Welcome to Lecture 4: Iteration

Class will start at 1:10.

In the meantime, we will go around. Tell me your name & favorite aquatic (water) animal

Today’s Topics

- Announcements
- Pre-Semester Survey
- Review
- **Variables + Functions**
- **Iteration**
- **Lists**
- Practice Problems!
- Summary

Announcements

- **Log into iClicker**
- Project 1 Party today from 3 to 5PM in Cory 400
- Project 1 due Wednesday, 2/5
- Lecture 2: Functions - didn’t record, but we posted old recording
    - Lecture Quiz deadline pushed to Tuesday, 2/4
- Your first exam will be in about two weeks! More info to come next week
- Added CS10 recently? Read this EdStem post

Pre-Semester Survey Results

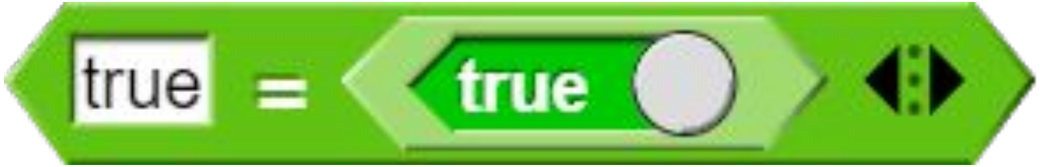
- I asked y’all, what grade you think you’re capable of getting
    - Every single one of you are capable of an A. In **ANY** class.
- *“What is the best learning skill for green hand” (beginner)*
    - Patience and keep trying! Have fun with it!
- *“Would like to know if I would be able to learn some other languages (i.e Solidity) by myself after this course.”*
    - Yeah, definitely!

Pre-Semester Survey Results

- Dumbledore - Harry Potter
- Iron Man - Iron Man
- Spider Man - Spider Man
- Fred or George Weasley - Harry Potter
- Voldemort - Harry Potter
- Dr. Strange - Dr. Strange / Avengers
- Grand Master Oogway - Kung Fu Panda
- Jess Mariano - Gilmore Girls
- Anakin - Star Wars
- Naruto - Naruto
- Ms. Incredible - The Incredibles
- Richard - Silicon Valley.
- Violet from Violet Ever Garden
- Uncle Iroh - Avatar last airbender
- Nausicaä from Nausicaä of the Valley of the Wind
- Linus - Charlie Brown / Peanuts
- Remy - Ratatouille
- Lorax - The Lorax
- Venom - Venom
- Ash Ketchum - Pokemon
- Magikarp - Pokemon
- SpongeBob SquarePants - SpongeBob SquarePants
- Tony Stark - Iron Man
- Bobby Axelrod - Billions
- Flash - The Flash
- Peach - Mario Brothers
- Miyamoto Musashi - Book of Five Rings
- Rick - Rick and Morty
- Li Liana
- Jonathan Joestar - JoJo’s Bizarre Adventure
- Phineas and Ferb - Phineas and Ferb
- Bilbo Baggins - Lord of the Rings
- Yoda - Star Wars
- Lucy Gray - Hunger Games: The Ballad of Songbirds and Snakes

Review from Last Lectures

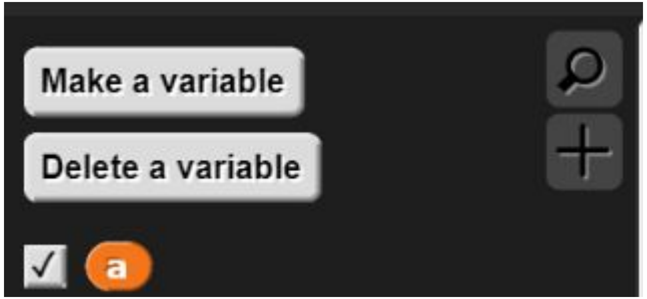
- Domain + Range
  - Domain: Input
  - Range: Output
  - There can be restrictions on data types for domain and/or range
  - Question:

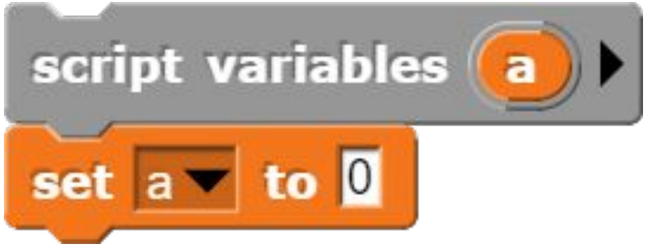


- Conditionals
  - If statements: A control expression. The domain is a Boolean.
  - If the statement evaluates to true, then code is run.

Variables (More on Wednesday)

- An assignment or binding of names to values/objects
- What can have variables?: scripts, functions, loops, etc.
- Two main types of variables
  - Global
    - Can access anywhere
  - Local
    - Can only access where in the place where it is defined

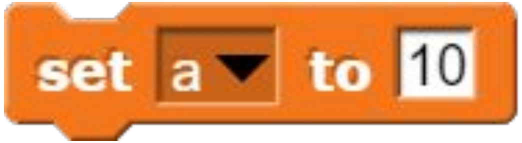




```
script variables (a)
set a to 0
```

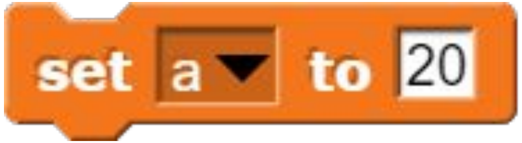
Variables

- I can assign variables AND re-assign variables
- If I say a = 10



```
set a to 10
```

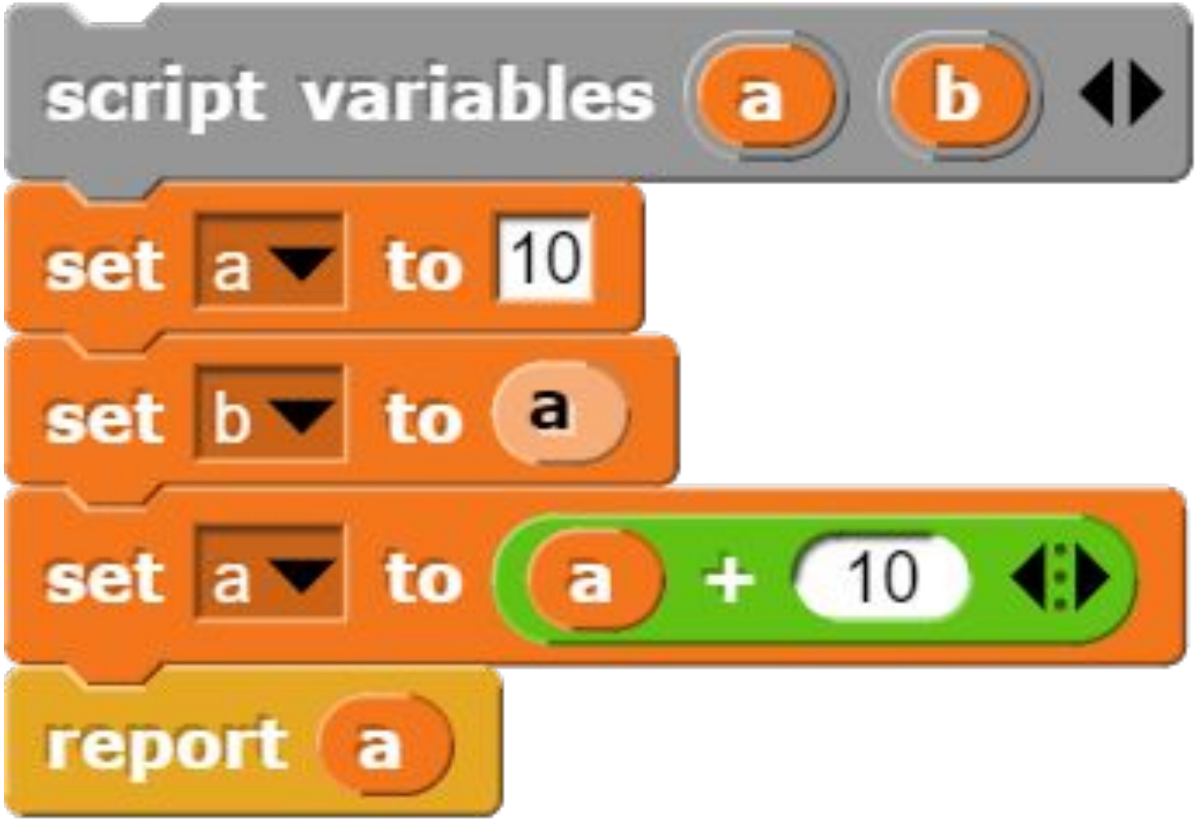
- Then, I say a = 20



```
set a to 20
```

