale system. When an order is completed, we must **add** 9.75% tax to it.

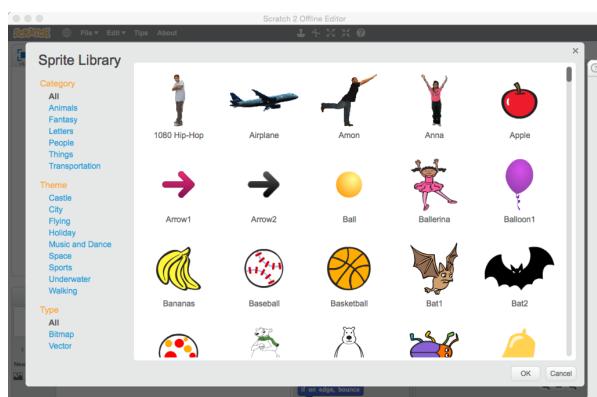
Activity:

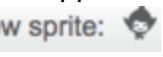
We will define our user interface elements.

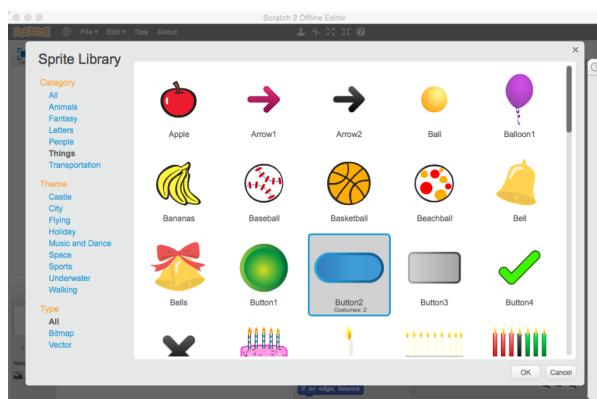


Activity:

Now, we will design the application in Scratch.



You will create a new application in Scratch. We're going to now add some custom sprites to our application. In the Scratch IDE, select **New sprite:**  from the IDE menu. You will now see the Sprites Library.

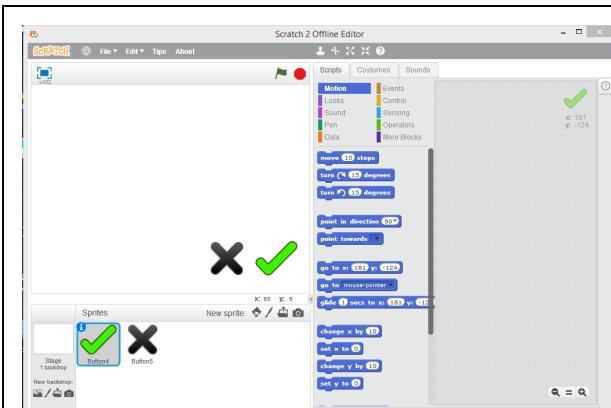


Select **Things** from the Library menu. We're going to add our two action elements: a **checkbox** and a **cancel “x”**. The checkbox will be used to complete orders and the cancel will be used to reset orders.

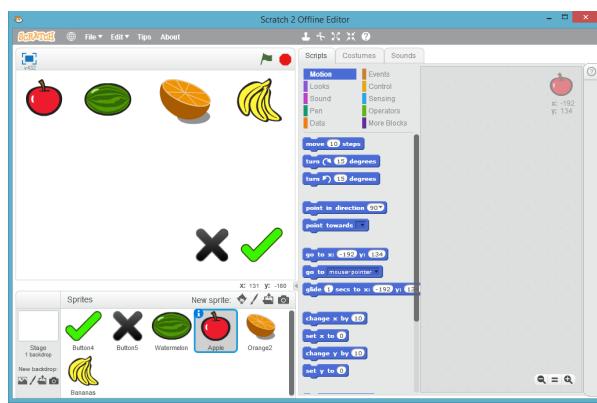
Intro to Programming

Pie Coding Collaborative (<https://pie-coding.github.io>)

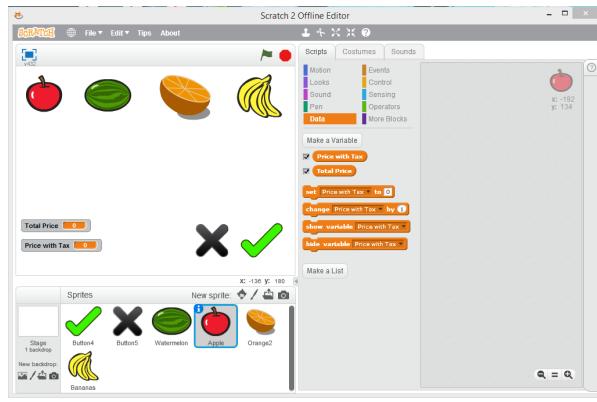
February 25th, 2015



Place these elements on your application next to each other.



