COMPUTATIONAL THINKING

- decomposition
- pattern matching
- abstraction
- algorithms

DECOMPOSITION

- Breaking down data, processes, or problems into smaller, manageable parts

PATTERN MATCHING

- Finding similarities between things
- Seeing trends and regularities in data
- Recognizing how problems fit into the pattern of problems you've solved in the past

ABSTRACTION

- Pulling out specific differences to make one solution work for multiple problems
- Identifying the general principles that generate patterns

ALGORITHMS

- A list of steps that you can follow to finish a task
- Developing the step by step instructions for solving a set of problems

DO I NEED TO...

- be able to do something with this piece of info later?

  -> variable
- do the same thing X number of times?

  -> loop
- do the same thing over and over until some condition is met?

  -> loop

- do the same thing to a list of values?

  -> iterate through a collection
- maybe do one thing and maybe do something else?

  -> if/else
- maybe do one thing and maybe do nothing?

  -> if with no else
- keep track of pairs of data?

  -> hash

- Do I need to write some series of operations in more than one place?

  -> define a method and call it in more than one place
- Does this method need any data to do its job? Does its work change slightly each time?

  -> define some arguments
- Does this method do the exact same thing every time I run it?

  -> no arguments needed

- Do I need some data back when I call this method?

  -> that data will be the return value and you'll need to save it in a variable when you call it