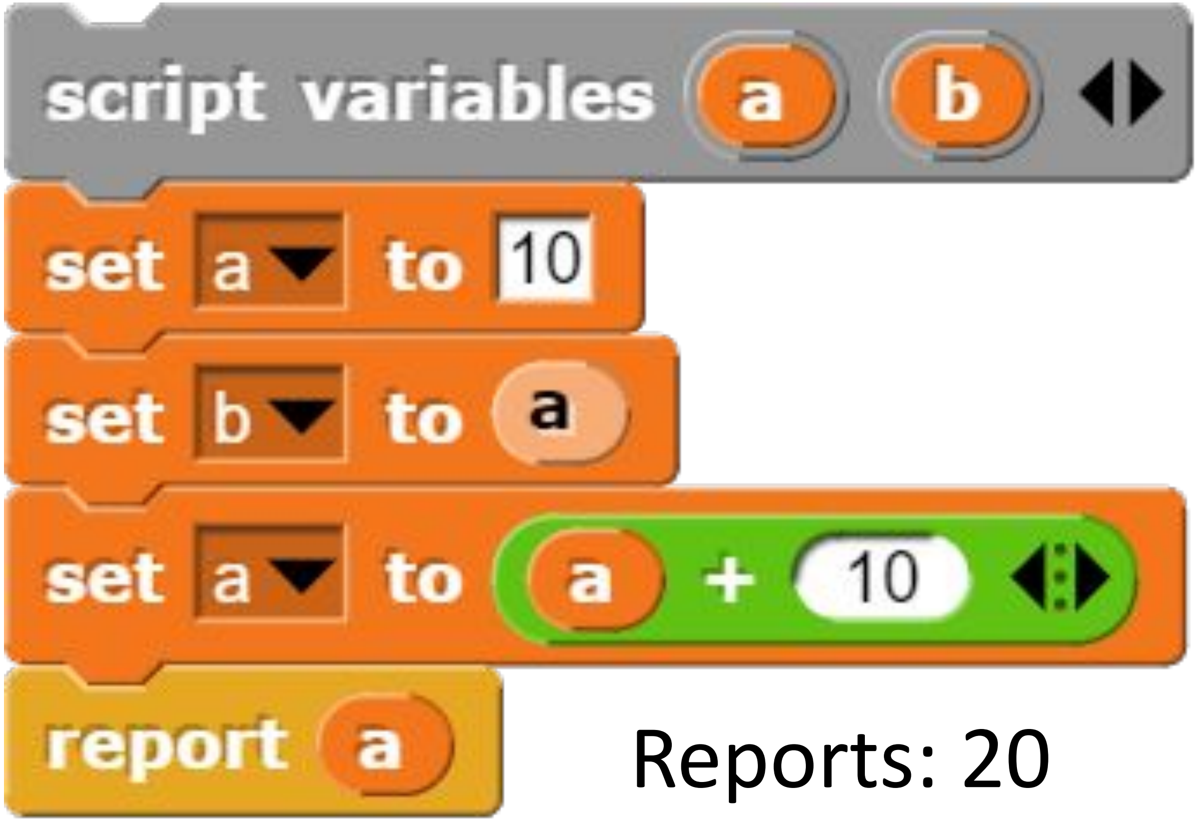
- The value of 20 did not change, **a** was just re-assigned to a new value

Guess that Value! - Practice at home



```
script variables (a) (b)
set a to 10
set b to (a)
set a to ((a) + 10)
report (a)
```

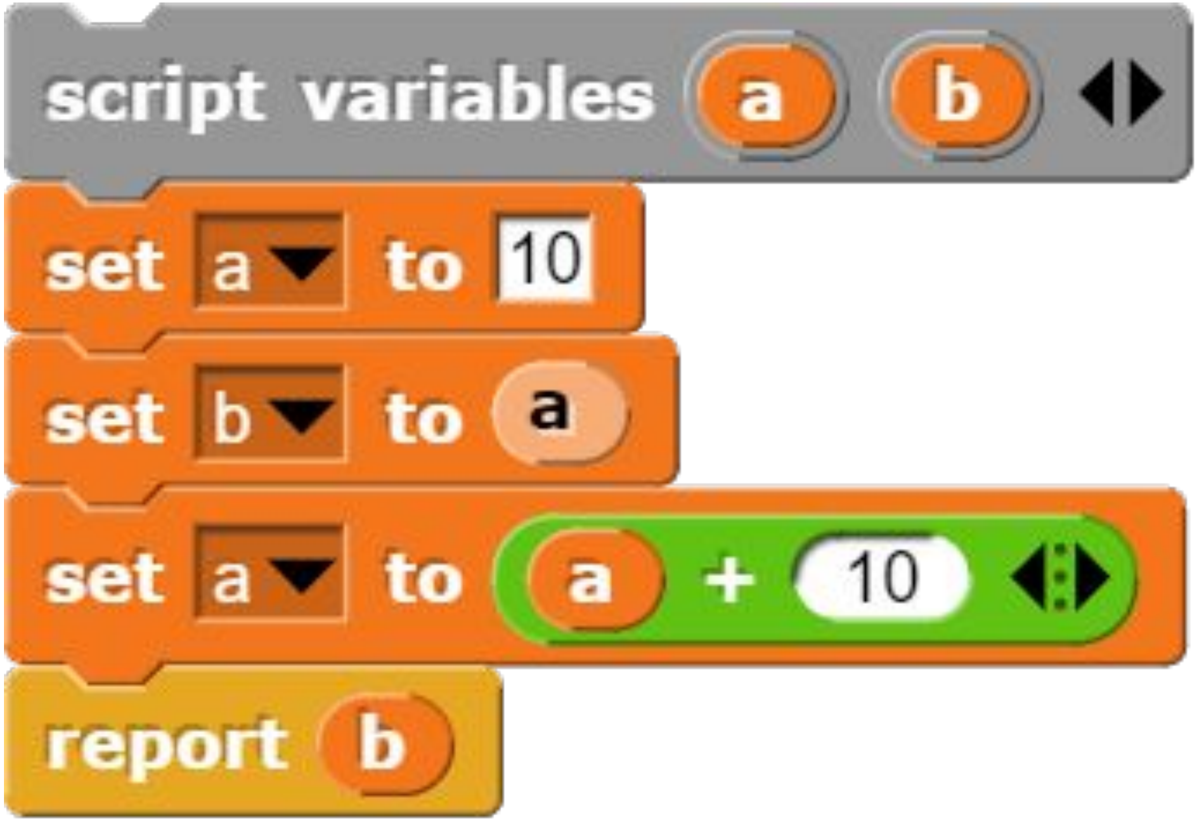
Guess that Value! - Practice at home



```
script variables (a) (b)
set a to 10
set b to (a)
set a to ((a) + 10)
report (a)
```

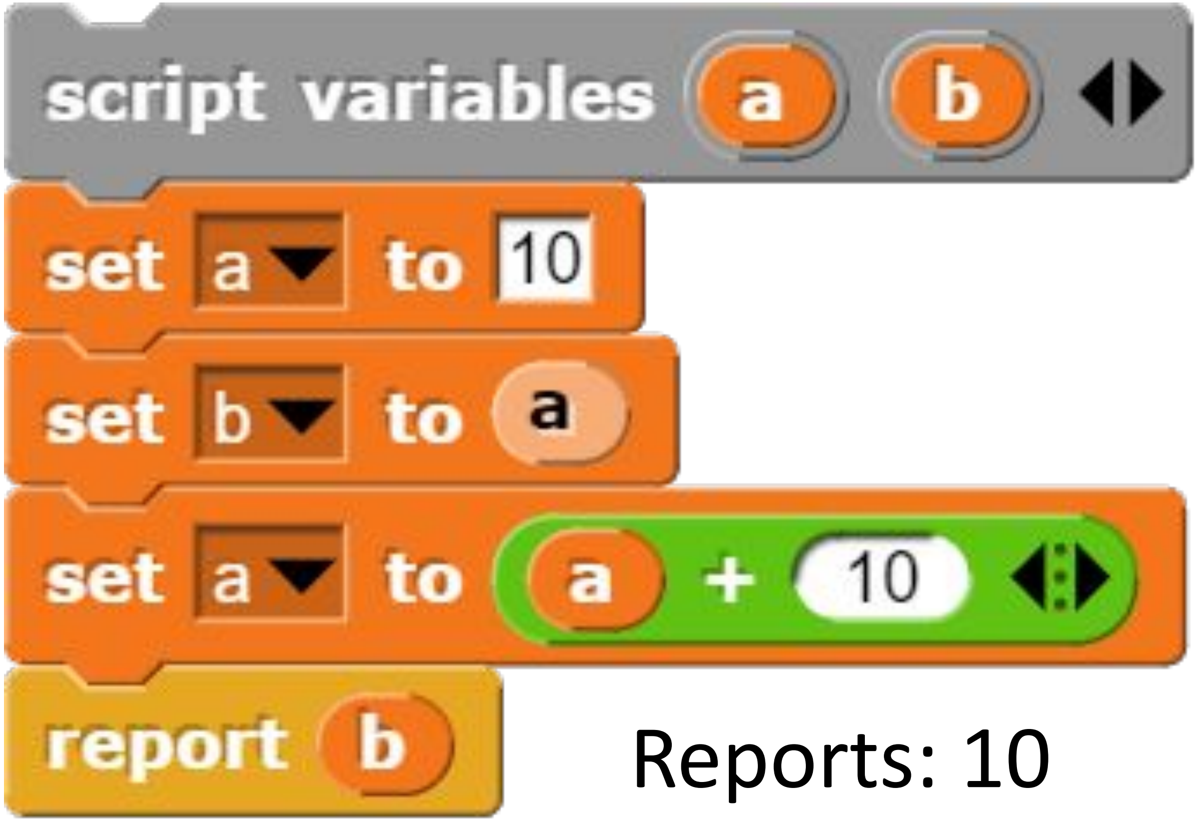
Reports: 20

Guess that Value! - Practice at home



```
script variables (a) (b) ◂▸
set a ▾ to 10
set b ▾ to (a)
set a ▾ to ((a) + (10) ◂▸)
report (b)
```

Guess that Value! - Practice at home

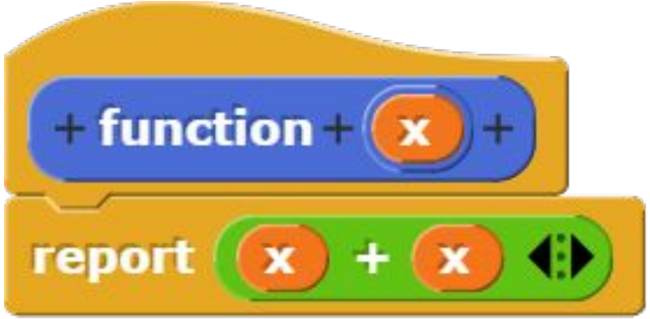


```
script variables a b
set a to 10
set b to a
set a to a + 10
report b
```

