Programming

or What's Happening to My Brain?

How to Read a Program

```
Thinking... for 2 secs
set index ▼ to 1
set sum ▼ to 0
repeat length of data
 change sum v by item index of data
 change index ▼ by 1
set sum ▼ to
             sum / length of data
   I've got the answer! for 2 secs
```

Can always be read sequentially; each block starts when the previous one finishes*.

One exception: if several things are happening in parallel.

^{*} BYOB provides a couple of blocks that this may not seem true for, such as broadcast and play sound. You should still think of these as sequential operations that can take care of most of their work while the remainder of your code runs.

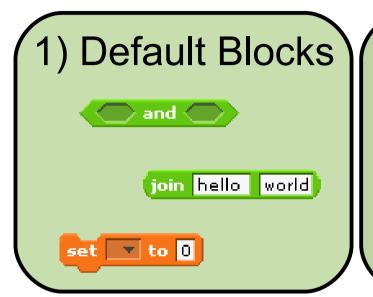
Programming: General Tips

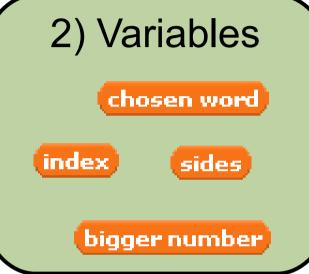
I'd recommend that you think about abstract "algorithms" before code. *Think about how you solve the problem in your head.*

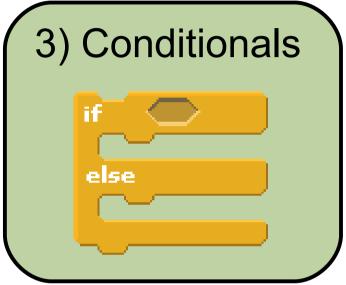
Drawing and writing on paper is not out of style.

This class (and programming in general) is rarely about getting THE right answer. There are often many correct solutions.

Our Toolbox (so far!)







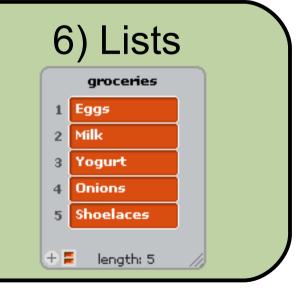
```
4) Loops

repeat 10

repeat until
```

```
5) Your blocks

get largest number in 
to the power of
```



Default Blocks





There are three types of blocks in BYOB:

COMMANDS
REPORTERS
PREDICATES

The best way to learn about the available blocks is to play around with them.

2 Variables



Variable Scope

Global – can be read and written by any script.

Local ("script") – can only be read and written by the current script / block

2 Variables: Example

Let's say you're building an adventure game like Zelda. What are some variables that you might create?

What if you're building Connect 4 or chess?

3 Conditionals

```
if balance > 5000

set interest rate v to 1.1

else

if balance > 1000

set interest rate v to .95

else

set interest rate v to .40
```

Conditionals allow our program to do different things depending on what state it's in.

IF IF / ELSE

MIDPOINT CHALLENGE

Write some code that can validate whether a password is secure (in this case: at least 8 characters long, includes letters and #'s).

check strength of

Note that you will also need a loop for this problem.

4 Loops

```
set count to 0

set letter num to 0

repeat length of word

letter letter num of word = a or

letter letter num of word = e or

letter letter num of word = i or

letter letter num of word = o or

letter letter num of word = u

change count by 1

change letter num by 1
```

Loops let us repeat code a varying number of times. They also lead to more compact code.

REPEAT
REPEAT UNTIL
FOREVER

4 Loops: Example

How would you write the following block?

to the power of

(don't worry about negative exponents)

5 Custom Blocks

get largest number in 🚪



to the power of

Blocks take a certain number of *parameters* (inputs) and produce either zero or one output.

WHY BUILD BLOCKS?

Custom Blocks: Example

How would you build the following block?

get largest number in 📒



6 Lists



Lists give us the ability to store large (and varying) amounts of related data under a single name.

ADD
DELETE
GET ITEM #___
LENGTH

6 Lists: Example

Let's say that we're writing a shopping cart program and have the following lists:

```
      items
      prices

      1 Toothpaste
      1 2.79

      2 Running Shoes
      2 49.99

      3 Mirror
      3 129.89

      4 DVD Player
      4 49.95

      5 Spatula
      5 8.99

      + ■ length: 5
```



How would we write a block to determine the total value of the items in the shopping cart?

FINAL CHALLENGE

How would you create the following block?

remove duplicates from 🖪

Your block should create a new list that should be identical to the input list except that all numbers in the list will be unique.