Now, let us add our items for sale: apples, bananas, oranges, watermelons, muffins, and cake. *Tip:* You can find all of these items in the “Things” sprite library.

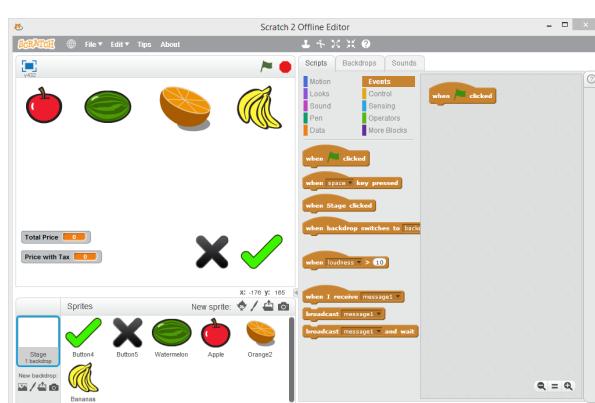


Now, we need a global **variable**. These are data types that we can use and modify through programming. We'll use the **Data** menu to create these. We will create two: “Total Price” and “Price with Tax”.

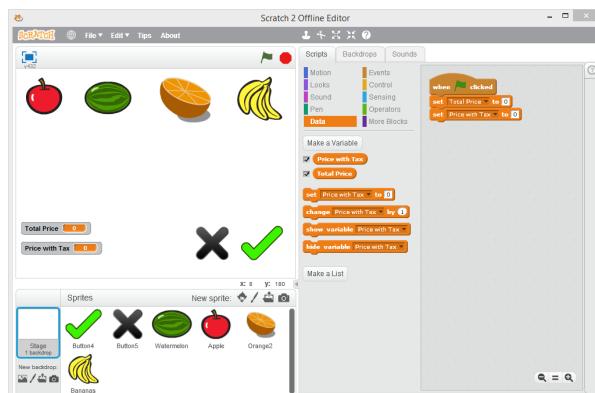
Intro to Programming

Pie Coding Collaborative (<https://pie-coding.github.io>)

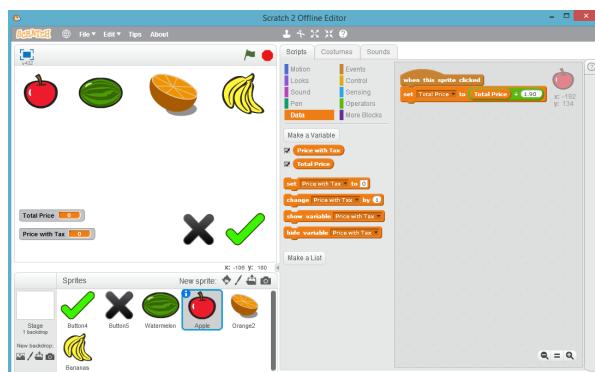
February 25th, 2015



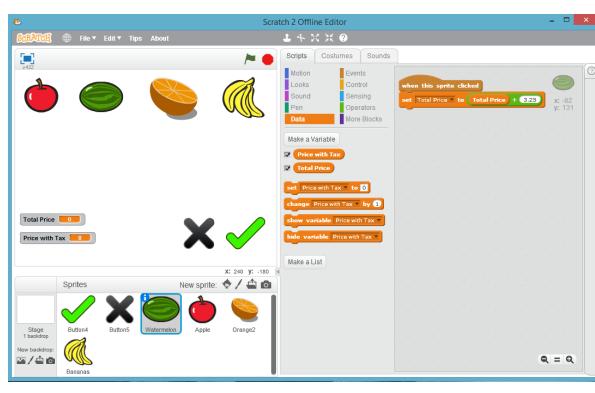
To start letting out application run, we need to create a new event tied to the **green flag** (run) button. Select the “**Stage**” and then open “**Events**” to create our starting event.



When it is clicked, we want to set our **variables** to 0.



Now, we need to modify the **total price** whenever an item for sale is clicked. Select the apple. When this sprite is clicked, we need to add the cost of the apple to the total price. We will then use the **operators** menu to add the cost to the total price inside of a set statement. Remember that Apples cost **\$1.90**.

