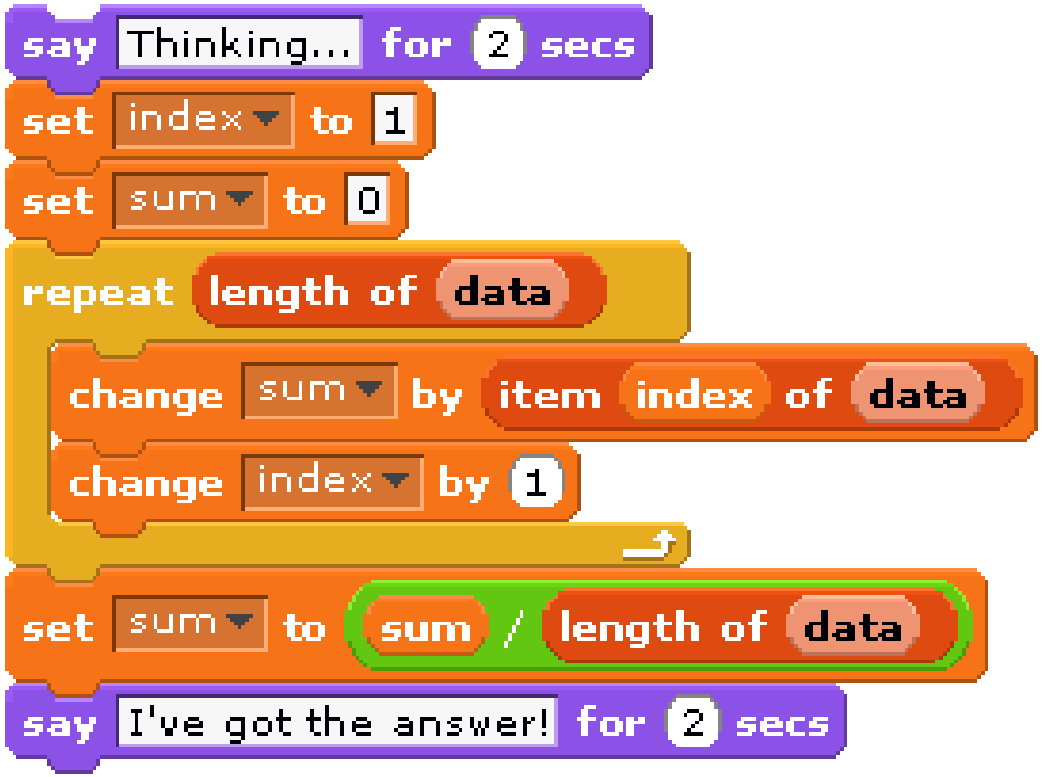
Programming

or

What's Happening to My Brain?

How to Read a Program



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.***

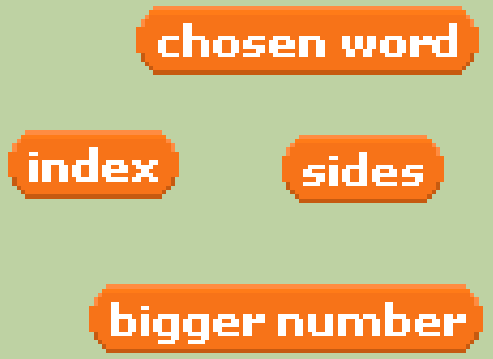
Drawing and writing on paper is *not* out of style.

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

Our Toolbox (so far!)

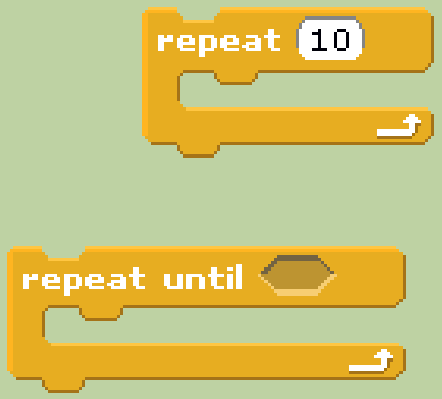
1) Default Blocks

2) Variables

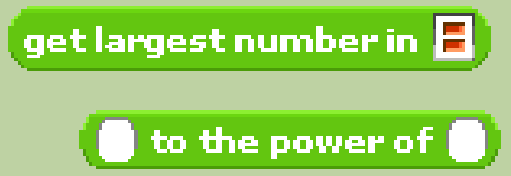


3) Conditionals

4) Loops

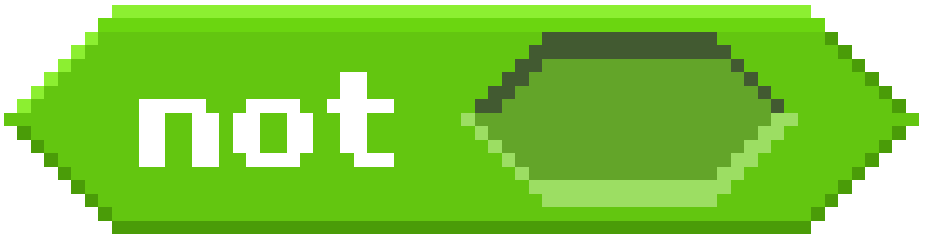


5) Your blocks



6) Lists

1 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.