Reports: 10

Functions + Inputs

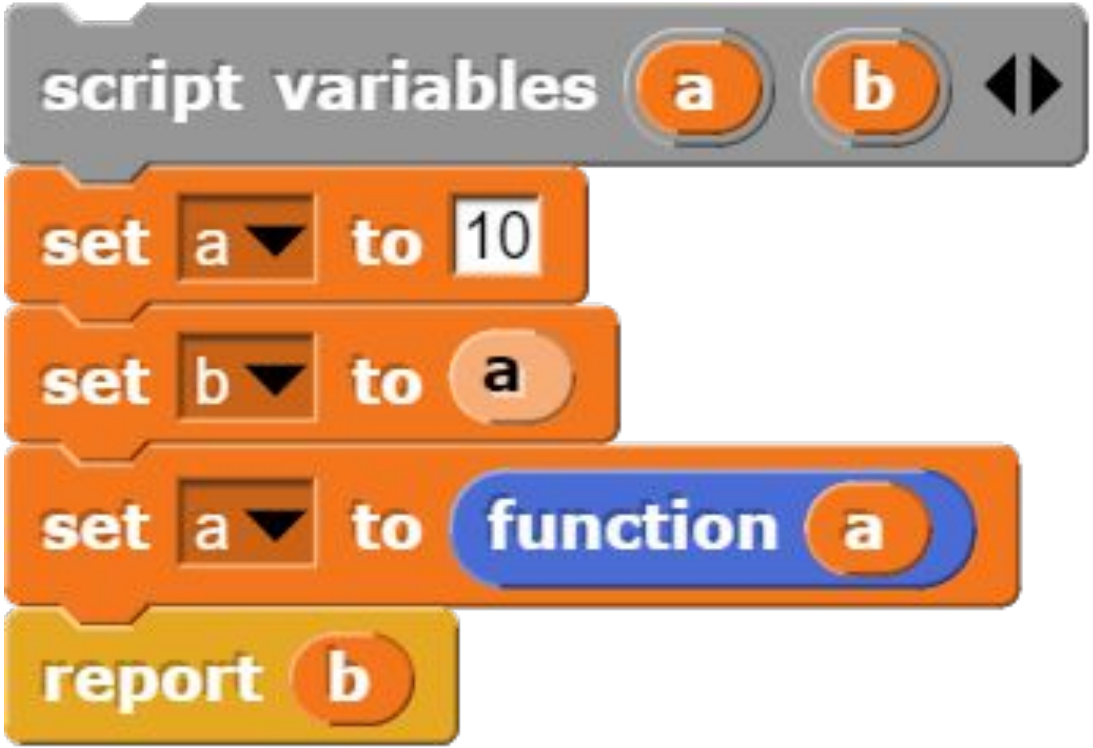
- Functions can have local variables.
- The input variable gets bound to the input value when the function is called



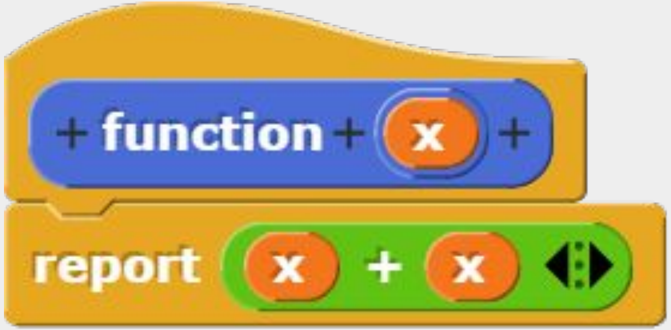
```
+ function + (x) +
report ((x) + (x))
```

- Once function 5 is called, x gets assigned to 5. The x ONLY exists inside of the function. X is local to function

Guess that Value! - Practice at home

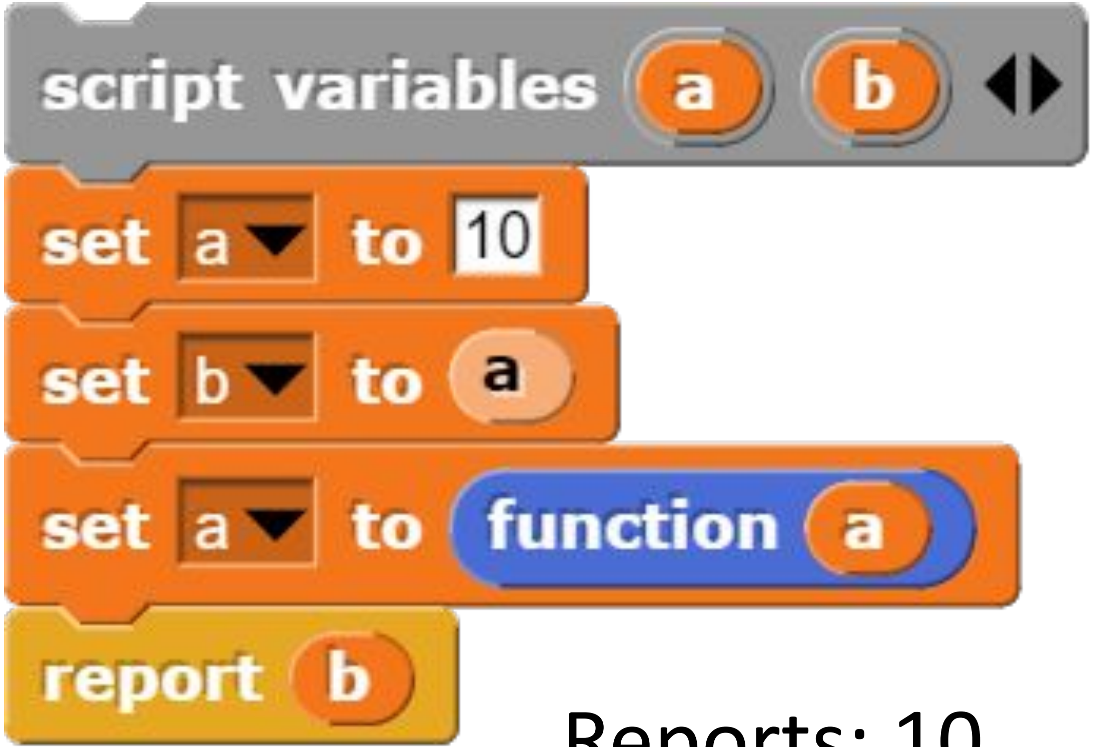


```
script variables (a) (b) ◀▶
set a ▼ to 10
set b ▼ to (a)
set a ▼ to (function (a))
report (b)
```



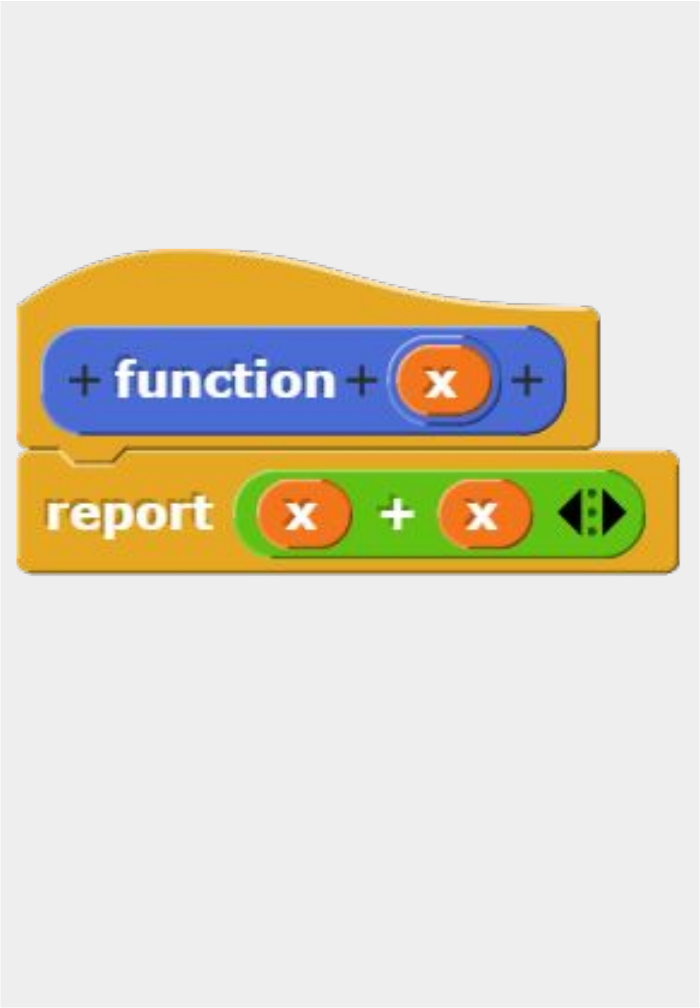
```
+ function + (x) +
report ((x) + (x) ◀▶)
```

Guess that Value! - Practice at home



```
script variables (a) (b) ◀▶
set a ▼ to 10
set b ▼ to (a)
set a ▼ to (function (a))
report (b)
```

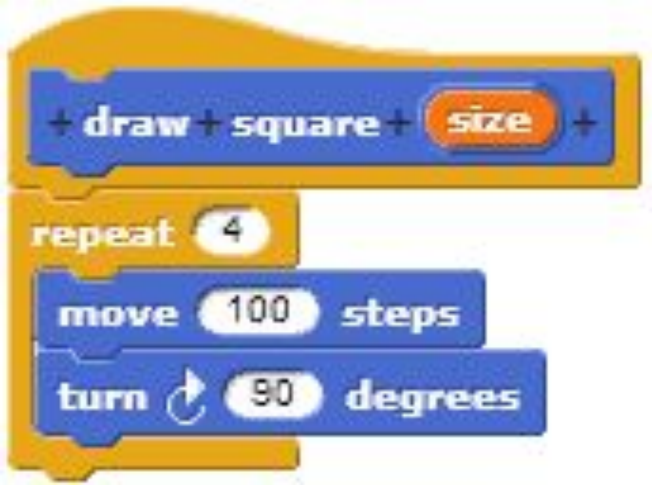
Reports: 10



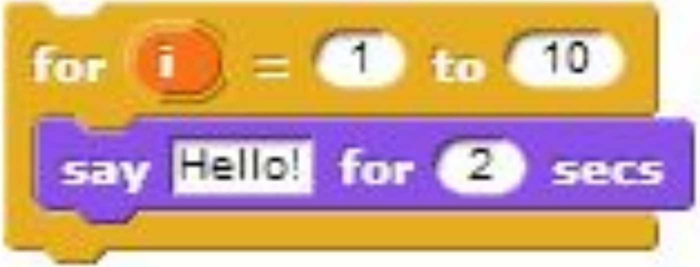
```
+ function + (x) +
report ((x) + (x) ◀▶)
```

Iteration

- Iteration is a set of instructions that are executed repeatedly a certain number of times (or until a condition is met).
- In lab, we've seen



