Introduction to App Development: Lesson 4

Loops & Timers

Before we get started, let's do some review.

- Last time, we used an IF Condition that looked like this:

- We could have actually replaced the equality to use the blocks below.

- What does the red block do?

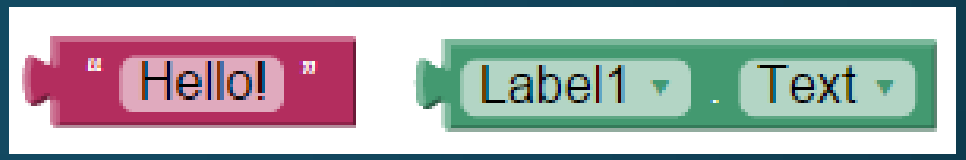
Find the “is empty” block and mouse over it.

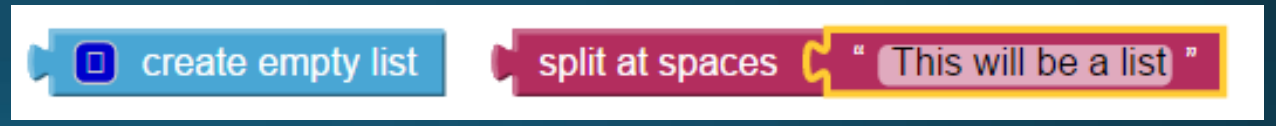
Pay attention to the word ***Returns.*** That word is referring to the output of the block

- The result is either true or false
- The *input type* is different from the *output type*
- *What is the input type?*
- *What are types? Do you remember any examples from the previous lectures?*

Types!

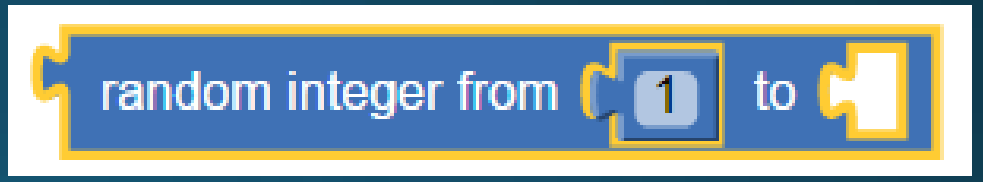
- Every value has a type
  - All the pairs of blocks here have different return types(output types)
- Here are all the Types in AppInventor
  - Text
  - Number
  - Boolean (true or false)
  - List
  - Colour
  - Object (This will be important tomorrow)
- Write these down, you'll need them for the next slide.





Slots have types too:

- What types are these ***arguments*** looking for?



Be sure to keep types in mind for today.

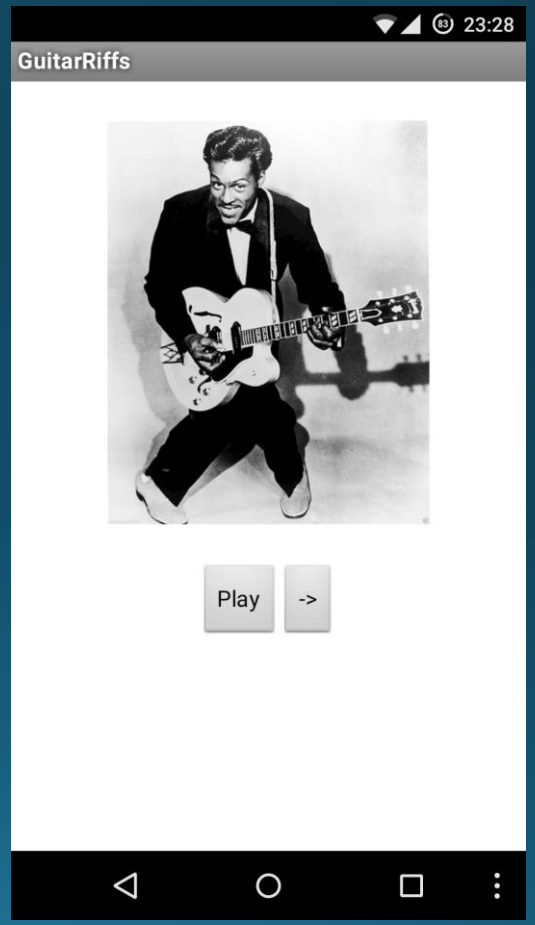
Hopefully this makes some blocks slightly less confusing.

Today's goal:

- To understand loops.
- We will use loops to do an action many times instead of just once.
- When the user clicks a button, we will draw many dots on the screen at the same time.

But, didn't we make a loop last time?

- Last lesson, we cycled through a set of mucisians whenever the user pressed a button.
- Today's concept is a little different.
- We want the computer to repeat **instructions** very quickly (almost in an instant).
- The result is, as with before (when we used a list), that we can save a lot of time and use less blocks.



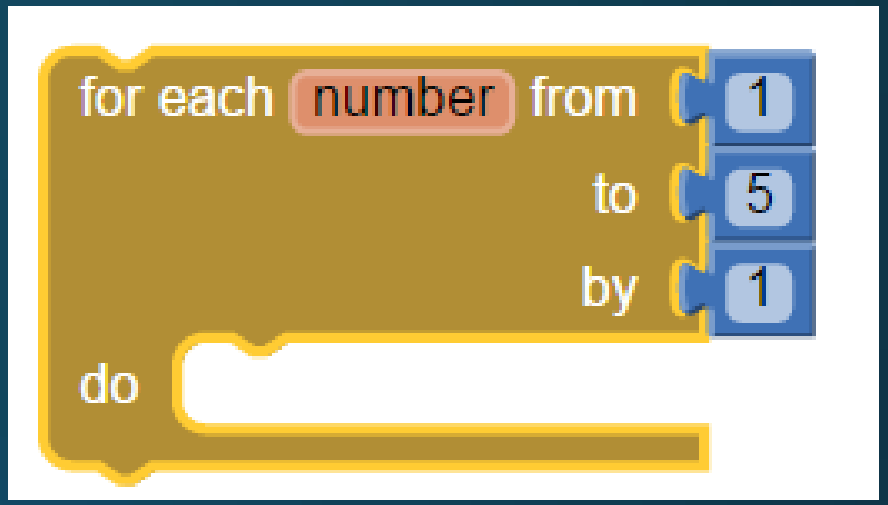
Let's look at Loops

Concept #8

- A loop is a control block that repeats an action several times.

- This loop block contains a number variable.
- In this case, the number variable:
  - Starts at 1 (the *from* slot)
  - Stops the loop at 5 (the *to* slot)
  - Increases by 1 every time the action happens.

- So, this block loops 5 times, because it goes from 1 to 5, by 1 each time.

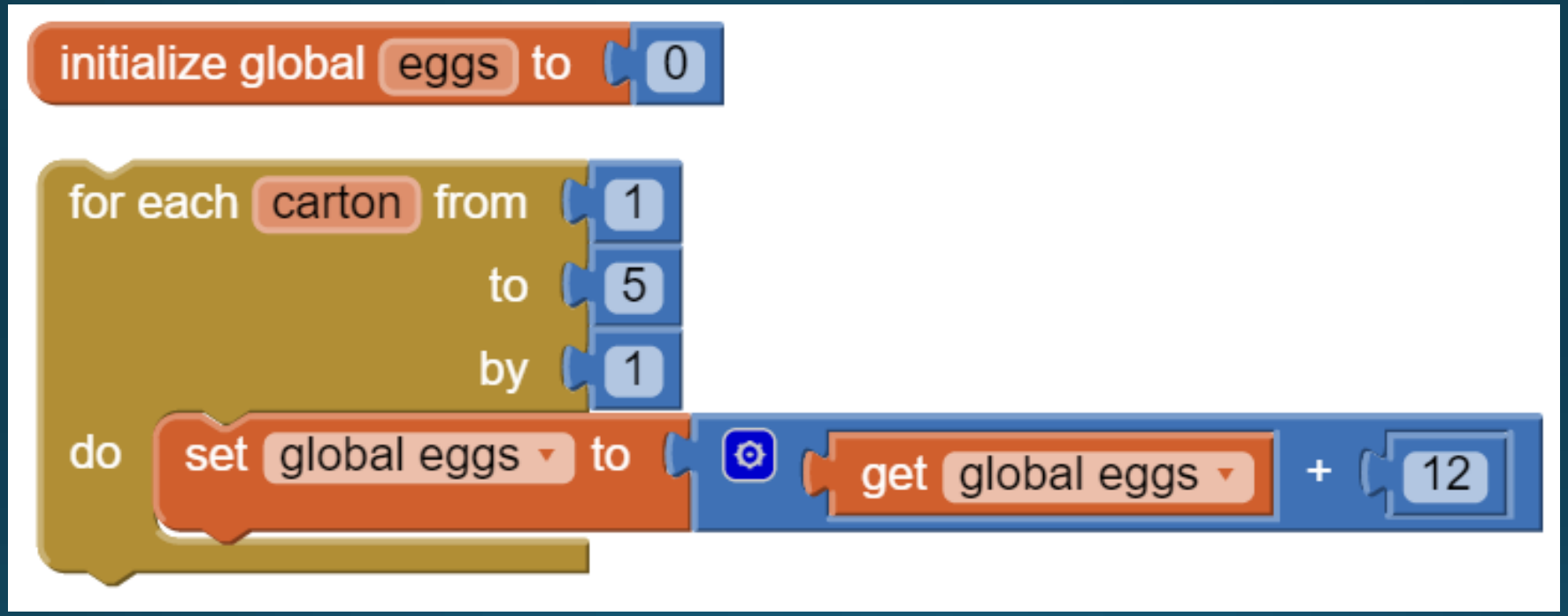


This kind of loop is called a FOR Loop

Let's look at some examples of loop iterations

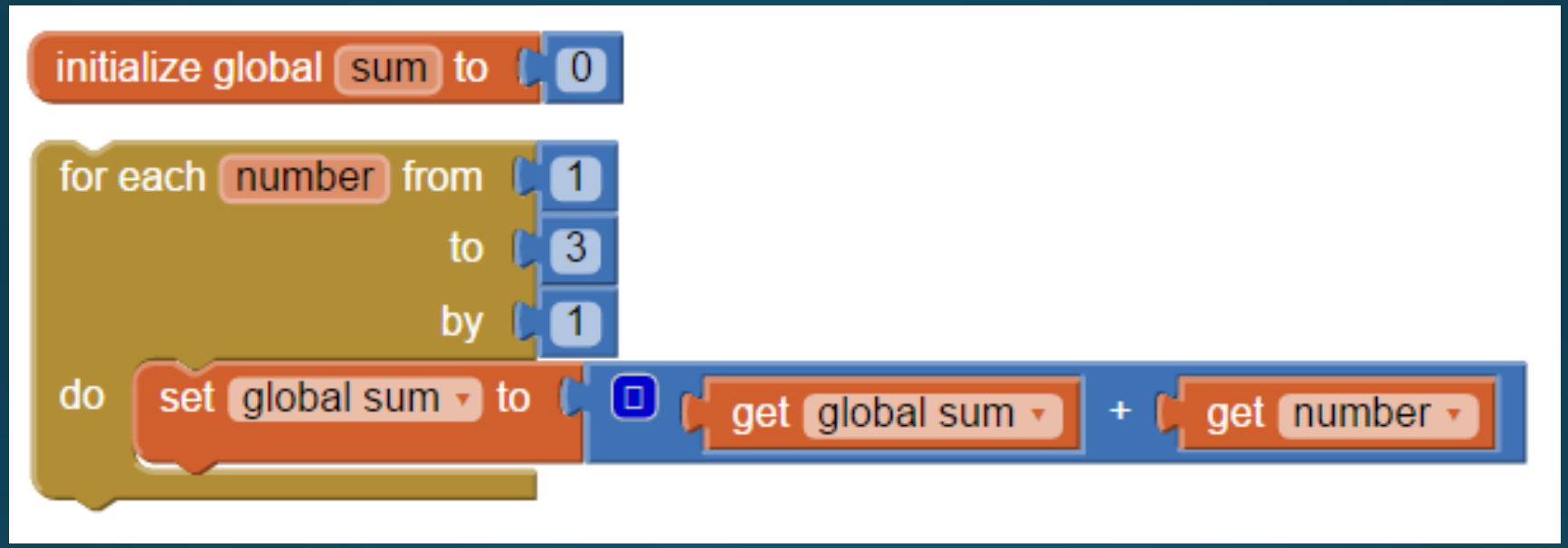
- From 1 to 5 by 1
  - 1, 2, 3, 4, 5
- From 1 to 5 by 2
  - 1, 3, 5
- From 0 to 10 by 2
  - 0, 2, 4, 6, 8, 10
- From 5 to 1 by -1
  - 5, 4, 3, 2, 1

