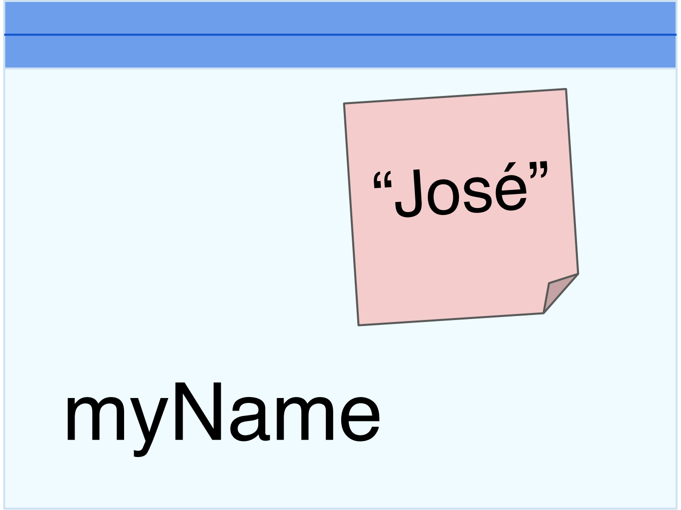
Warm Up

● ○ ○

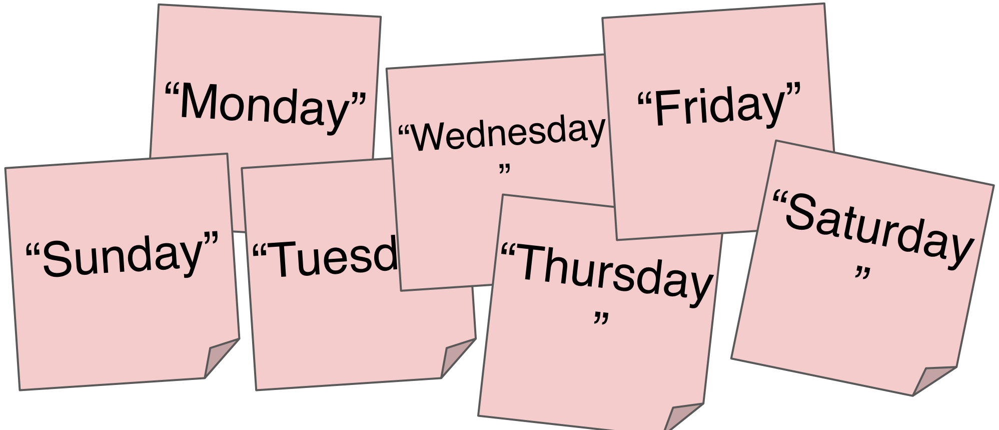


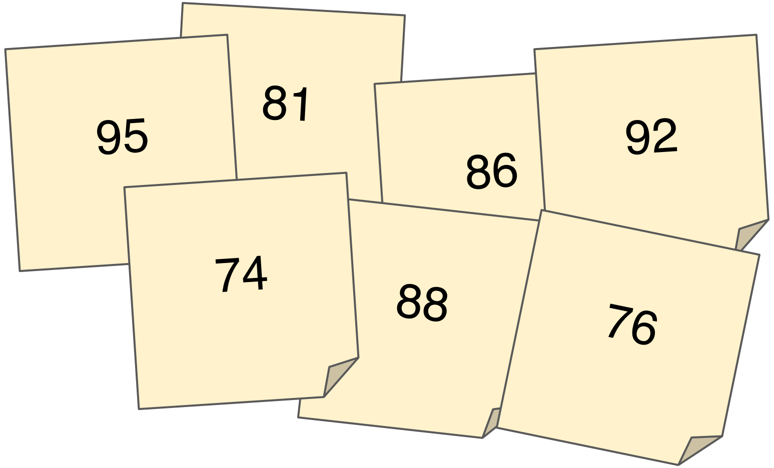
```
var myName =
“José”;
```

But what if we have a lot of information?

Do This:

Think about the challenges that you'd encounter if you need to store 10, or 100, or 1,000 pieces of information using variables.