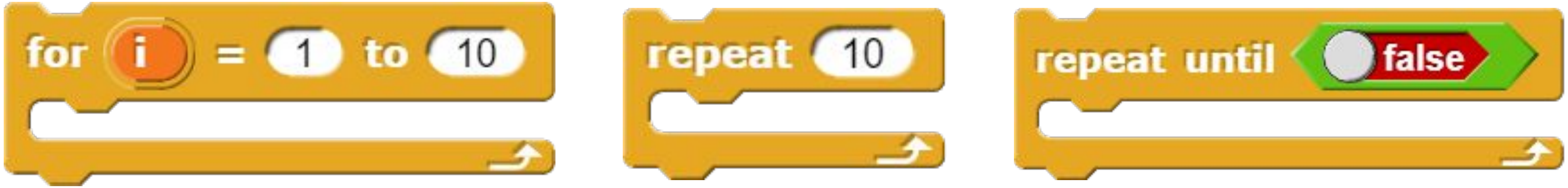
```
+ draw + square + (size) +
repeat (4)
    move (100) steps
    turn ↻ (90) degrees
```



```
for (i) = (1) to (10)
    say [Hello!] for (2) secs
```

Iteration

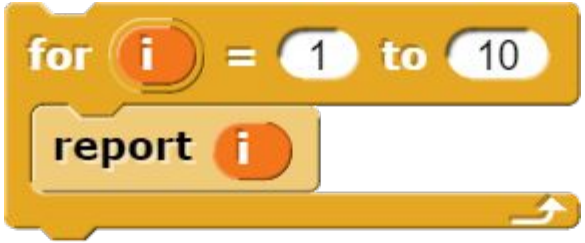
- Here are some different iterative statements in Snap*!*



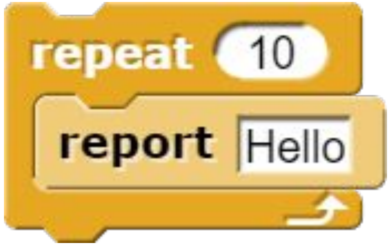
- All three of these iterative statements…
  - ○ Will run everything inside of the loop and repeat it
  - ○ *Note: can be terminated early if a "report" block is called inside*

Iteration

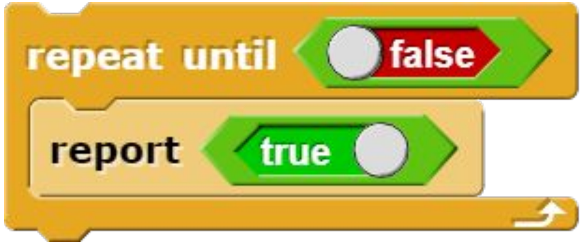
- *Note: can be terminated early if a “report” block is called inside*



```
for i = 1 to 10
    report i
```



```
repeat 10
    report Hello
```

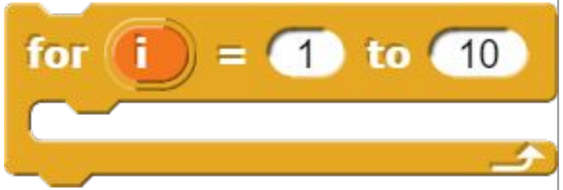
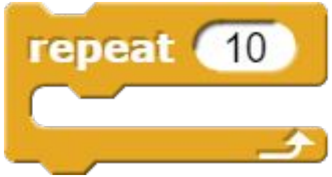
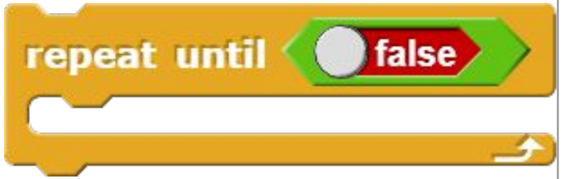


```
repeat until false
    report true
```

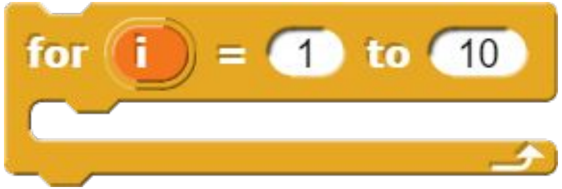
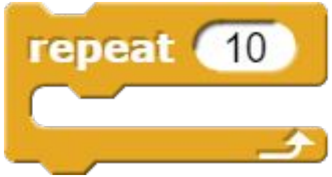
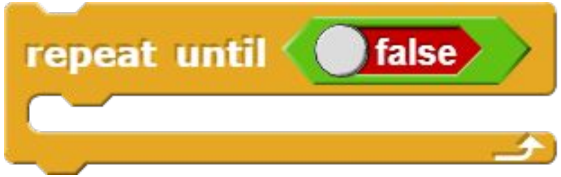
- Once the report is called, the script immediately terminates

Iteration

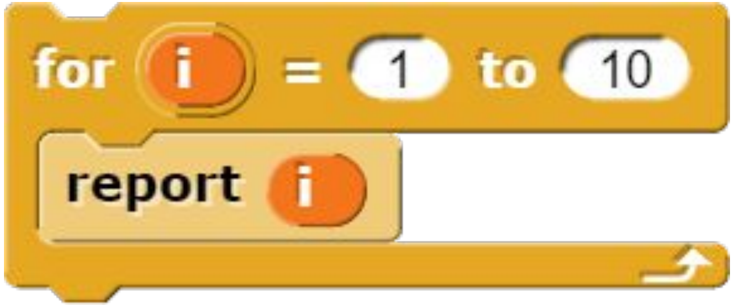
- What are the differences between these?

 for i = 1 to 10	 repeat 10	 repeat until false
•	•	•

Iteration

 for i = 1 to 10	 repeat 10	 repeat until false
● Iterates over a range of values ● “i” is a variable that increases after code inside is done running ● Loop is finished once it’s done running the last number	● Repeats the instructions a set number of times.	● Continues iterating UNTIL the condition is true. ● The condition must be a boolean ● Can use a variable (like “i”) but has to be initialized outside of loop

For i loop



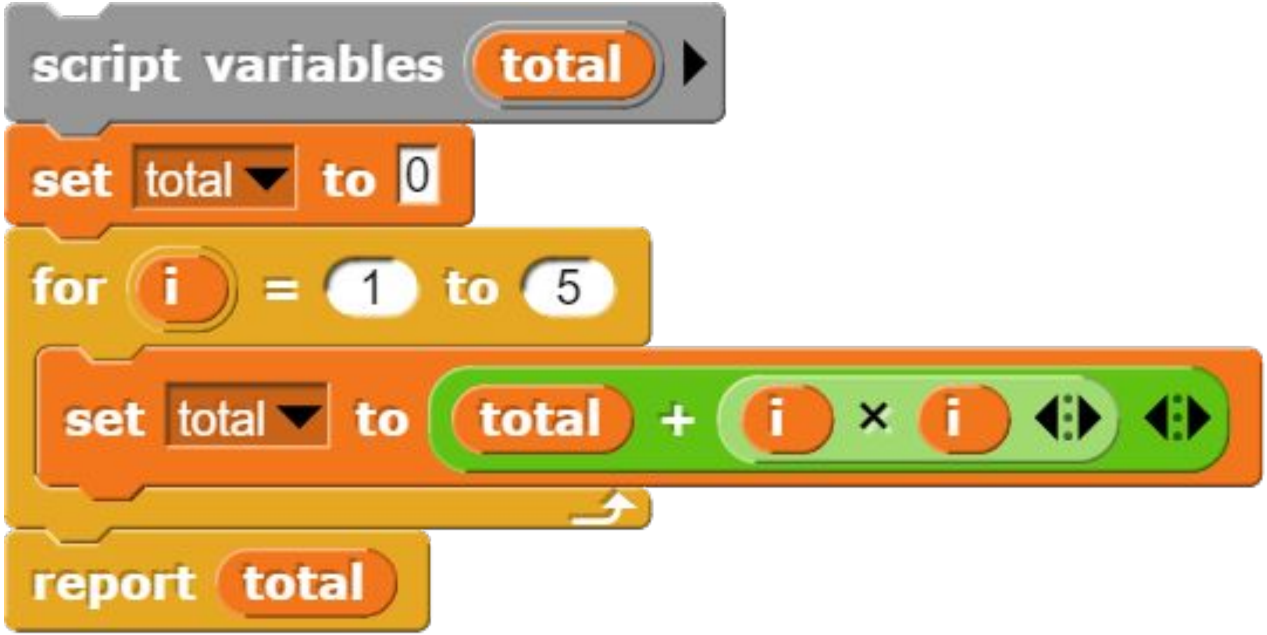
```
for i = 1 to 10
    report i
```

- Structure
    - for i = <num1> to <num2>
        - expressions
- The process
    - ‘i’
        - is a variable that only exists inside of the for loop
    - Assigns “i” to num1 at start
    - Do all steps inside the block
    - Increment i
    - Repeat until i = 10 and it is done with all steps at i = 10

For `i` loop: Example 1

- Problem: I want to find the square of each number from 1 to 5 and then sum them all together.
- Things to ask ourselves
  - What functions / blocks do we need?
  - What should be outputted?

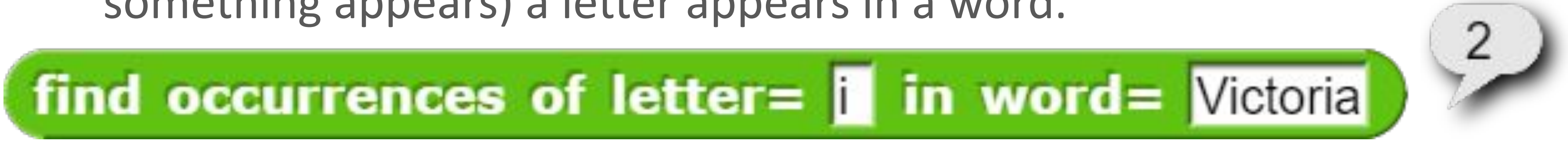
For i loop: Example 1

- Problem: I want to find the square of each number from 1 to 5 and then sum them all together.
- 

```
script variables (total)
set total to 0
for (i) = 1 to 5
    set total to ((total) + ((i) × (i)))
report (total)
```

For i loop: Example 2

- Problem: I want to find the number of occurrences (number of times something appears) a letter appears in a word.