Example 1: Number loop



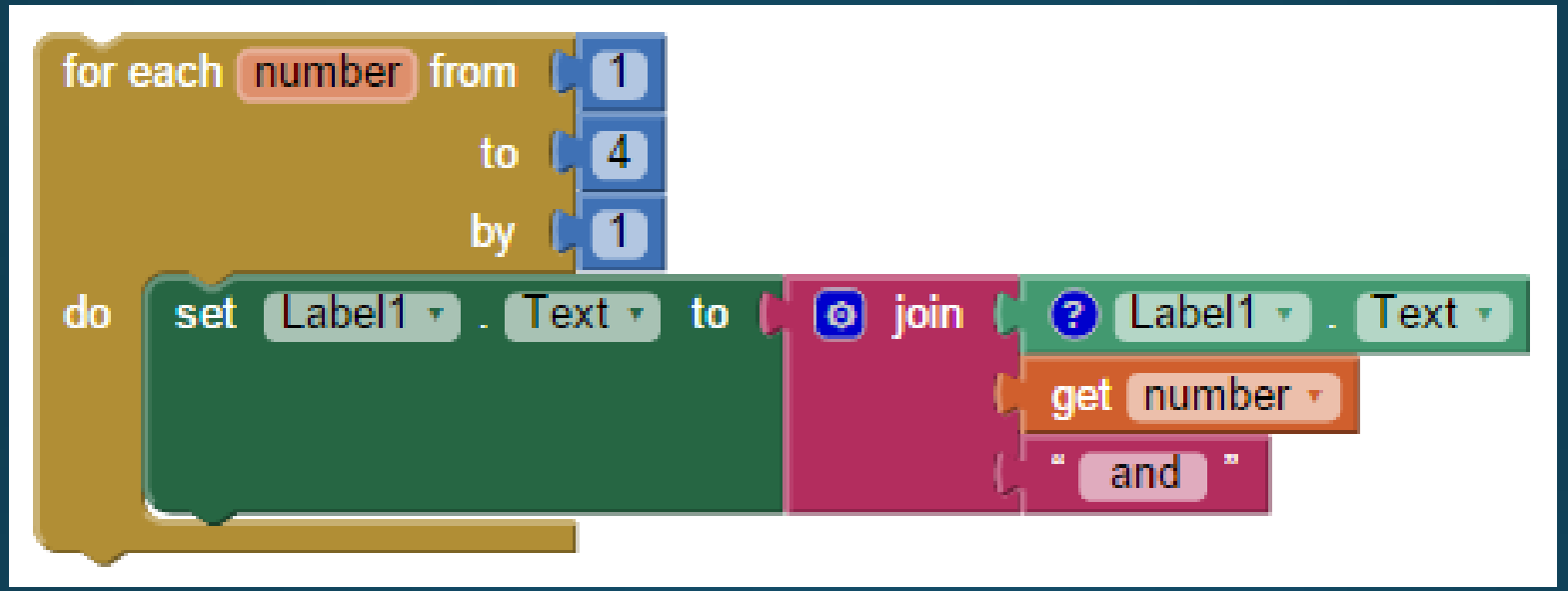
- How many loop iterations occur?
- What is the value of the eggs variable at the end?
- Do we use the **carton** variable at all?
- How many eggs do you have at the end?

Example 2: Number loop to calculate sum



- How many times does *this* loop run?
- What is the difference between the sum variable and the number variable?
- What is the value of the sum variable at the end?

Example 3: Adding text to a label



- If we change the by argument to 2:
  - How many loop iterations would occur?
  - What would the label text be at the end of the loop?

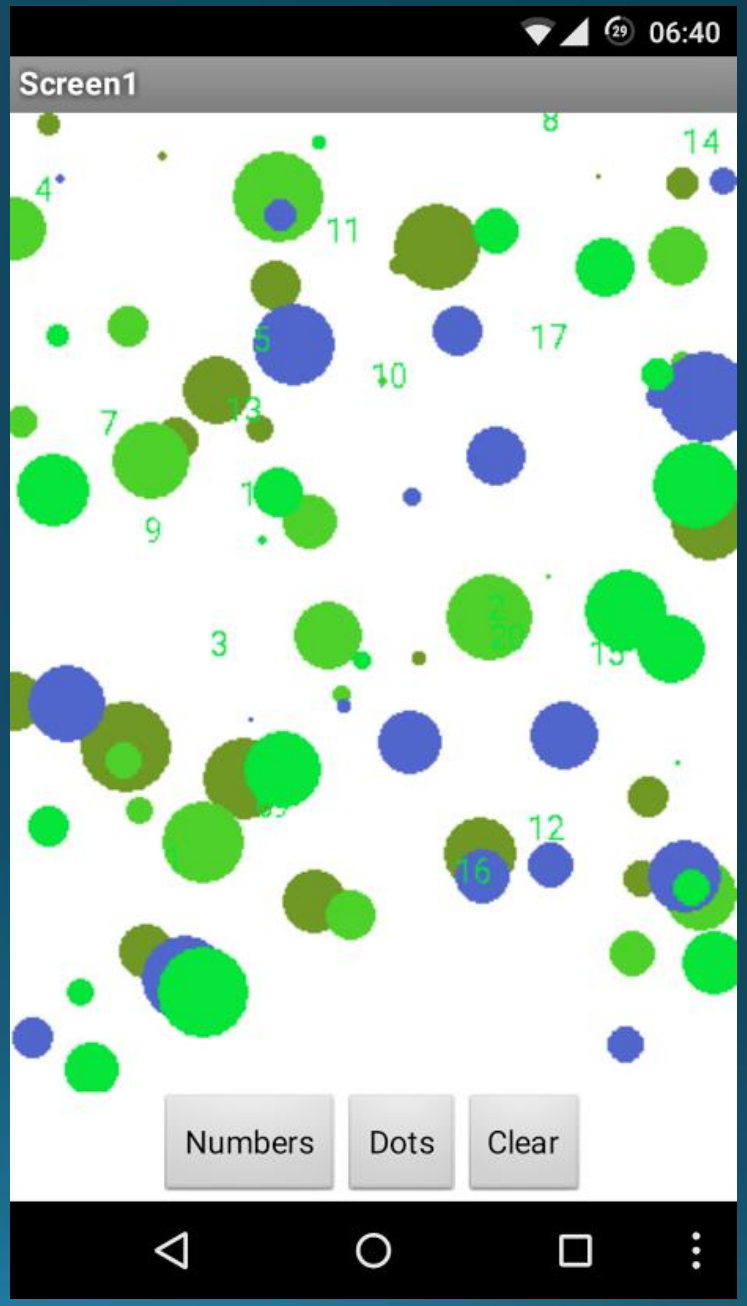
Iteration#	number	Label Text
First	1	"1 and "
Second	2	"1 and 2 and "
Third	3	"1 and 2 and 3 and "
Fourth	4	"1 and 2 and 3 and 4 and "

- The loop happens in the blink of an eye, so you only see the end result!

Loops are everywhere in software

- Loops are mainly used in conjunction with lists. As you saw last session, lists are very powerful, and allow us to scale our apps with little effort.
- Often, when you see a list, there is a loop that creates it.
- Today, we'll try to explain loops without using lists by creating a visualization app, then we'll use loops in a more practical application.

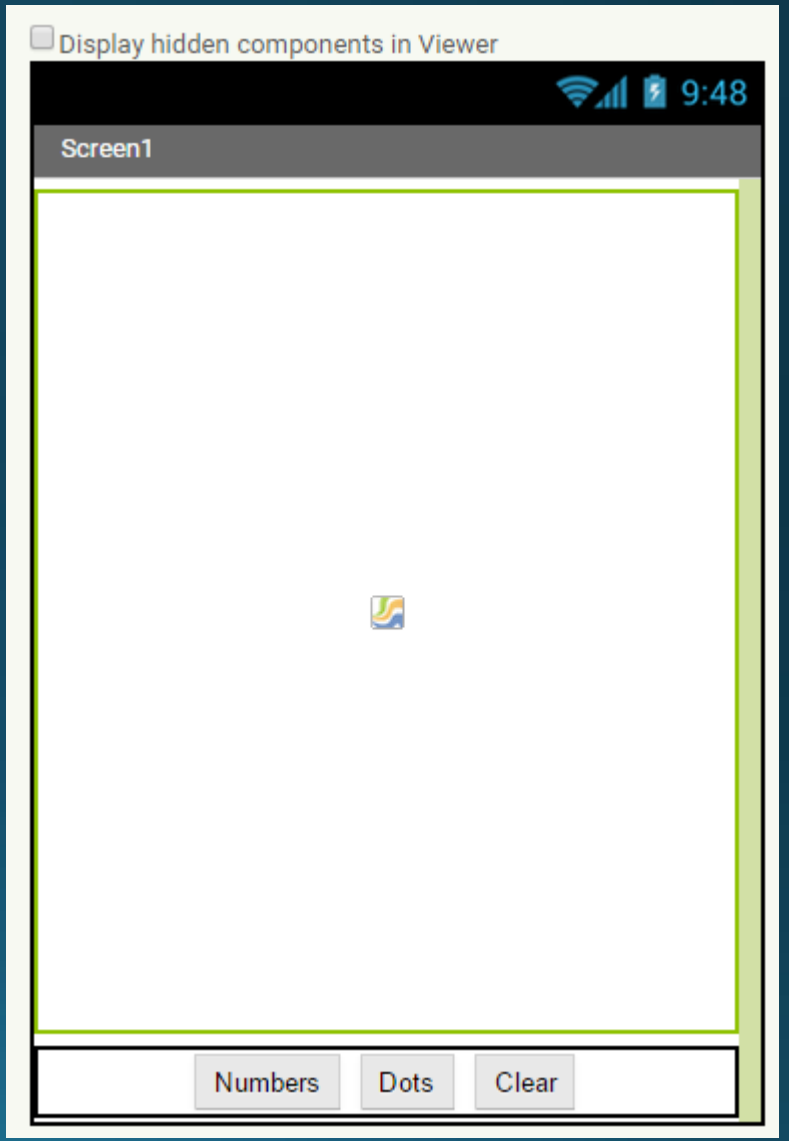
Today's First App: PollockPainter



Does this look cool?