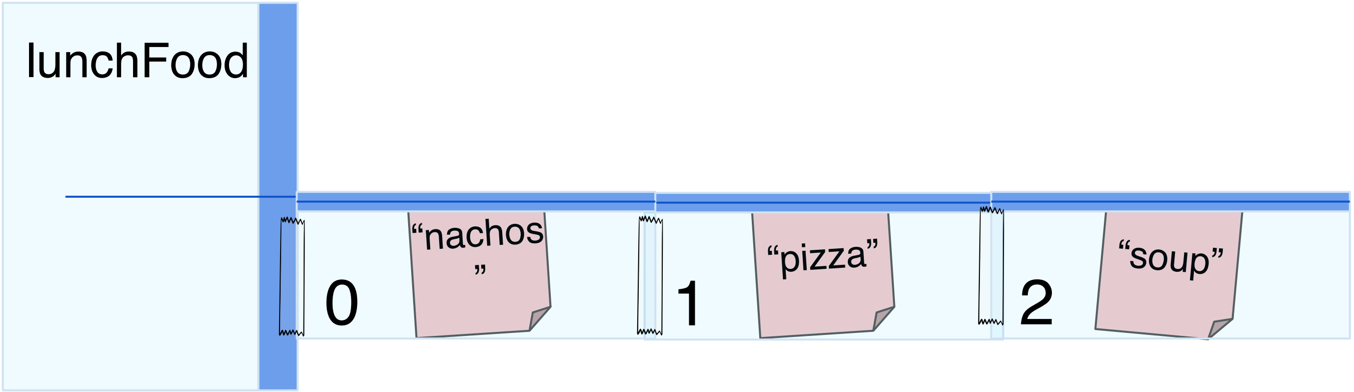


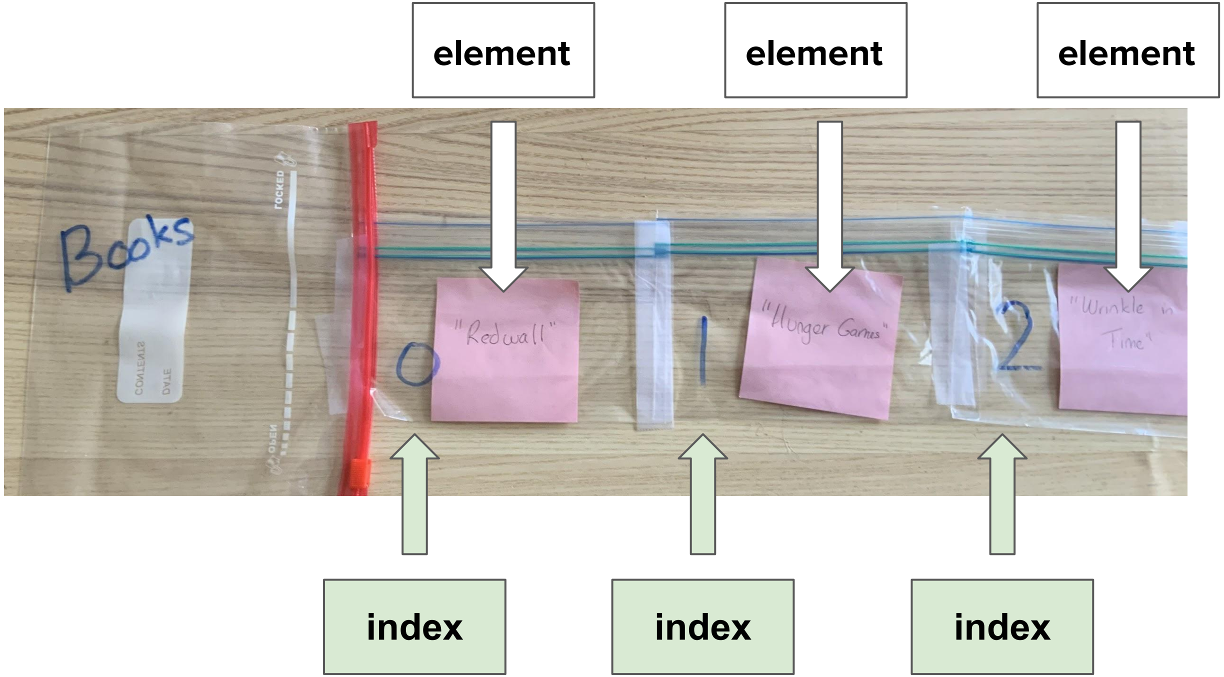


Creating, Accessing, and Updating Arrays/ Lists

Arrays

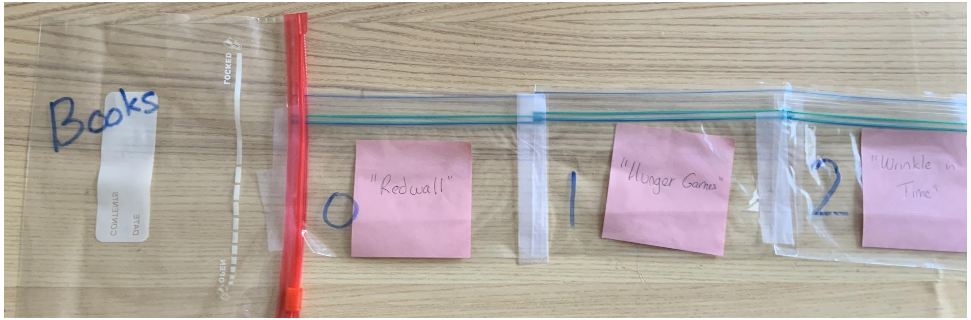
- Gallon baggie connected to smaller baggies
- Each smaller baggie can hold one value
- Gallon baggies are named with the same rules as a variable (no spaces, can’t start with number)
- Smaller baggies are numbered starting at 0