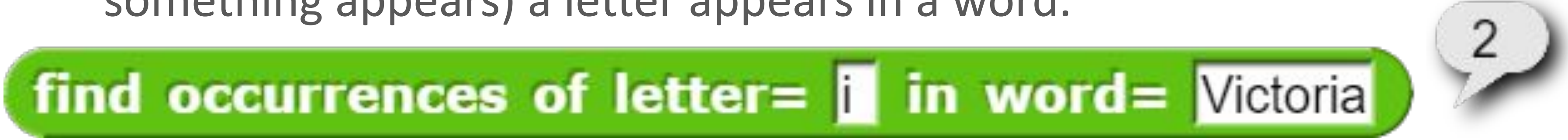
find occurrences of letter= i in word= Victoria

2

- What is the domain?:
- What is the range?:
- What kind of blocks will be need?:

For i loop: Example 2

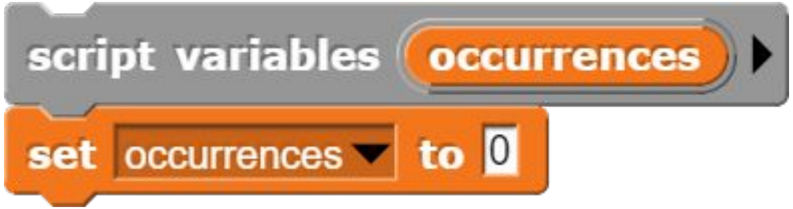
- Problem: I want to find the number of occurrences (number of times something appears) a letter appears in a word.



find occurrences of letter= i in word= Victoria

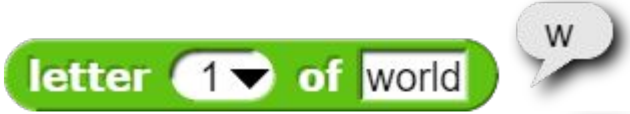
2

- What is the domain?: letters for both inputs!
- What is the range?: A number
- What kind of blocks will be need?:



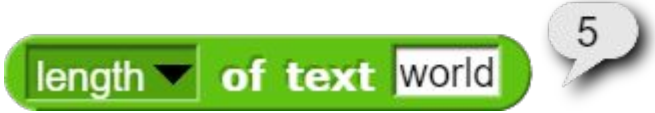
script variables occurrences

set occurrences to 0



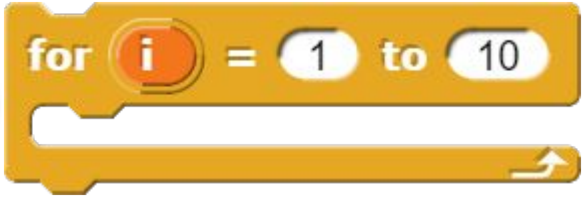
letter 1 of world

w

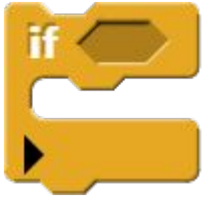


length of text world

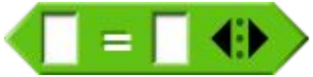
5



for i = 1 to 10

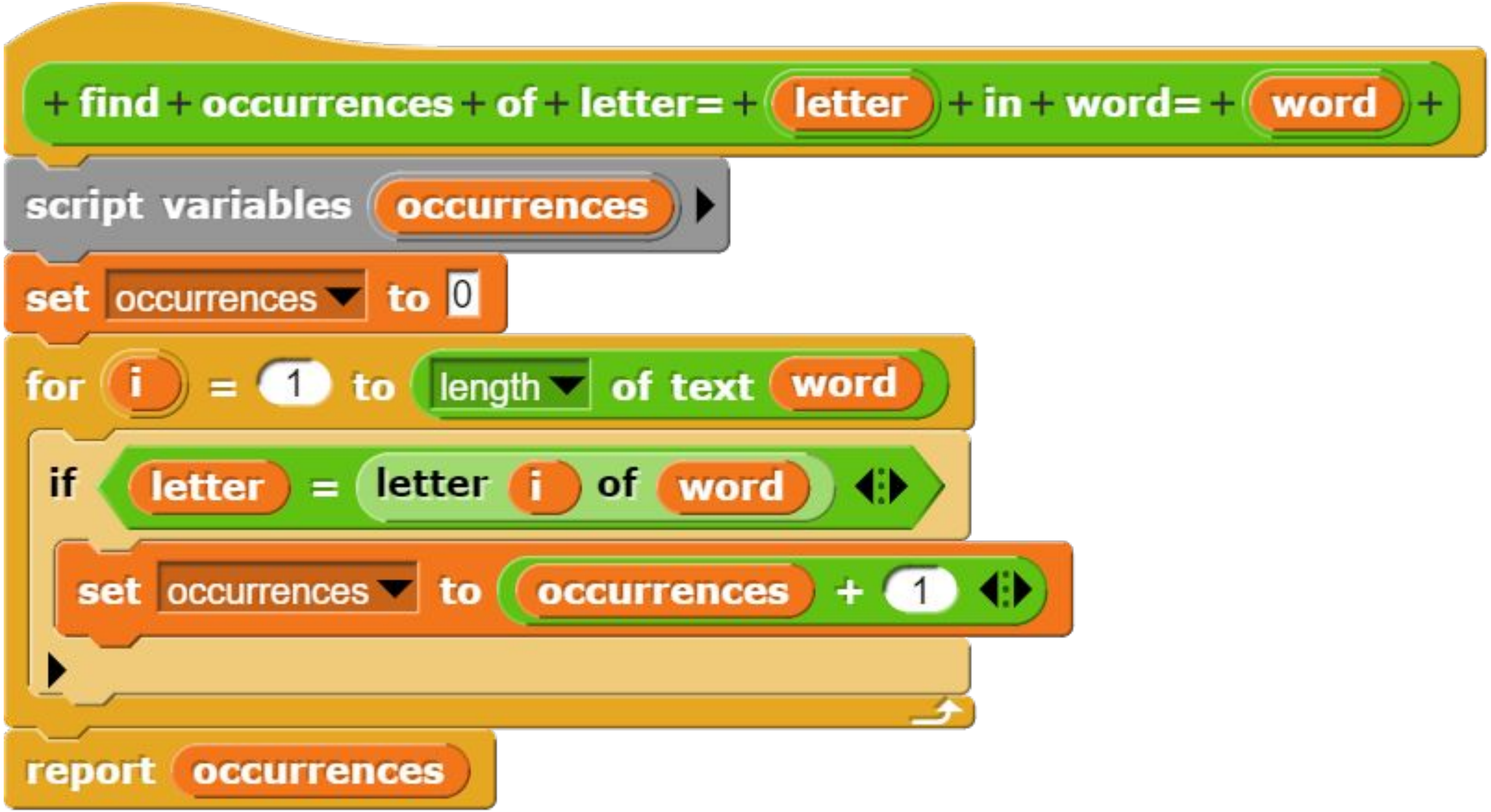


if



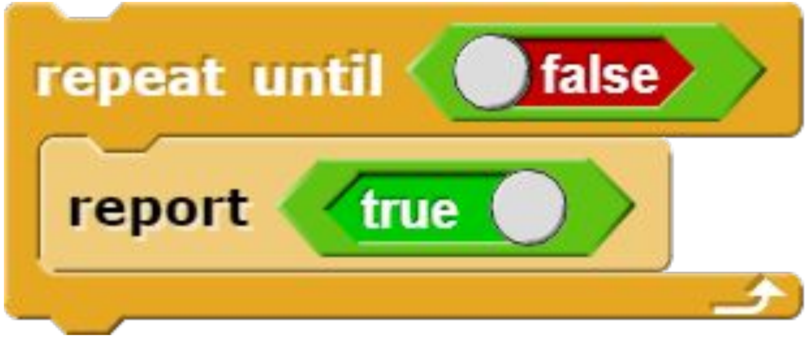
□ = □

For i loop: Example 2



```
+ find + occurrences + of + letter= + letter + in + word= + word +
script variables occurrences ▸
set occurrences ▾ to 0
for i = 1 to length ▾ of text word
    if letter = letter i of word ◂▸
        set occurrences ▾ to occurrences + 1 ◂▸
    ▸
report occurrences
```

Repeat until



- Structure
    - Repeat until <condition>
        - expression
- The process
    - <condition> evaluates to a Boolean
    - if <condition> is False, do all the steps inside the block
    - check <condition> again
    - repeat until <condition> is True
- Runs possibility of forever loop (bug) if the <condition> never evaluates to true

Repeat until: Example 1

- Problem: Same question as for i loop: Example 1 – . “I want to find the square of each number from 1 to 5 and then sum them all together.”

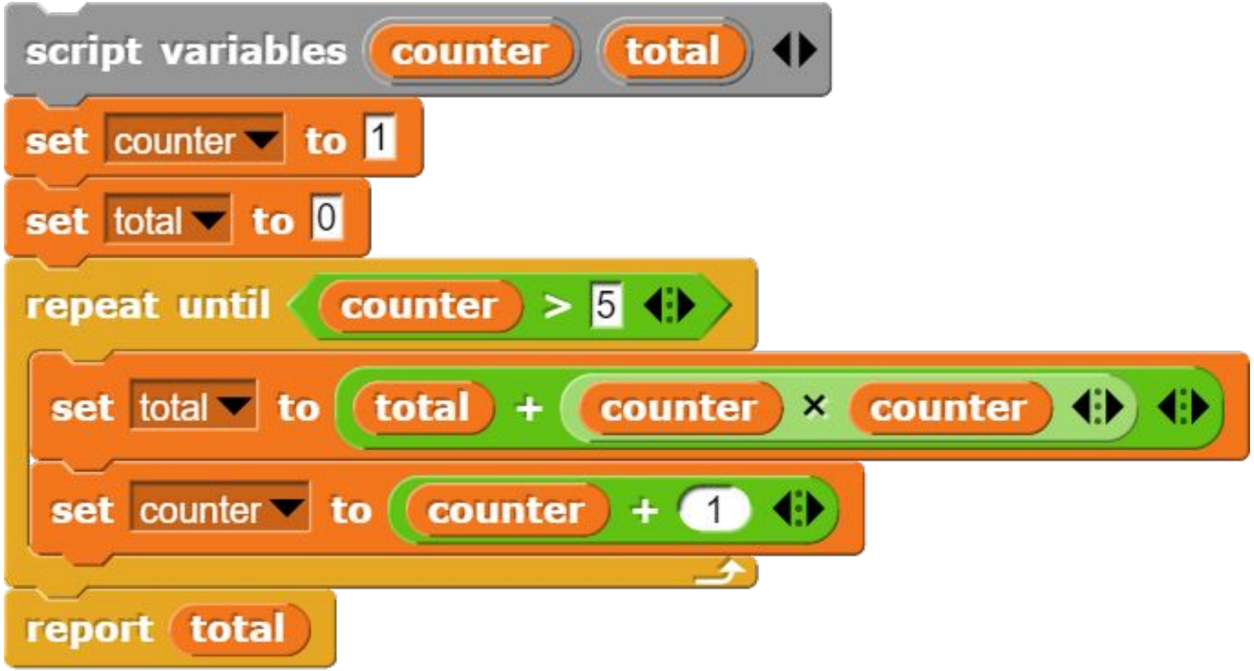
Repeat until: Example 1

- Problem: Same question as for i loop: Example 1 – “I want to find the square of each number from 1 to 5 and then sum them all together.”
- What do I need?
    - Don’t have an i to increment so I will need a variable to keep track. Let’s call it ‘counter’
    - I will need an additional variable for the total sum
    - **When should I end the statement?**