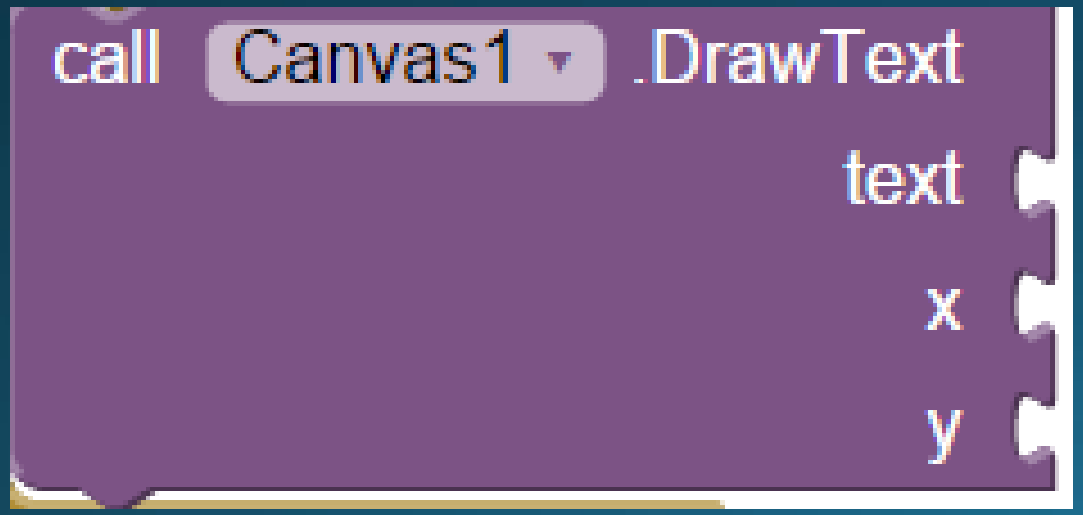
Create a new Project

- Call the project 'PollockPainter'.
- First steps:
- Drag in a Canvas Component
  - Set its Width and Height to 'Fill Parent'
- Drag in a Horizontal Arrangement under the Canvas
  - Set its AlignHorizontal to Center
- Drag **three** Buttons into the Horizontal Arrangement
  - Rename the first to NumbersButton (and Text to 'Numbers')
  - Rename the second to DotsButton (and Text to 'Dots')
  - Rename the third to ClearButton (and Text to Clear)

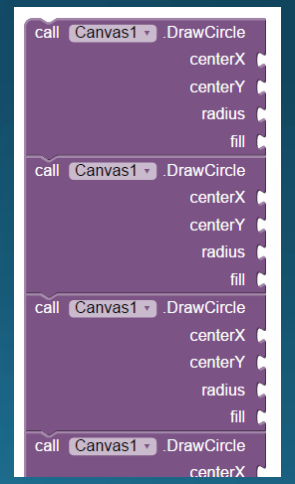


Let's draw some numbers

- When the NumbersButton is clicked, we want to draw the numbers from 1 to 10 on the canvas
- We could use this block:



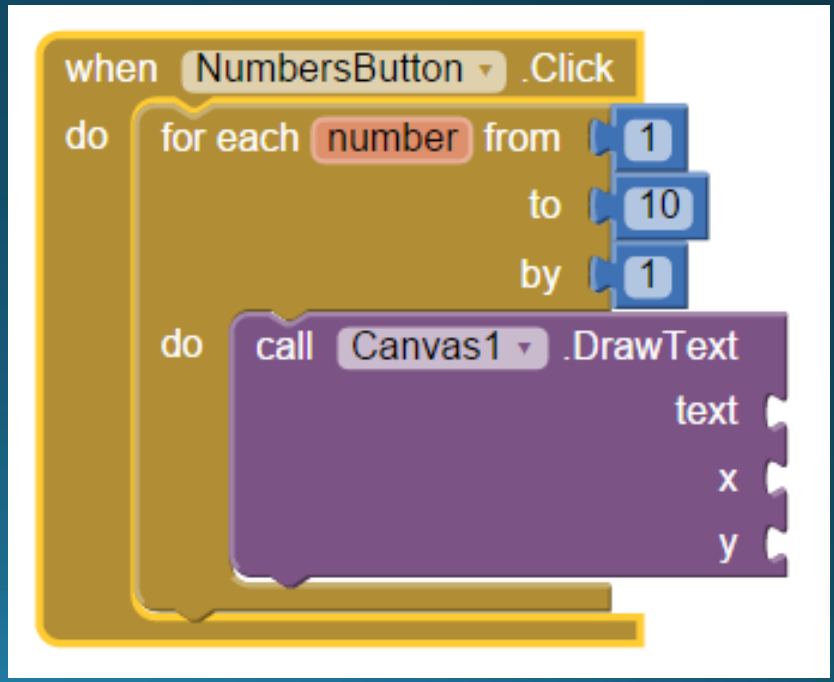
But repeating 10 times?



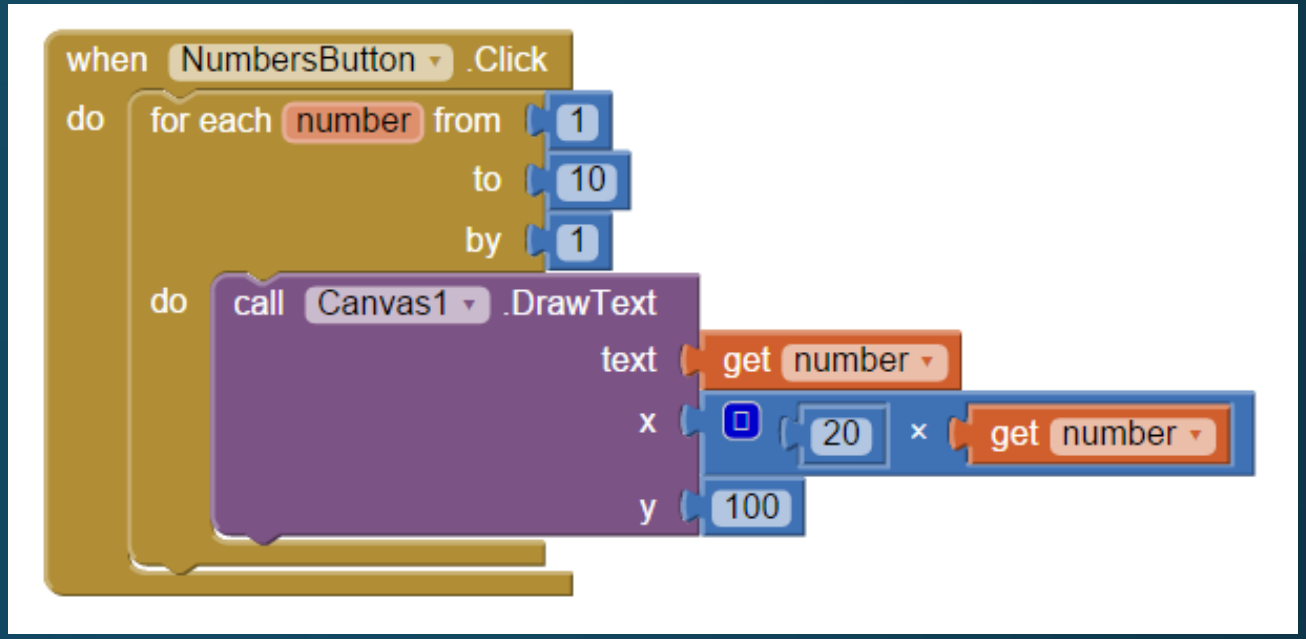
No thanks…

Let's make a Loop!

- Lets draw the numbers 1 to 10 on the canvas.
  - When the NumbersButton is Clicked, we will loop 10 times, and draw the number at each loop iteration (1 to 10).
- We'll use the loop block with a DrawText block inside it.
- We want to draw the numbers across the screen. What do we need to put in the empty slots of the DrawText block?



The finished Loop

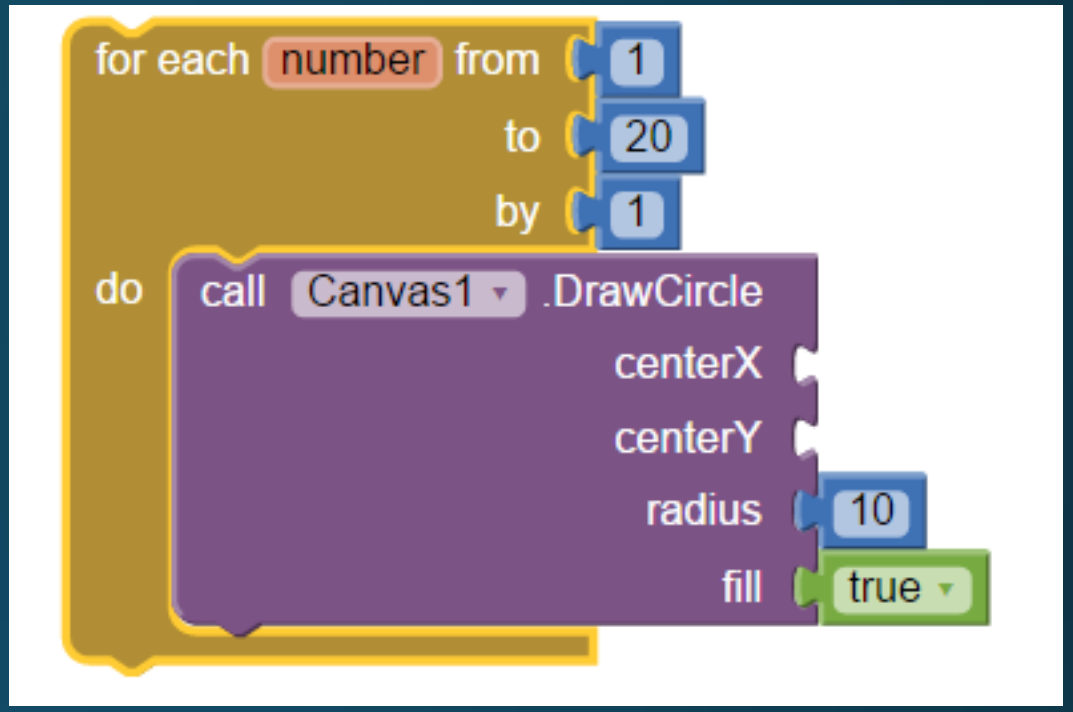


- For the text slot, we use the loop variable (called number), which changes at each loop iteration.
- For the x slot, we will multiply the number by 20 (so it will be 20, 40, 60, etc) so that they will be drawn progressively to the right.
- For the y slot, we'll always use 100 to keep the text at a constant height.

- Now, test the app! What do you see when you click the button?