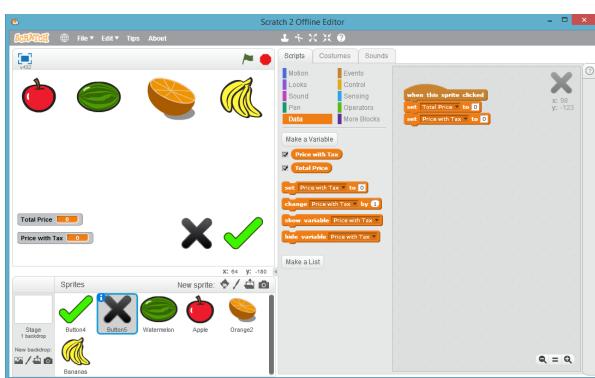


You will need to repeat this step for each of the other three items. Prices are as follows: **bananas** (\$2.15), **oranges** (\$1.00), and **watermelons** (\$3.25).

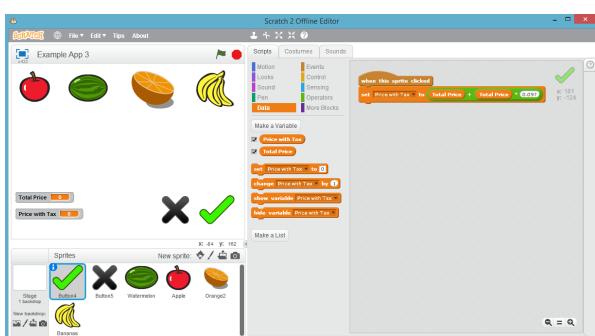
Intro to Programming

Pie Coding Collaborative (<https://pie-coding.github.io>)

February 25th, 2015



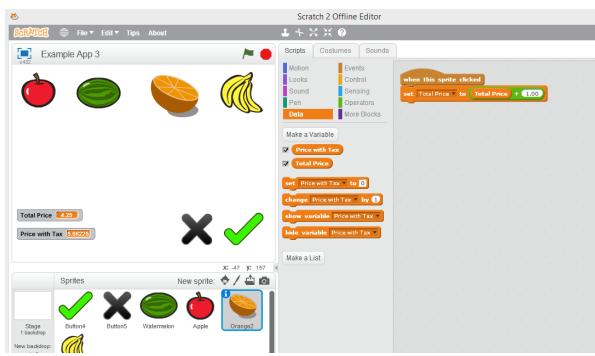
Now, we need to make sure our cancel button clears out any order. We will create an event that when it is clicked it will set both of our variables to 0.



Finally, we will give our “Checkout” button the ability to calculate tax. We will do that by using the formula **sales tax = total price * tax rate**. We will then add our total to that price. Using order of operations, we can construct our math to be:

$$\text{Total with Tax} = \text{Total Price} + (\text{Total Price} * \text{Sales Tax})$$

We just need to use two math operators nested inside of each other to achieve this.



And that is it! Hit “Run” and test your application out. Congrats; you have created your first fully functional application.

Next Steps:

Now that you have learned the fundamentals of programming, we recommend reading more about the concepts taught today. While different platforms will use a variety of programming languages, the fundamentals are the same. Below are some great reads for the novice programmer:

The Little Introduction to Programming - <http://codingintro.com/chapter/en/1/introduction>

Try Git (a popular source control application) - <https://try.github.io/levels/1/challenges/1>

Khan Academy's Intro to Programming course -

<https://www.khanacademy.org/computing/computer-programming>

Code Academy offers a great selection of free tutorials and classes that teach various programming languages - <http://www.codecademy.com/>

Coursera offers a plethora of free, online courses from acclaimed universities -

<https://www.coursera.org/>

EdX uses a platform similar to Coursera's and offers a variety of classes from recognized universities for free. <https://www.edx.org/>

If you want to get a head start on next week's programming class, check out the Learn Java Course (free): <http://www.learnjavaonline.org/>

EdX is offering an intro to Java programming course starting April 15th this year for free:

<https://www.edx.org/course/introduction-programming-java-part-1-uc3mx-it-1-1x#.VO1JIPnF98E>

Once you have the basics covered, think about what you'd like to do with programming. Do you want to make websites? You should look into front end development. Do you want to write apps for iOS or Android? You will need to look into those platforms. You will find countless of free online resources available. If you have any questions or would like further advice or guidance, do not hesitate to contact us! You can find our contact information on our website at <https://pie-coding.github.io/>