Repeat until: Example 1

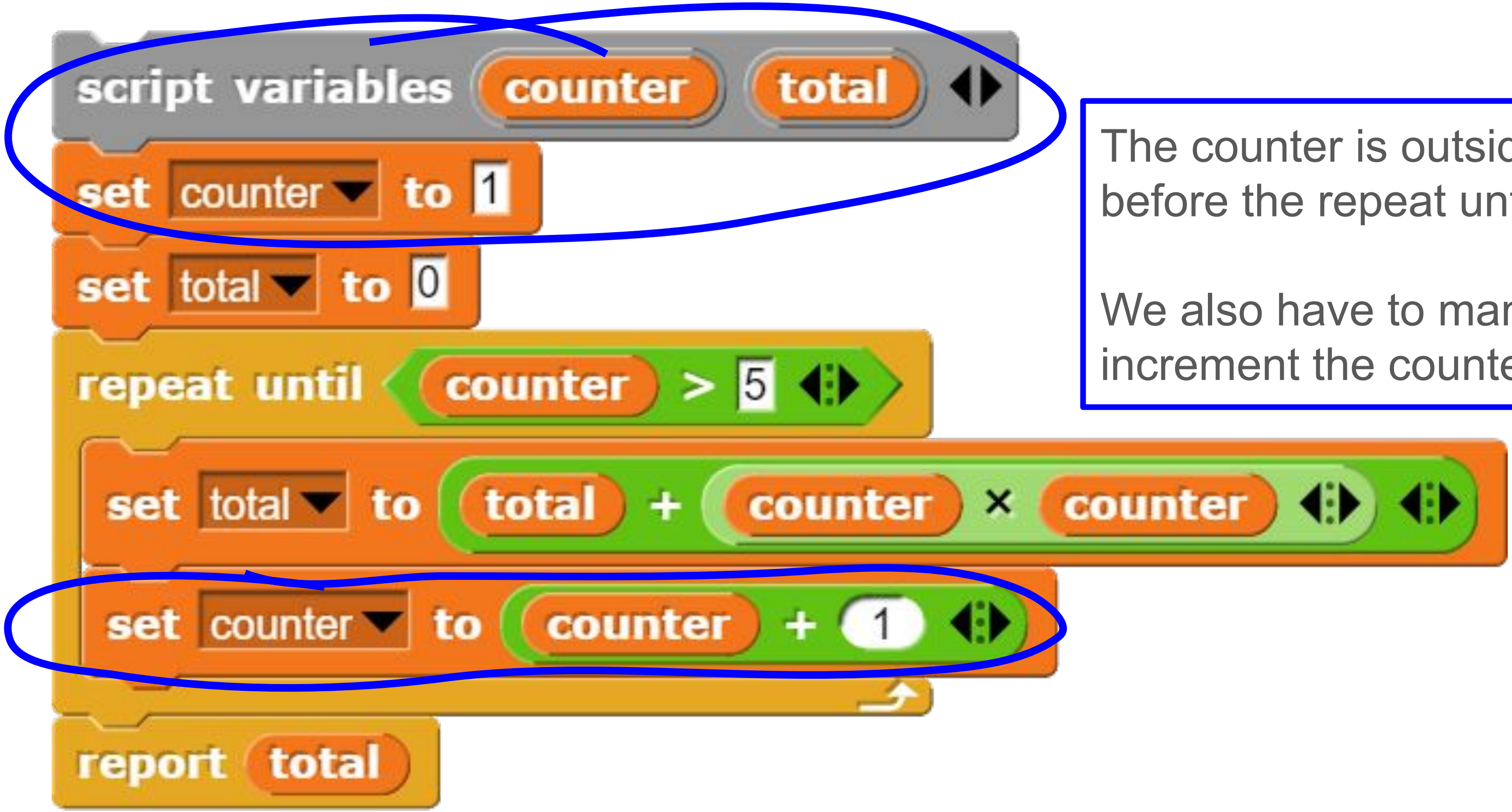
- Problem: Same problem as for i loop: Example 1 – “I want to find the square of each number from 1 to 5 and then sum them all together.”
- What do I need?
    - Don’t have an i to increment so I will need a variable to keep track. Let’s call it ‘counter’
    - I will need an additional variable for the total sum
    - **When should I end the statement?**
        - Once I’m done running the 5. So I can end when counter > 5

Repeat until: Example 1

- Problem: Same question as for i loop: Example 1 – “I want to find the square of each number from 1 to 5 and then sum them all together.”
- 

```
script variables counter total
set counter to 1
set total to 0
repeat until counter > 5
    set total to total + counter × counter
    set counter to counter + 1
report total
```

Repeat until: Example 1



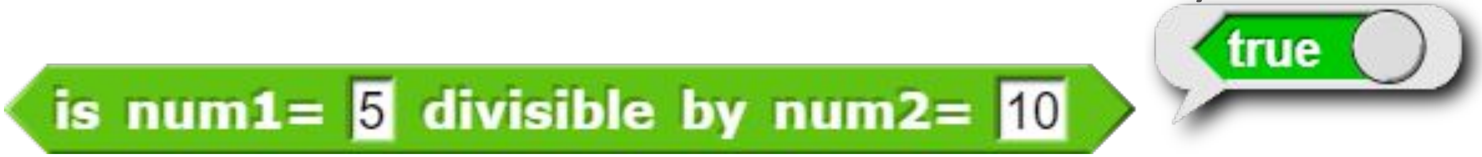
```
script variables counter total
set counter to 1
set total to 0
repeat until counter > 5
    set total to total + counter × counter
    set counter to counter + 1
report total
```

The counter is outside and before the repeat until loop

We also have to manually increment the counter

Repeat until: Example 2

- Problem: I want to find if a number is divisible by another number



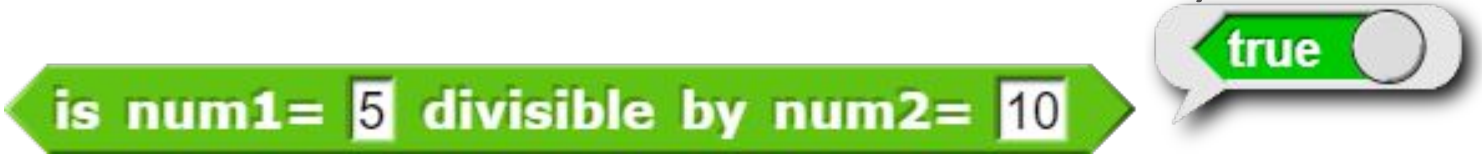
is num1= 5 divisible by num2= 10

true

- What is the Domain?:
- What is the Range?:
- What kinds of blocks will we need?:

Repeat until: Example 2

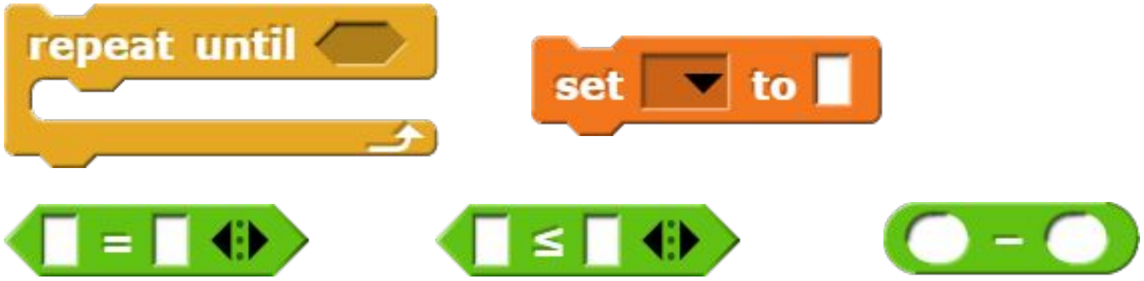
- Problem: I want to find if a number is divisible by another number



is num1= 5 divisible by num2= 10

true

- What is the Domain?: Both are numbers!
- What is the Range?: A boolean
- What kinds of blocks will we need?:



repeat until

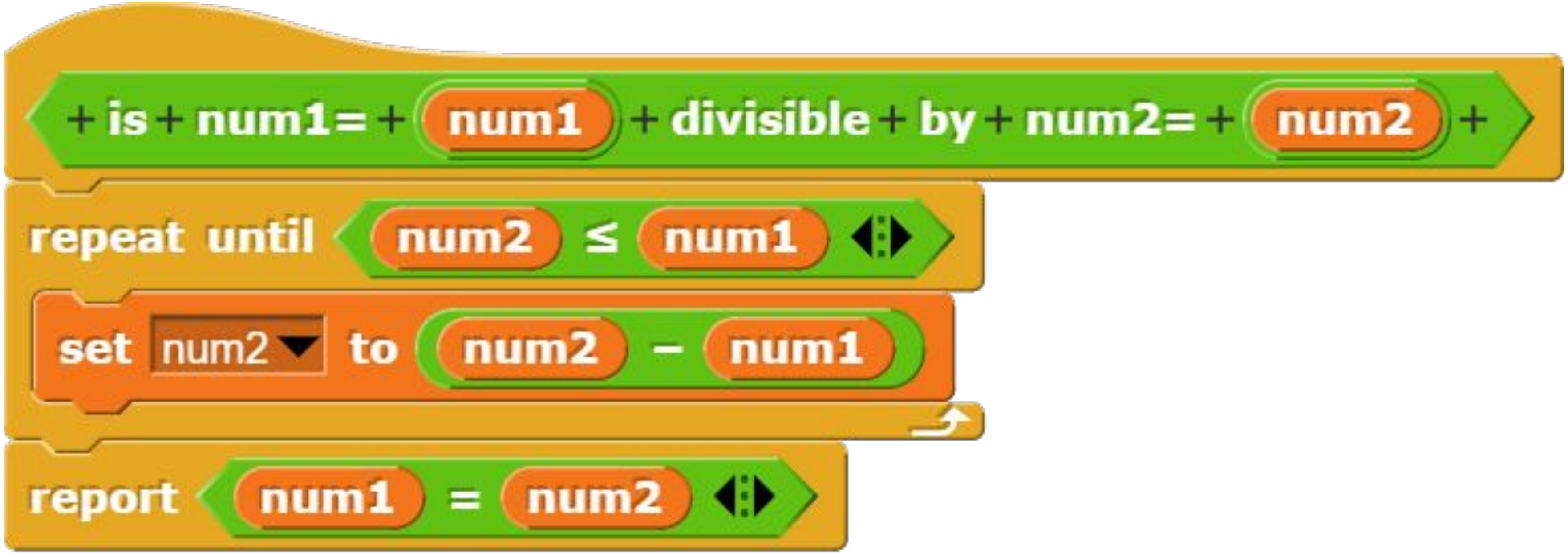
set ▾ to

=

≤

−

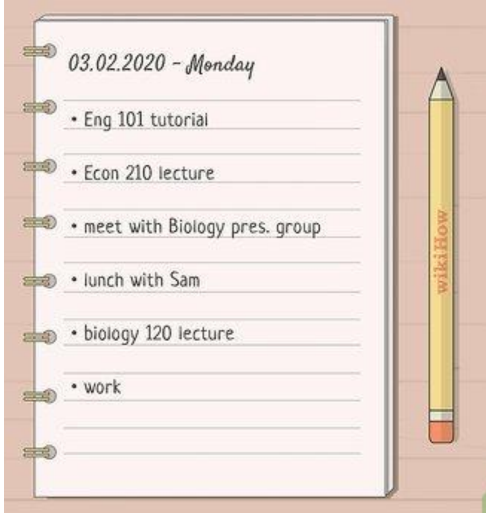
Repeat until: Example 2



```
+ is + num1= + (num1) + divisible + by + num2= + (num2) +
repeat until < (num2) ≤ (num1) >
    set num2 to ( (num2) − (num1) )
report < (num1) = (num2) >
```

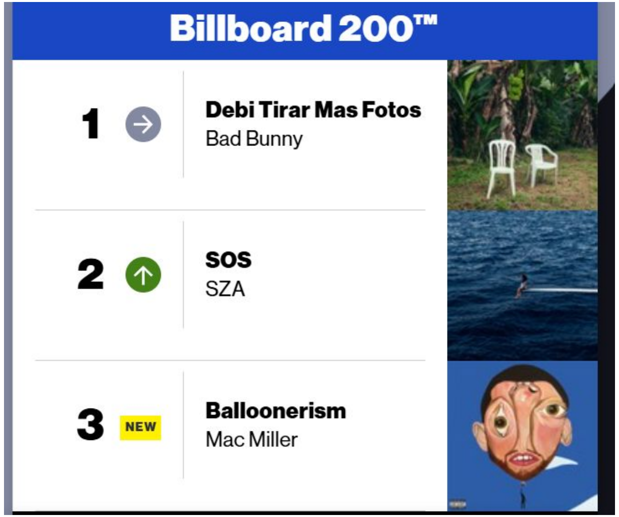
Lists

- Similar to ‘to do list’, ranking, excel sheet



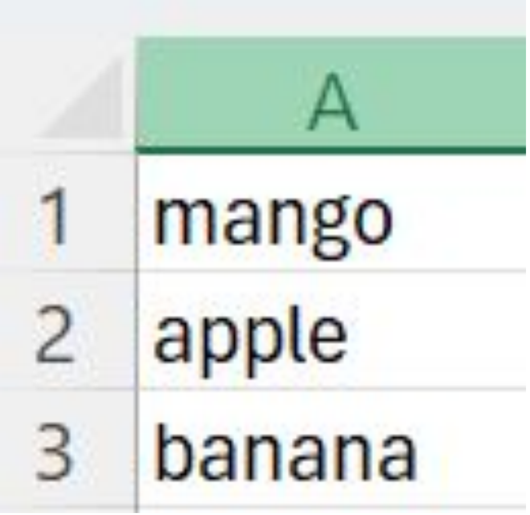
03.02.2020 – Monday

- Eng 101 tutorial
- Econ 210 lecture
- meet with Biology pres. group
- lunch with Sam
- biology 120 lecture
- work



Billboard 200™

1	→	**Debi Tirar Mas Fotos** Bad Bunny
2	↑	**SOS** SZA
3	NEW	**Balloonerism** Mac Miller

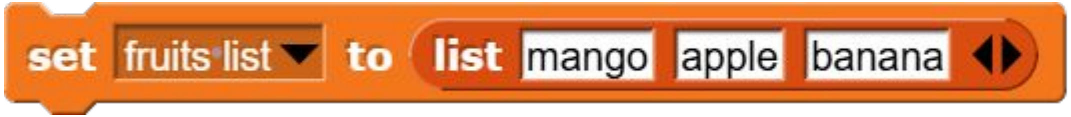


	A
1	mango
2	apple
3	banana

Lists

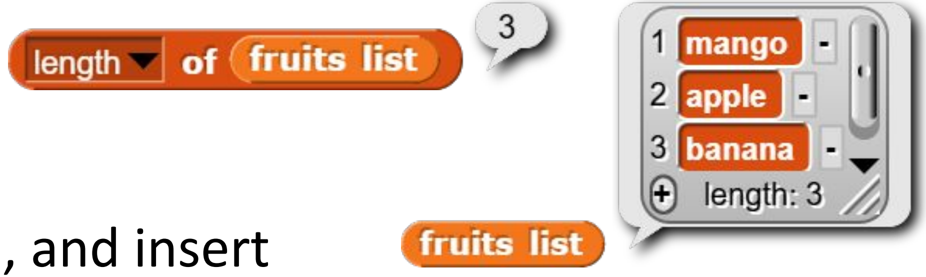
- Data type that holds individual data types
    - Text, Numbers, Booleans, etc
    - Lists of Lists (2D lists) --> future lecture
- Ordered collections of values
- Each item as a location/address in a list - called the “index”

Lists Example

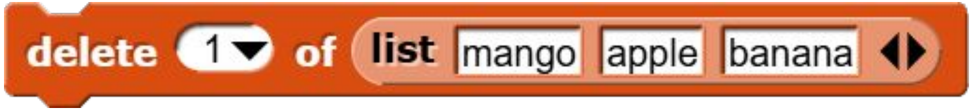


```
set fruits list to list mango apple banana
```

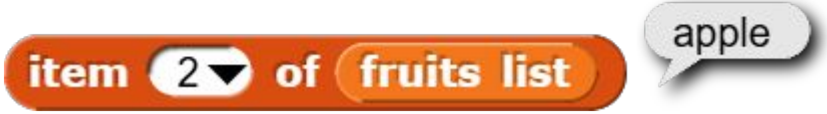
- Fruit list =
  - [‘mango’, ‘apple’, ‘banana’]
- Lists are indexable
- You can also add, delete, replace, and insert elements in a list



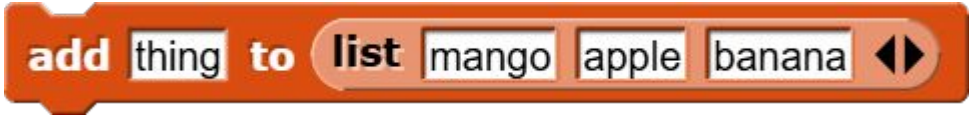
```
length of fruits list
3

fruits list
1 mango
2 apple
3 banana
length: 3
```



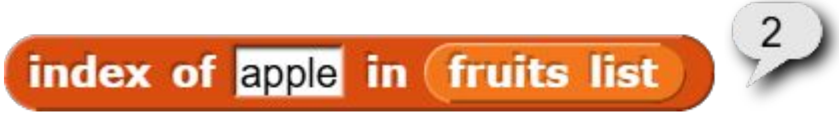
```
delete 1 of list mango apple banana
```



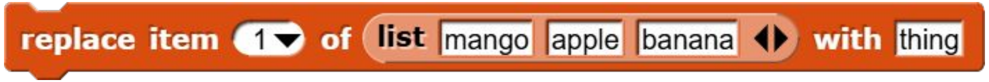
```
item 2 of fruits list
apple
```



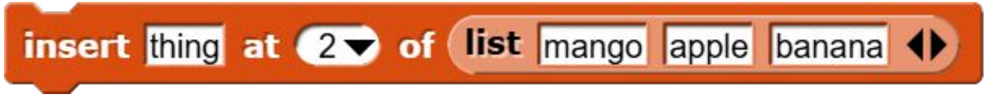
```
add thing to list mango apple banana
```



```
index of apple in fruits list
2
```

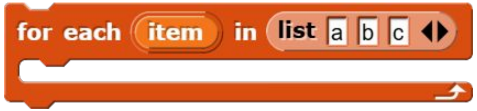
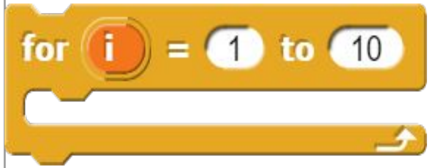
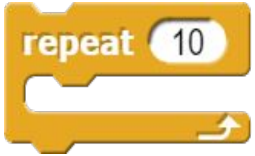
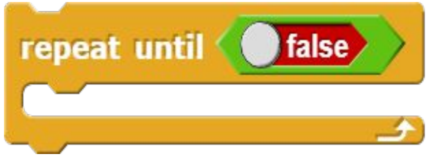


```
replace item 1 of list mango apple banana with thing
```