Drawing random circles

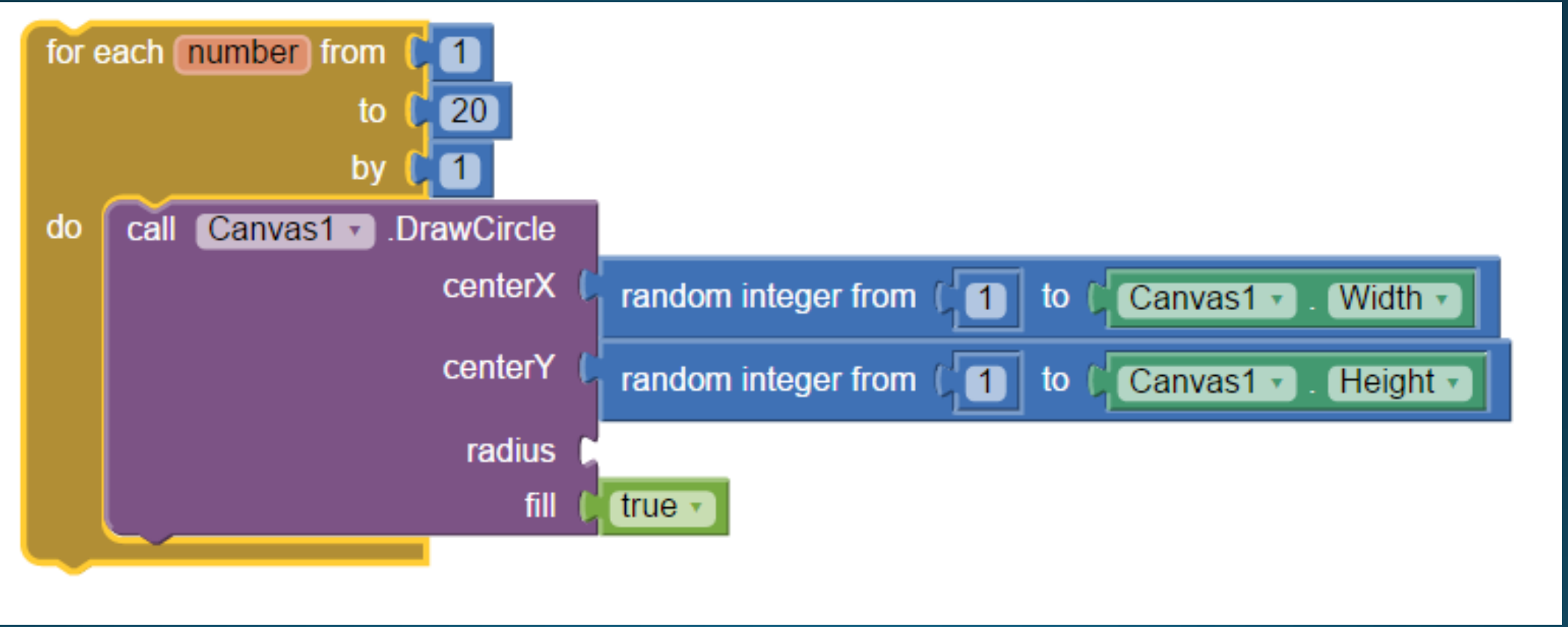
- When the DotsButton is clicked, we want to instantly draw 20 circles in **random** places on the canvas.



- Let's use another loop, this time we'll use the DotsButton event, DrawCircle, and count the loop to 20 instead.

- Mini challenge: try and fill in the centerX and centerY slots yourself.

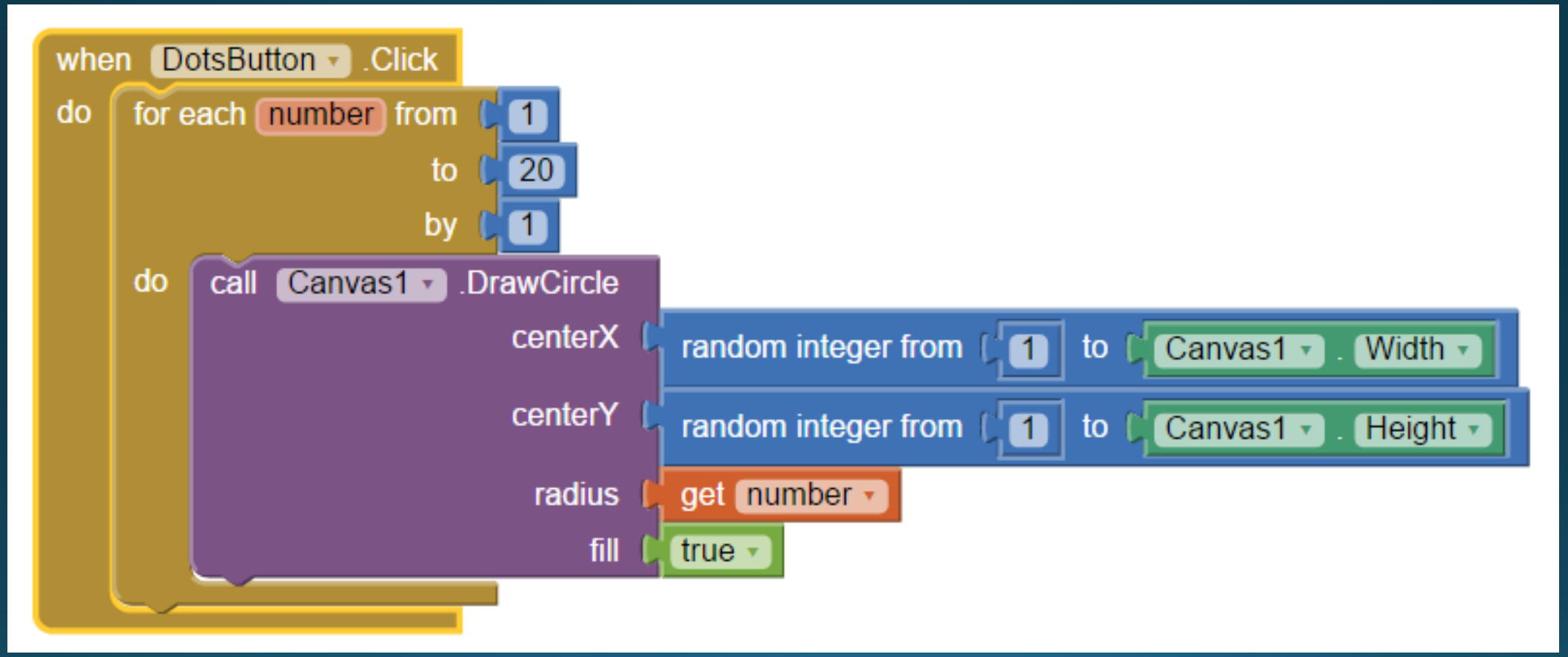
Next step:



With each circle we draw, from 1 to 20, the radius should increase.

How would we do that?

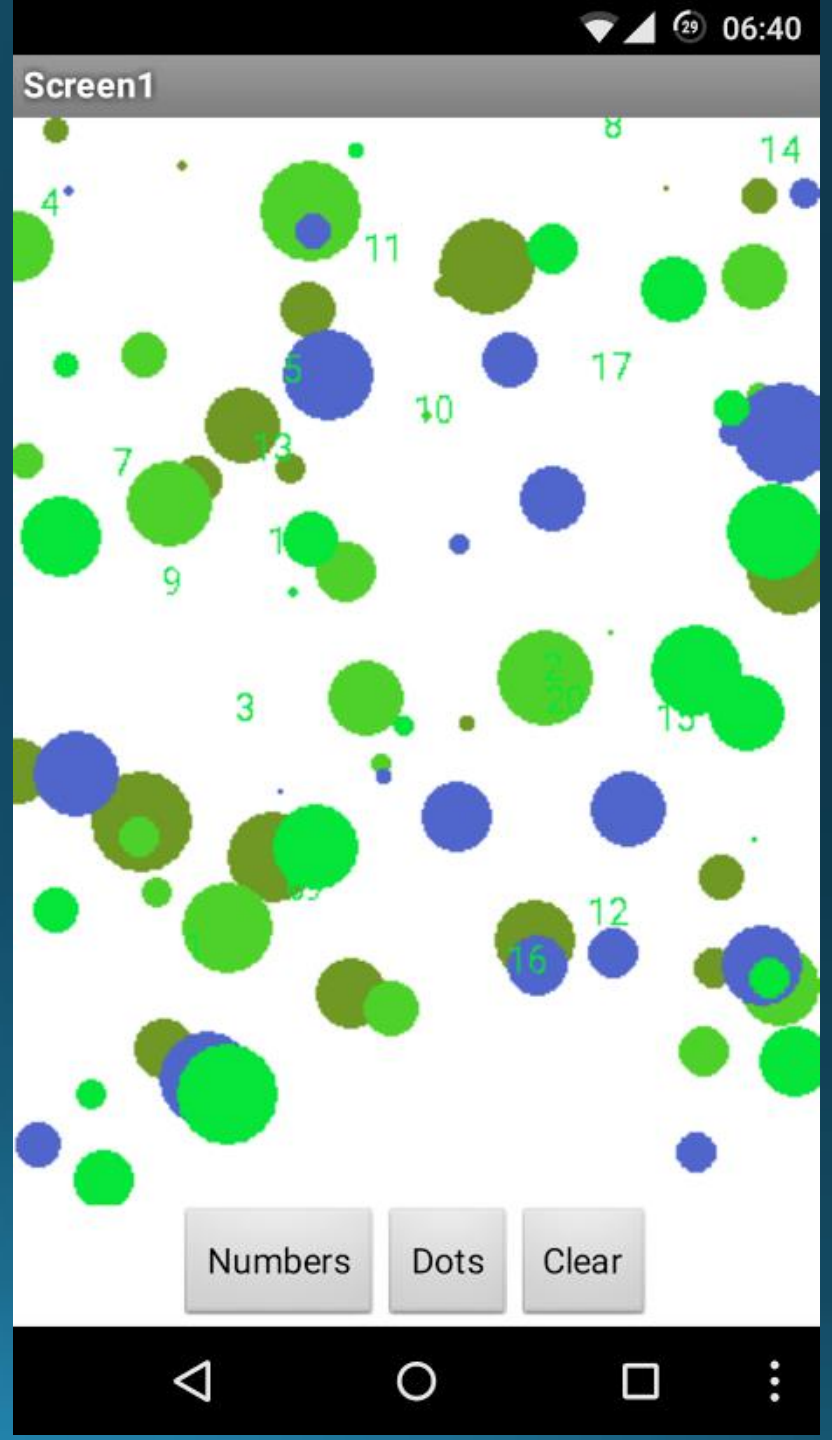
We use the loop variable



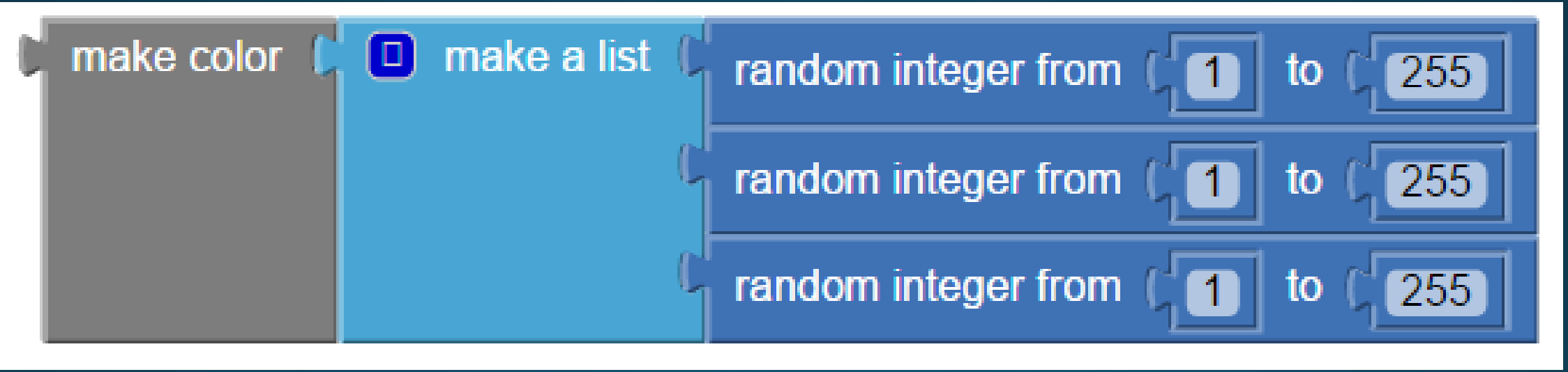
Test the App.