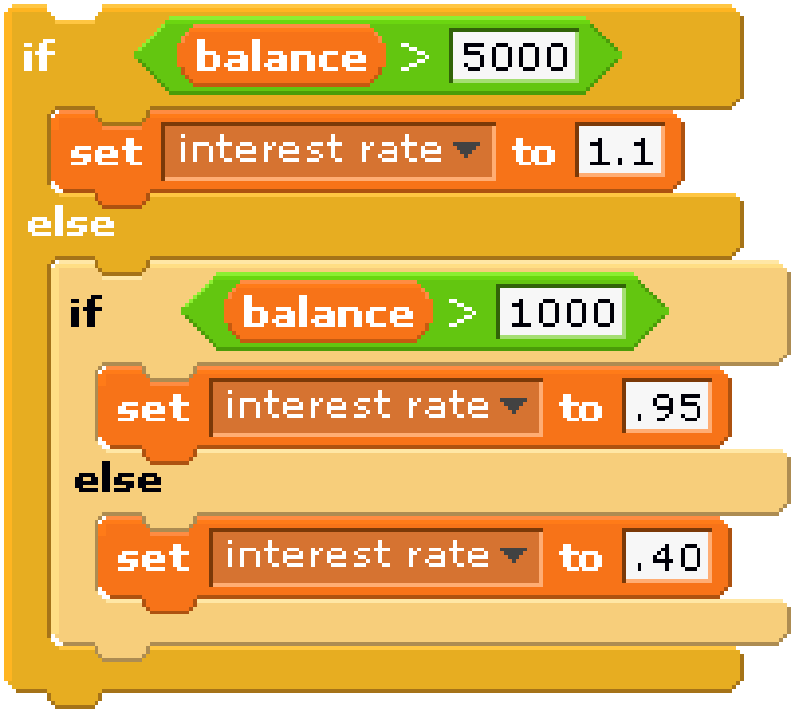
Local (“script”) – can only be read and written by the current script / block

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



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

IF

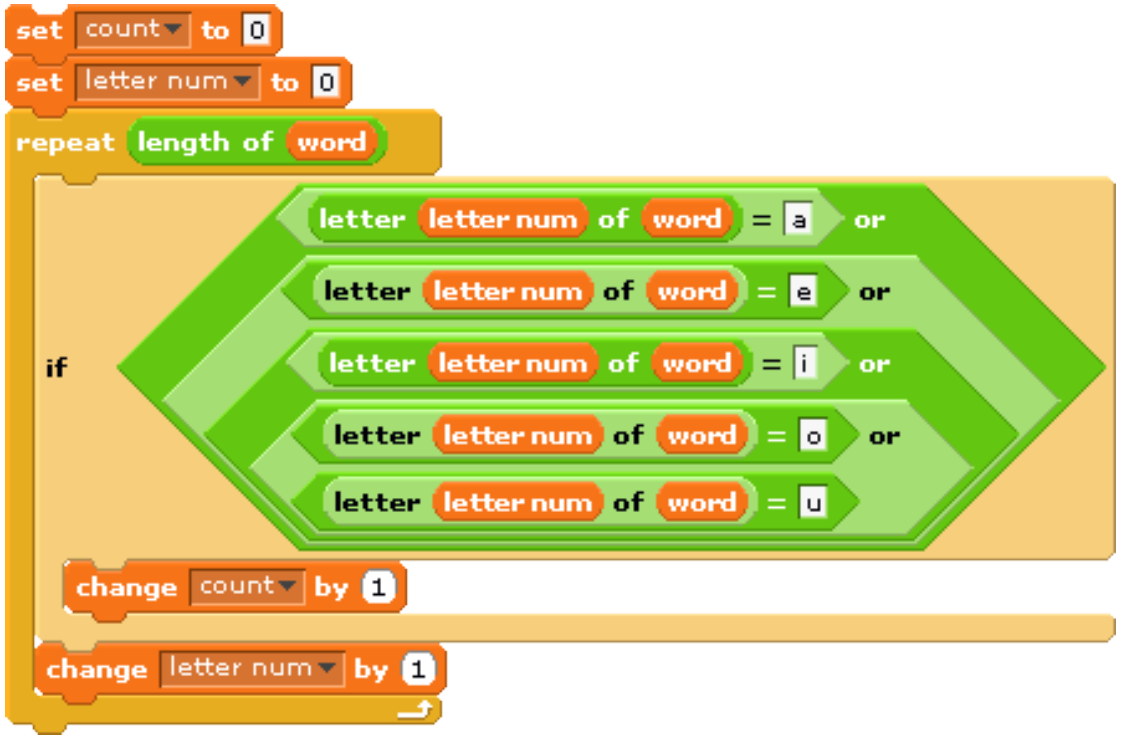
IF / ELSE

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).

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

4 Loops



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

REPEAT

REPEAT UNTIL

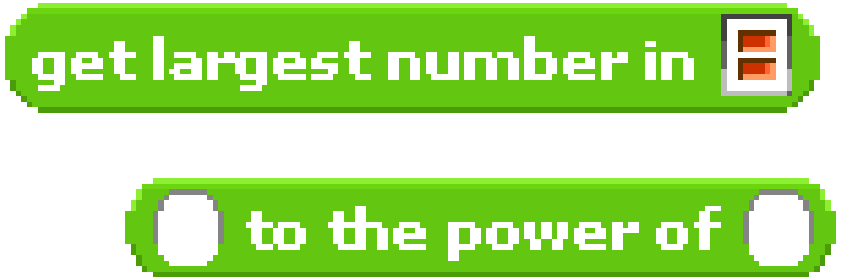
FOREVER

4 Loops: Example

How would you write the following block?

(don't worry about negative exponents)

5 Custom Blocks



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

WHY BUILD BLOCKS?

5 Custom Blocks: Example

How would you build the following block?

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:

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?

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.