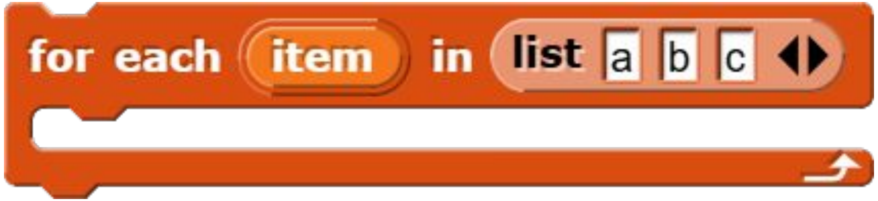


```
insert thing at 2 of list mango apple banana
```

Iteration

 for each item in list a b c	 for i = 1 to 10	 repeat 10	 repeat until false
● Iterates over EACH ELEMENT in the list ● Item will get set to ‘a’, run the code inside the loop, then item will be set to ‘b’, run code, then ‘c’. Then, it will stop	● Iterates over a range of values ● “i” is a variable that increases after code inside is done running ● Loop is finished once it’s done running the last number	● Repeats the instructions a set number of times.	● Continues iterating UNTIL the condition is true. ● The condition must be a boolean

For each **item** loop

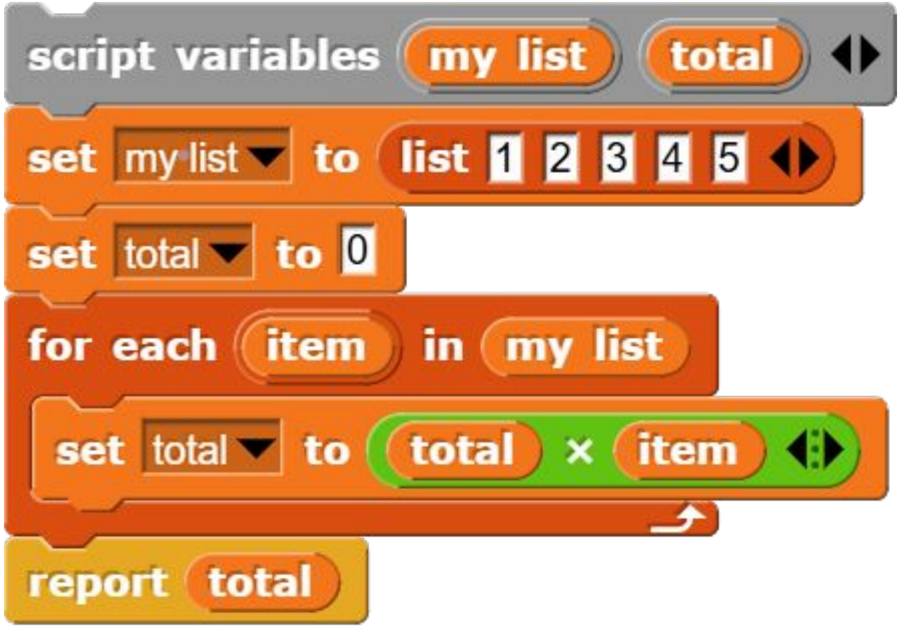


```
for each (item) in (list a b c ◀▶)
```

- Structure
  - for each item in <list>
    - expressions
- The process
  - item:
    - is a variable that only exists inside of the for each loop
  - Assigns “item” to the first element in the list
  - Do all steps inside the block
  - Item gets reassigned to the next element
  - Repeat until item has been assigned to all items in the list

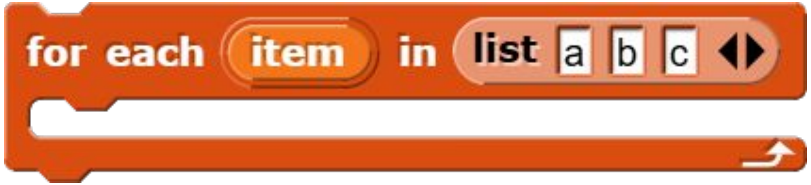
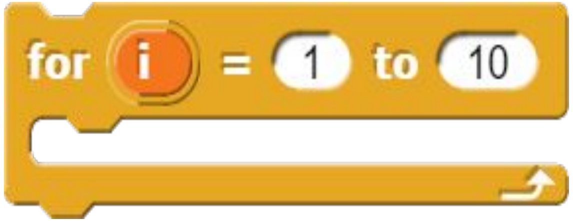
Lists – For each item loop: Example

- Problem: I want to find the square of each number in my list and then sum them all together.



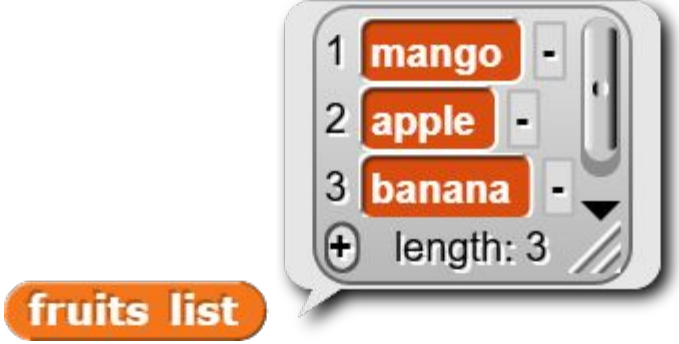
```
script variables (my list) (total)
set my list to (list 1 2 3 4 5)
set total to 0
for each (item) in (my list)
    set total to ((total) × (item))
report (total)
```

For each item vs For i

 for each (item) in list a b c	 for (i) = 1 to 10
● Input data type = lists ● Item can be any data type (the value is whatever the list contains) ● Can be used to access each element in list	● Input data type = numbers ● 'i' will always be a number within the range of numbers ● Can be used to access the INDEX of a list ● It can also be used to just work with numbers ● It can also be used to access the index of text (strings)

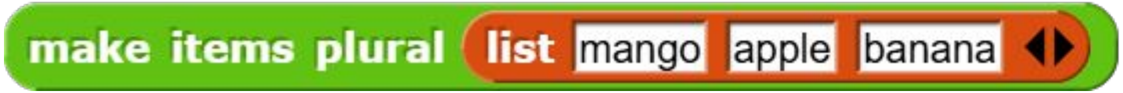
Lists – For i loop: Example

- Problem: I want to take a list of strings, and make it plural by adding an ‘s’ to the end. **<u>Use the ‘for i’ loop.</u>**
- *Hint: Replace would be helpful!*
- What is the domain?:
- What is the range?:
- What kind of blocks will be need?:

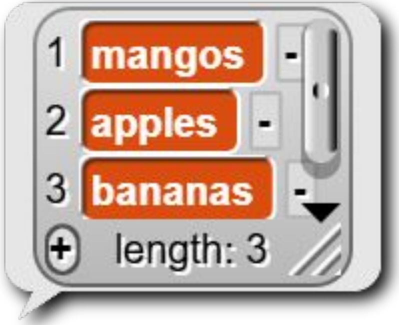


fruits list

1 mango
2 apple
3 banana
length: 3



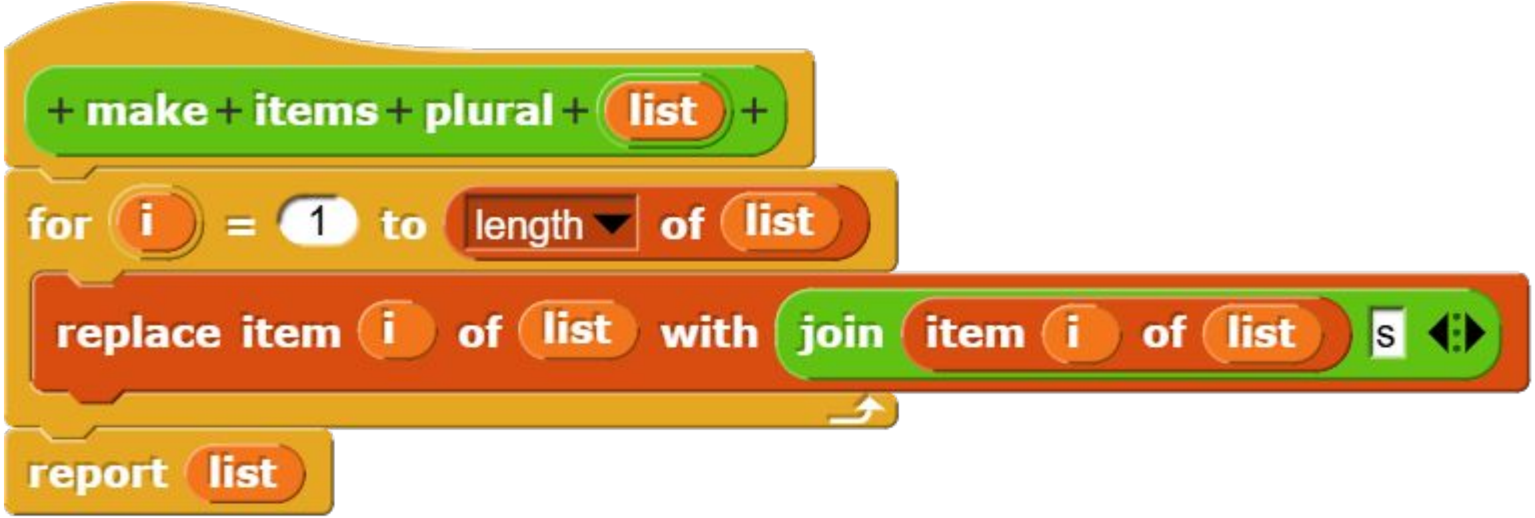
make items plural (list mango apple banana)



1 mangos
2 apples
3 bananas
length: 3

Lists – For i loop: Example

- Problem: I want to take a list of strings, and make it plural by adding an ‘s’ to the end. Use the ‘for i loop’. *Hint: Replace would be helpful!*



```
+ make + items + plural + (list) +
for (i) = (1) to (length of (list))
    replace item (i) of (list) with (join (item (i) of (list)) (s))
report (list)
```

Summary

- **Loops Repeat Actions** – Instead of writing the same code multiple times, loops let the computer do it for you
- **Loop Control** – Check that your loop starts and stops when it should!
- **Use loops to control your program**
- **Lists** - data structures that contain values, ordered, indexable. Iteration can be performed on them.
- **For each item** – get the element in a list
- **For i** - can get the index of a list, text, or just work with numbers, etc.