Let's make it Colourful!

- Before painting each new batch of circles, we want to change the Colour to a **random** colour
- Look at the Blocks in the Colour menu. Is there a single block for random colour?
- Using the blocks you see there, how do you think we can do it?



Random Colour

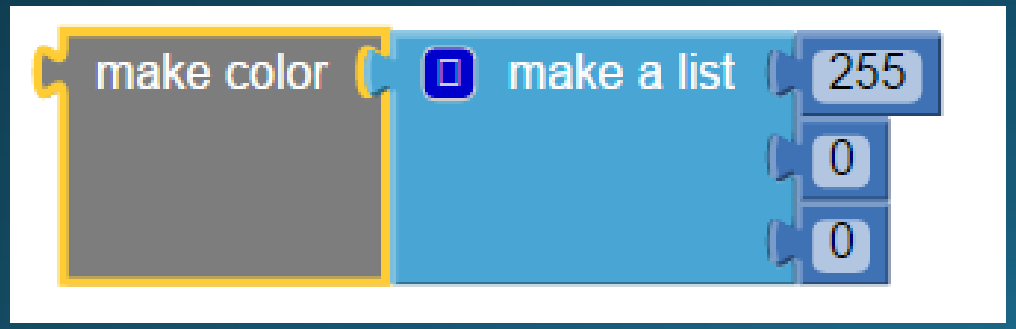


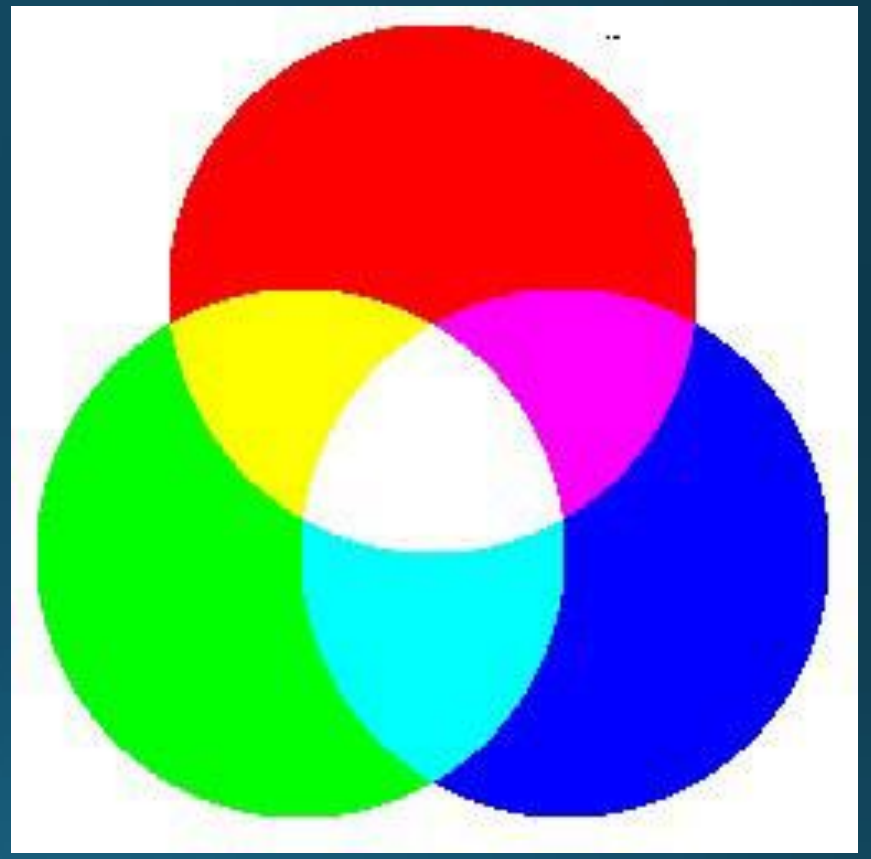
What is the Input Type (slot) of the **make Colour** block?

What is the **Output type**? (peg)

Colour by Numbers

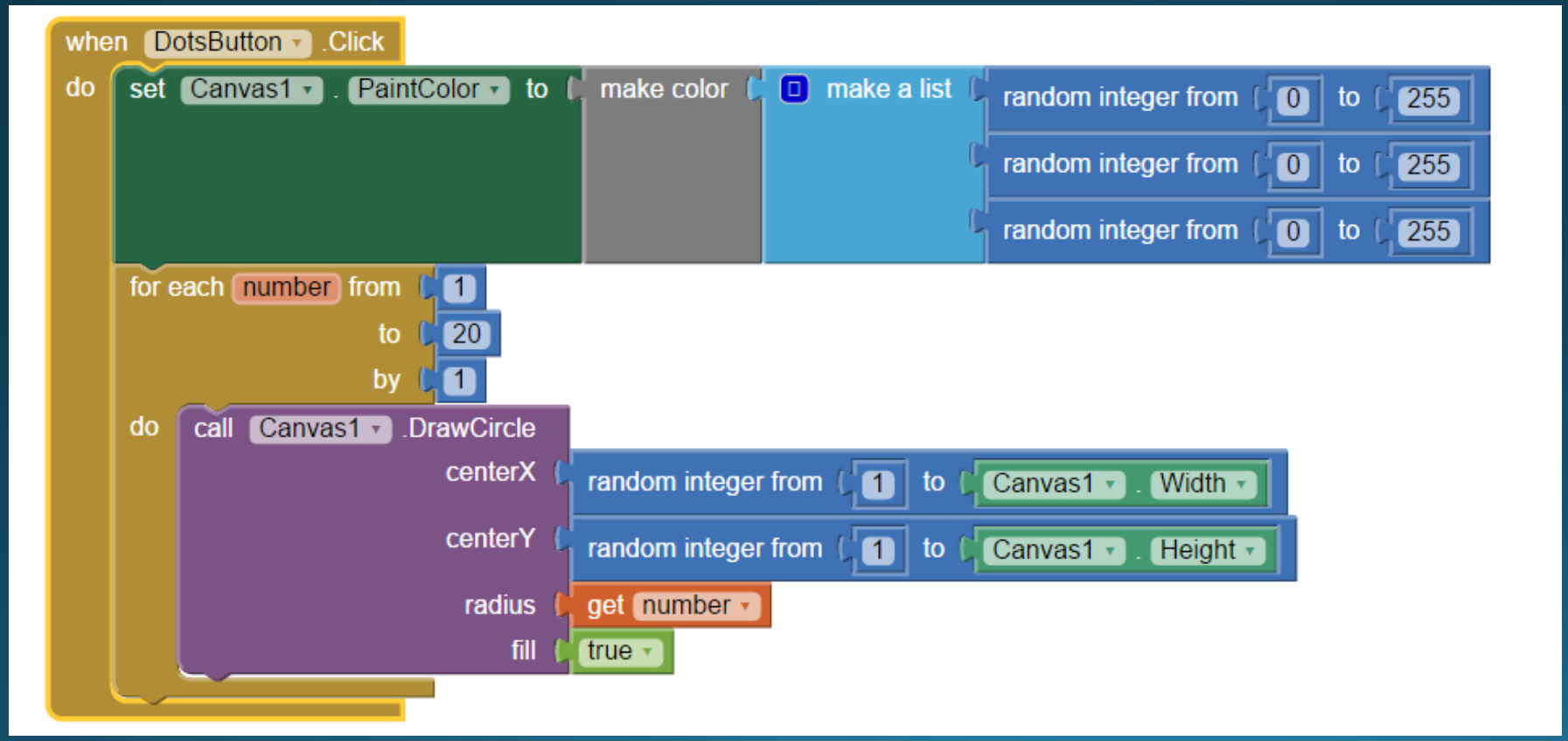
- The block we just used makes a Colour by specifying the Red, Green and Blue values of that Colour.
- What do you think this Colour is?





Random Colour

- Every time, before drawing the circles, set the Canvas Colour to the random colour.
- Test your app, what happens when you press the Numbers Button?
- What would happen if you set the colour ***inside*** the for loop. Try it!



More loops ahead

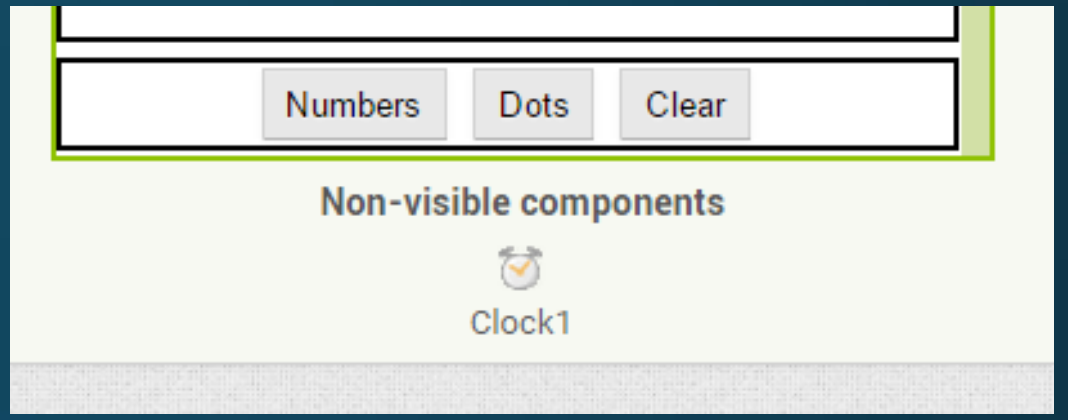
Take a short break

Let's take some time.

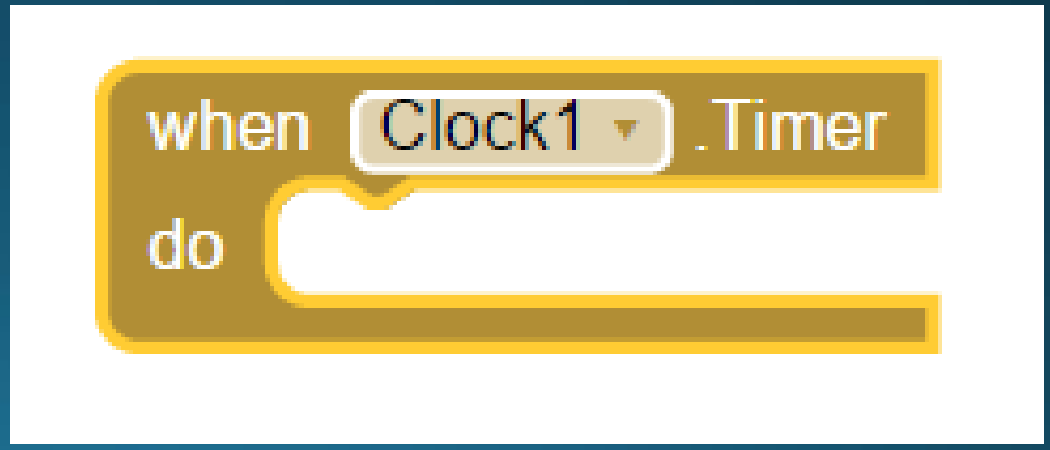
- How fast does your phone operate?
- Computers are really fast at doing certain things.
- When we make a loop to draw 20 circles one after another, it happens so quickly that it seems they were all drawn at the same time!
- We can use a timer to make this happen slowly

Timer

- In your designer view, under Sensors, drag in a Clock Component.



- In the blocks view, from the Clock1 Menu drag out Clock1.Timer event.

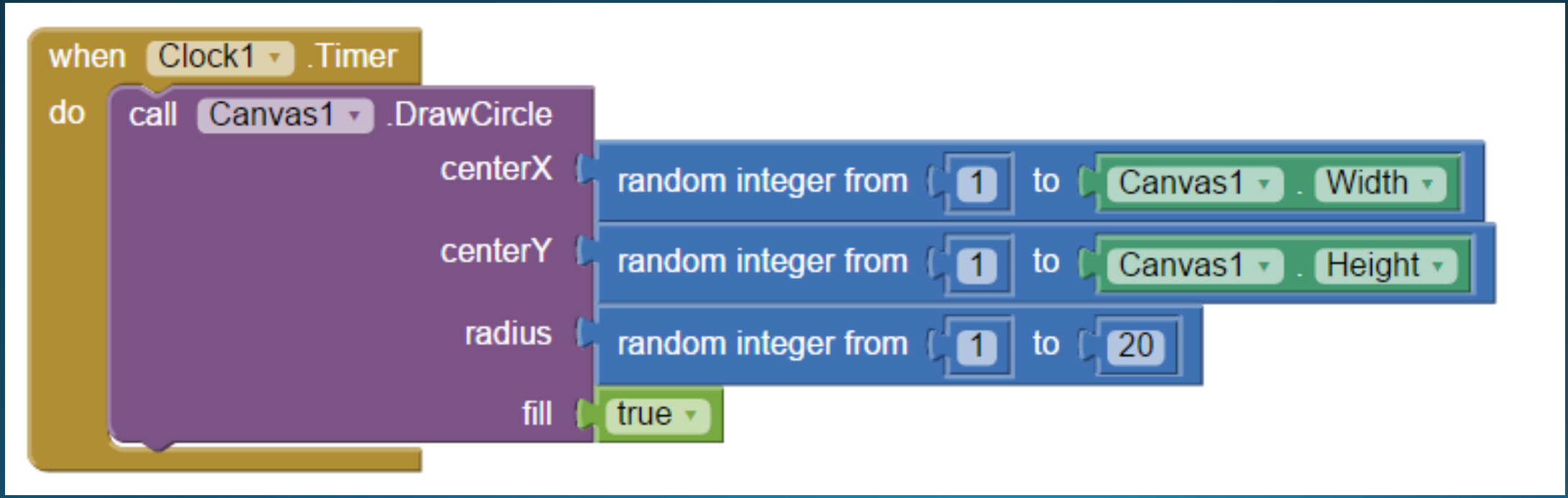


Time to code.

- The code inside the Clock1.Timer event will execute once every second.
- In essence, you're making a really slow loop!
- Let's try to draw a dot every second.

Blocks:

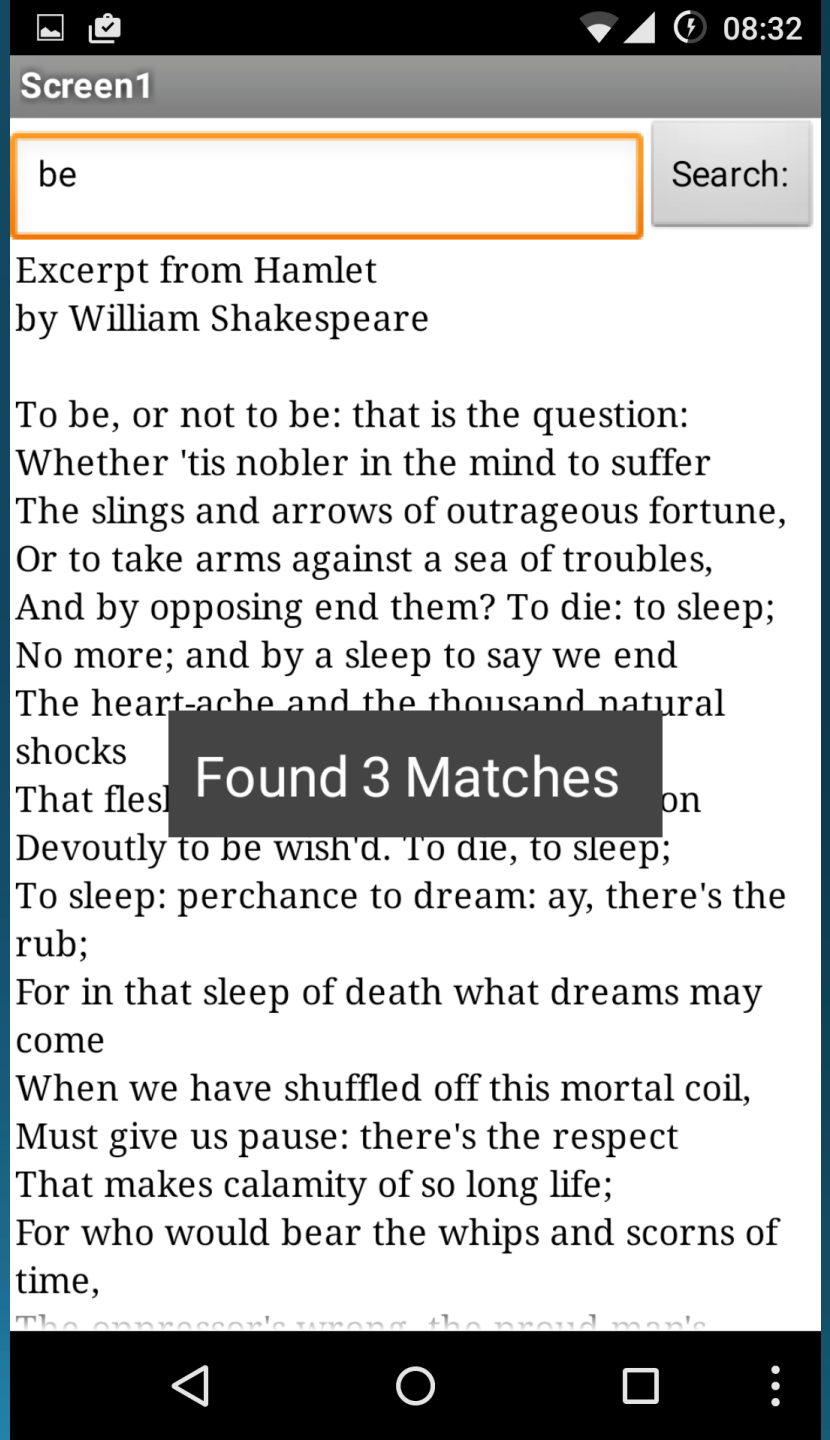
- Copy the drawCircle block from the DotsButton.Click event.
- Since we don't have a loop variable for the radius anymore, let's use a random number instead.



Try the app!

- The app was meant to show you the strengths of loops and timers.
- Can you think of any way to make the app more interactive?
  - Perhaps using the Canvas touch events?
  - Try drawing lines
  - Adding balls/sprites
- Let's take a short break, Think about it.

Today's Second App: ShakesParser



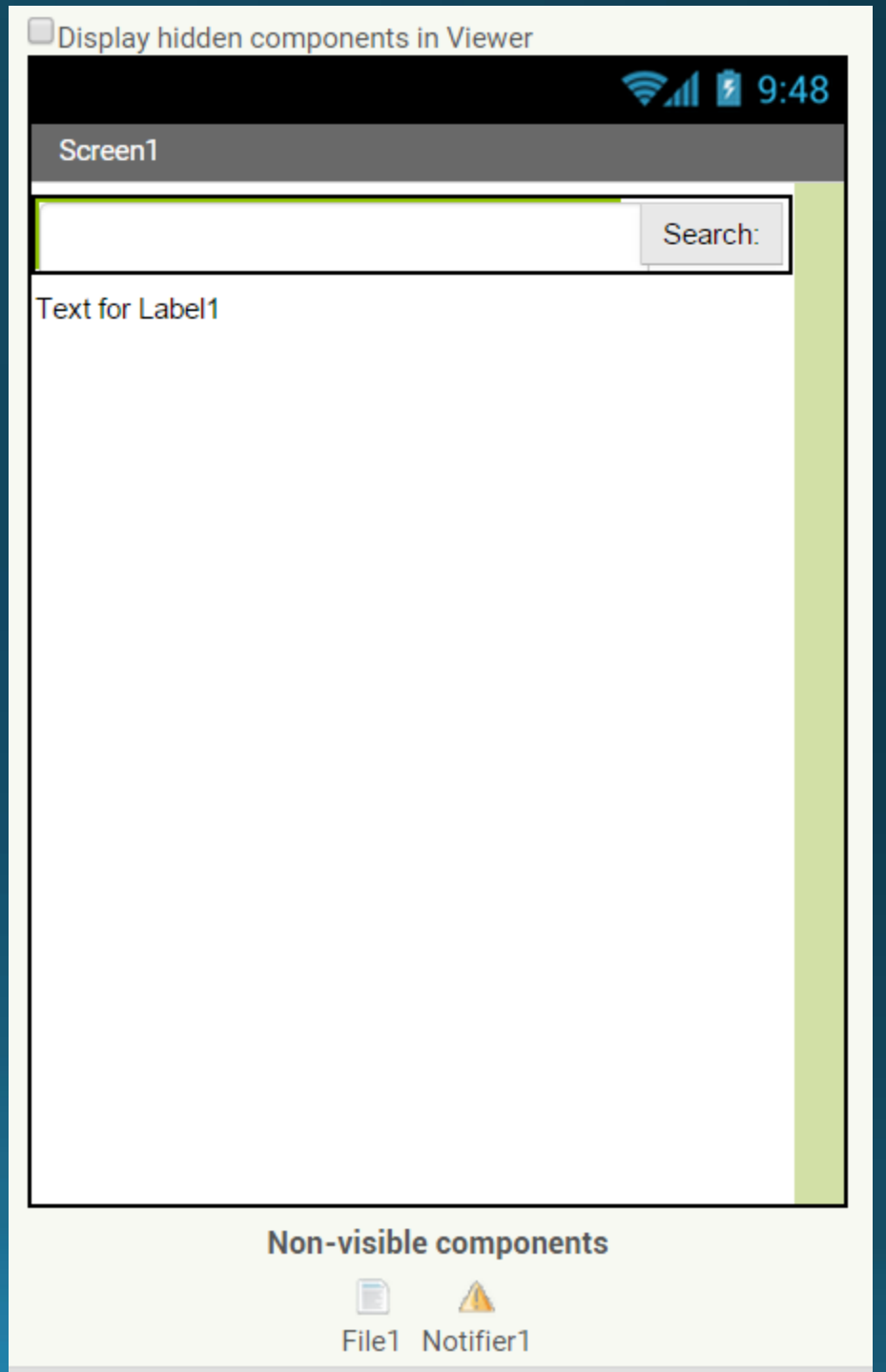
How did google get rich?

- With one weird trick: Search!
- Today we'll learn an Algorithm!

- We'll search a block of text for a specific word that the user enters.

- We'll count the number of times it appears.

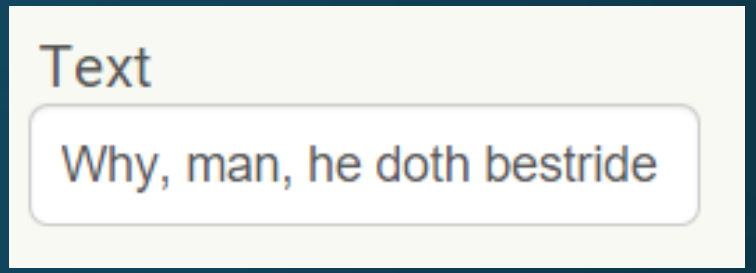
- As usual, make a new Project:
  - Name it ShakesParser
- Drag in the following components:
  - HorizontalArrangement
    - Set the width property to fill parent
  - Textbox (Inside the Horiz. Arr.)
  - Button (Inside the Horiz. Arr.)
    - Set the text property to "Search!"
  - Label
  - File (Located in the Storage tab)
  - Notifier
- Also, on the Screen1 Component, set the Scrollable property to True.



Let's add some text to search:

- Open your browser to this website:

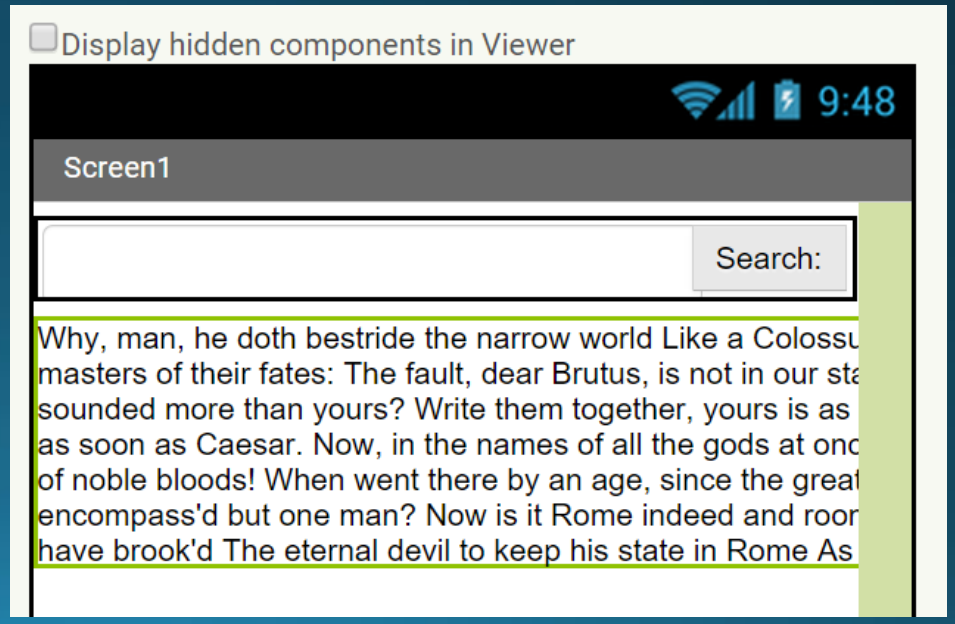
http://goo.gl/64NsJK



- Copy a **small** paragraph of text (Ctrl+C or Cmd+C)

- Paste it into the text property of your label (Ctrl+V or Cmd+V)

- Your Viewer may look a bit weird, but that should be fine.

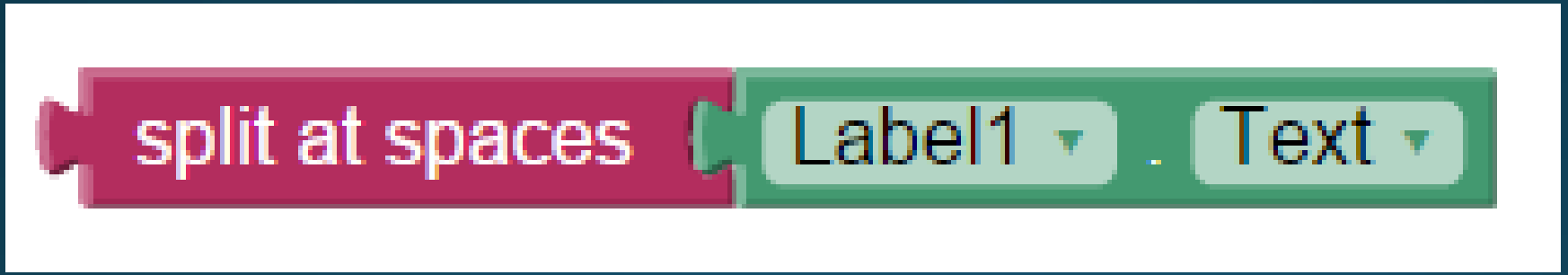


So, what is the plan:

1. First, we'll convert the text into a list of words
2. Next, we'll iterate over each word.
3. If the word equals the word that the user enters
   - Increment a counter
4. Finally, we'll show a notifier alert with the results.

1. Convert text to list of words:

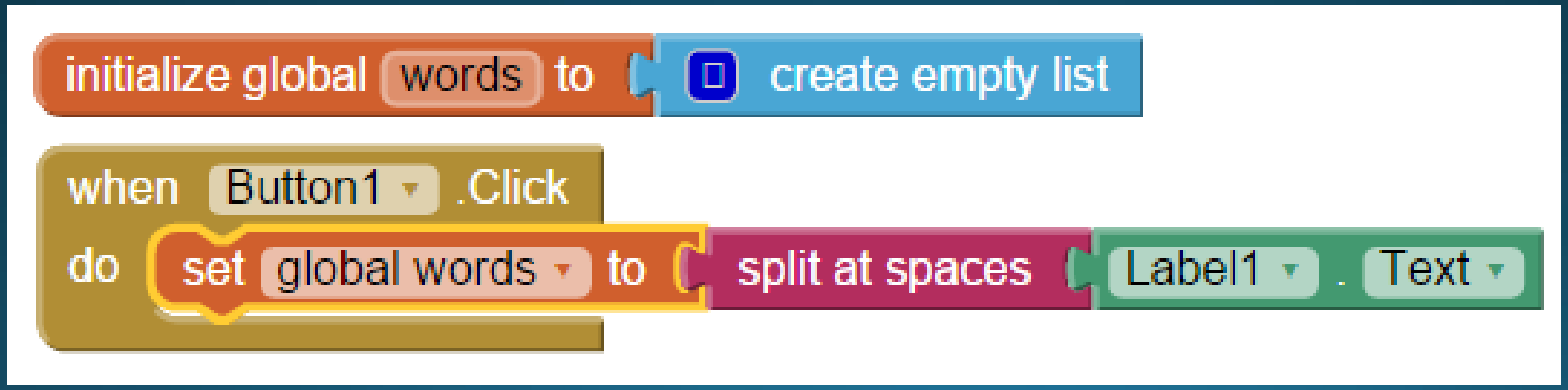
- This is the block that we will use to convert the text:



- What does this block do?
- How do we store the result to access it later?

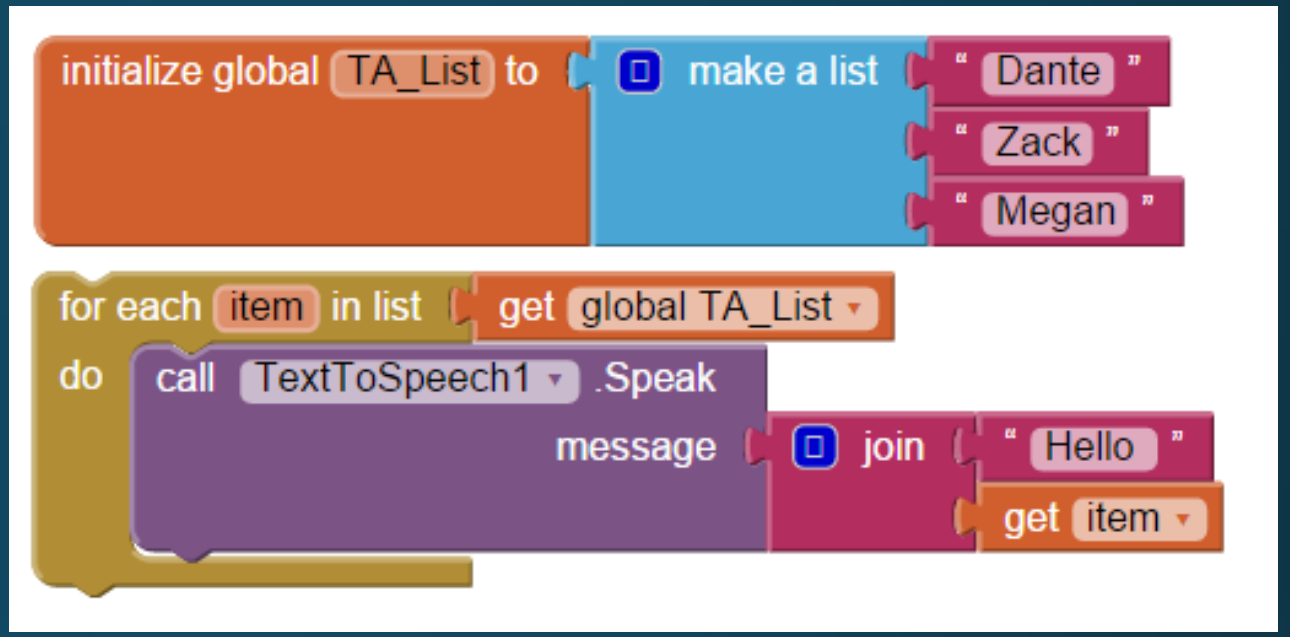
Storing the words.

- In order to store the words, we'll have to use a variable
- Because we're using a property (properties are not allowed in variable declarations), we'll have to initialize it to an empty list.
- Then, in the Button.Click event, we'll fill it up.



“For Each” loop

Concept #9

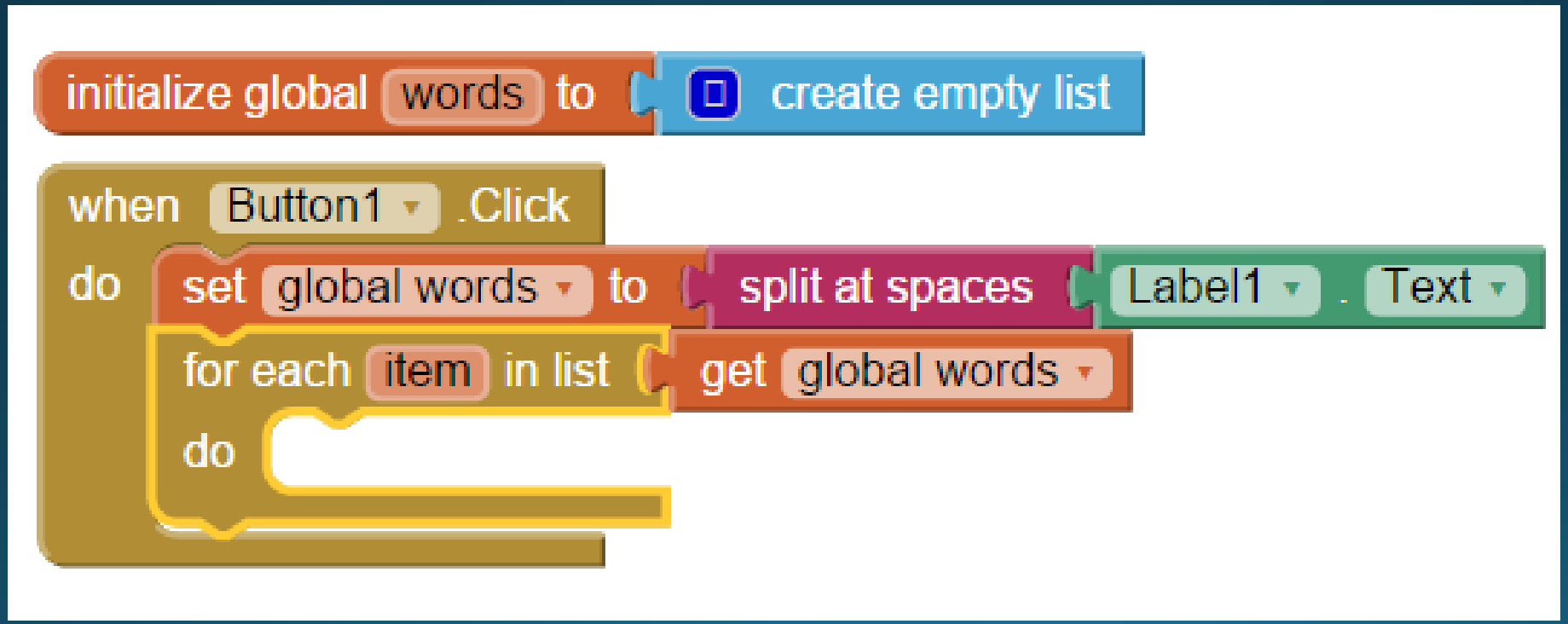


- Here is a different loop block: the List Loop block.
- With this block, we loop over every item in a list, one item at a time.
- With each iteration, the list item is stored in the item variable.

Iteration#	item variable
First	"Dante"
Second	"Zack"
Third	"Megan"

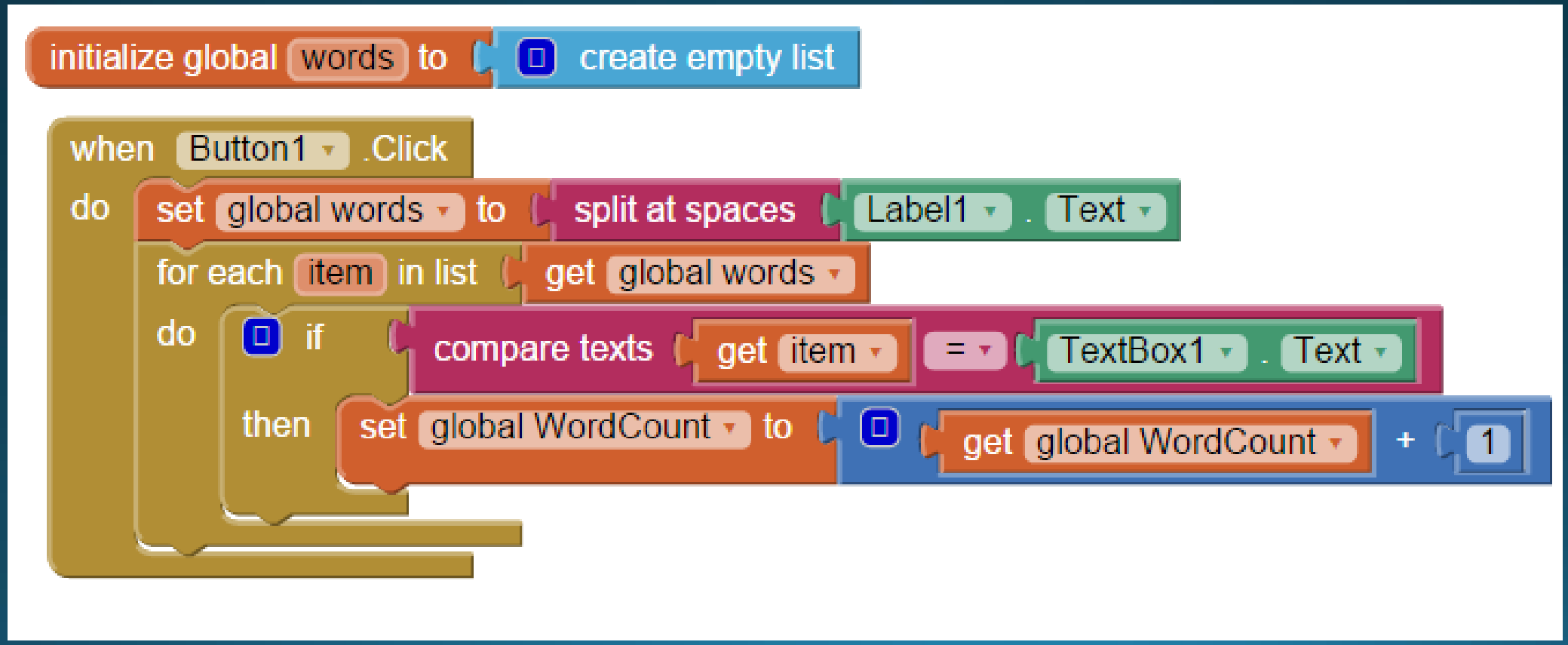
2. Iterate over the words.

- Now, we can use a for each item loop to iterate over the words.



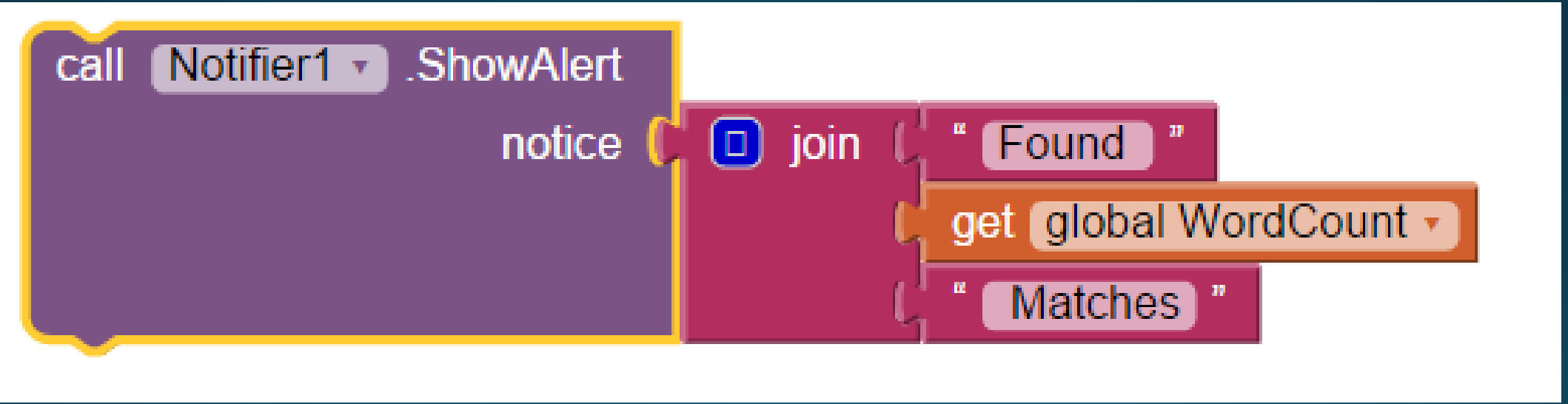
3.Comparing text

- Let's use our trusty if block to check if the word is in the text.
- What does the variable "item" mean in this case?



Show Notifier

- Let's make a fancy notification:



- Where does this block go? Be Specific!

Test your app: Does it work?

Nope.

- There are **three** crucial bugs!
- Your challenge for today is to find and fix all three of these bugs.
- Testing for logical errors in a program is a crucial part of developing software.

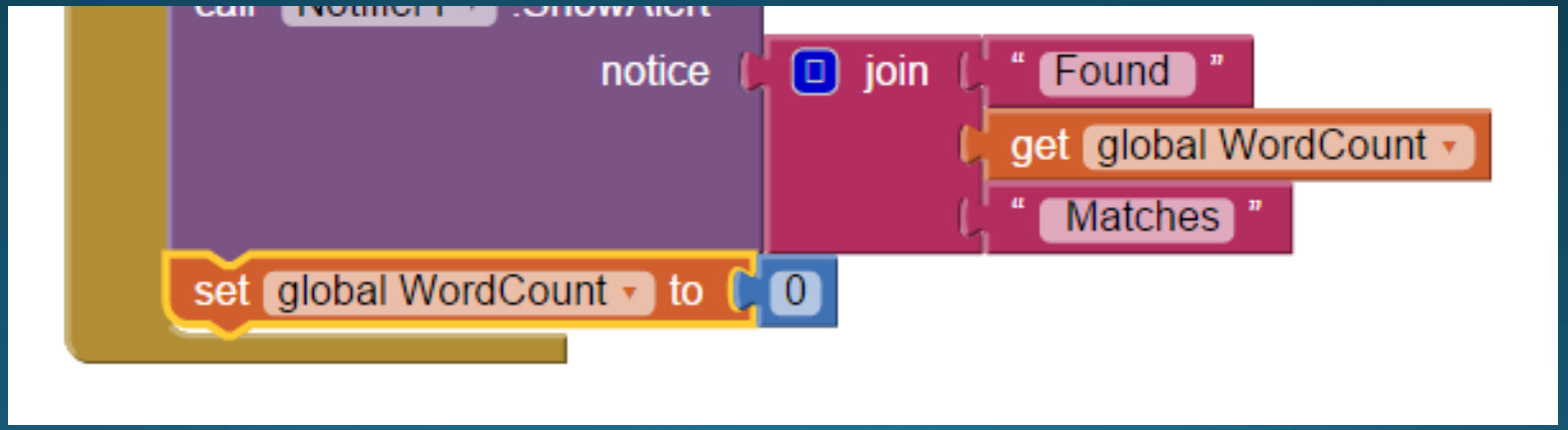
Please fill out your surveys
before you leave!

Take a short break, then..

Debug.

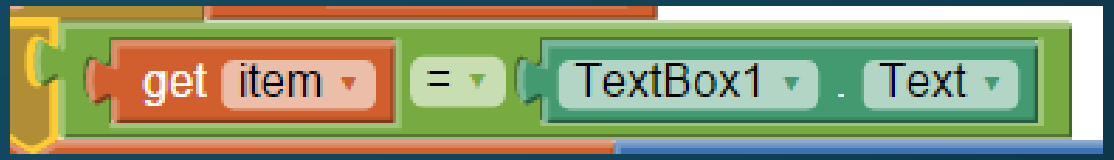
Bug #1

- Every search returns a bigger number of entries!
- Reason:
  - We forgot to reset the Count variable after iterating
- Fix:



Bug #2

- Words with capitals are not counted!



- Reason:
  - The equals block checks for exact matches.
- Fix:

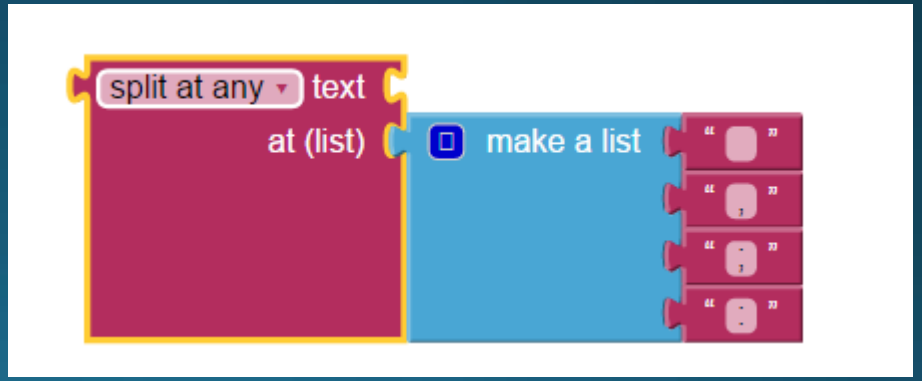
- Compare everything in uppercase

Bug #3

- Punctuation words are not counted!

- Reason:
  - the split at spaces block does leave words attached to punctuation

- Fix:
  - Use the Split At Any block instead
  - You will have to make a list with:
    - Space
    - Other punctuation you want to skip.



Test your app:

- Does it work?

- Is it bug-free?
  - Unfortunately, you can never say that a program is bug free. There is always a chance for something you didn't expect to happen.