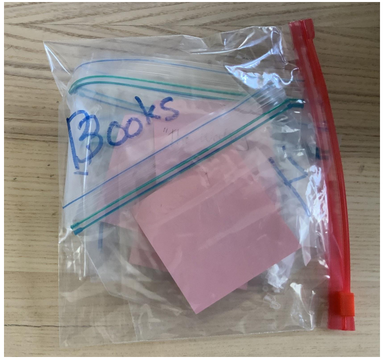




A list is made up of **elements.** Each element has its own **index**. Indexes start at 0 and count up. The **length** of the list is how many elements it contains. This list has 3 elements and indexes from 0 to 2.

Unit 5 Lesson 1

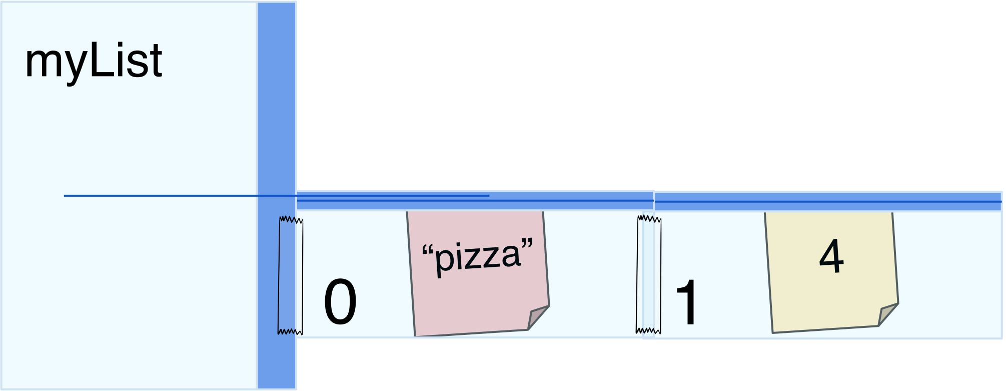




Notice that all the bags can be folded up and be placed inside the big variable baggie. Sometimes we want to think about the whole list, sometimes we want to think about individual elements

```
var myList = [“pizza”, 4];
```

- This command creates a new list and assigns it to a variable
- A list is indicated with square brackets
- Each value in the list is separated by commas



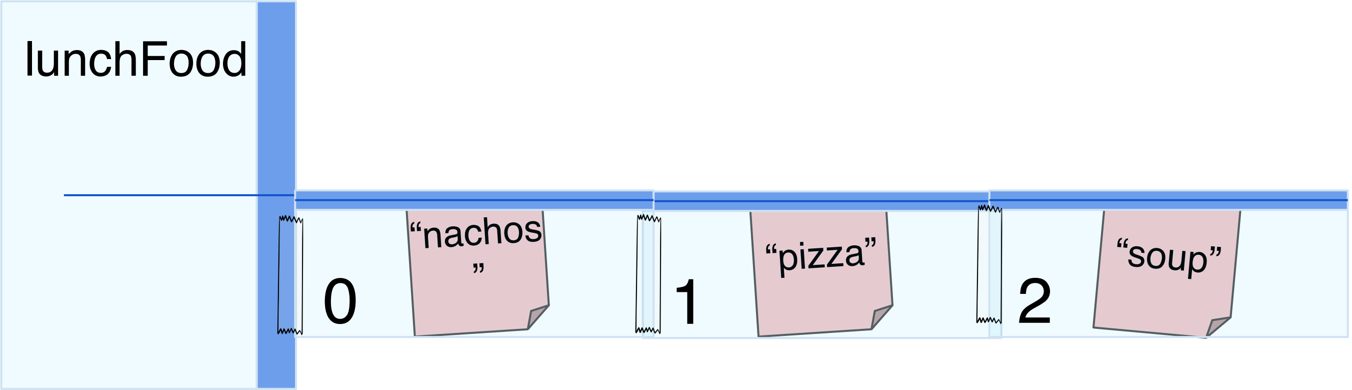
`myList[1]`

- This expression “accesses” the value at that index of the list.
- Also uses square brackets

Do This:

What do lunchFood[0] and lunchFood[2] access?

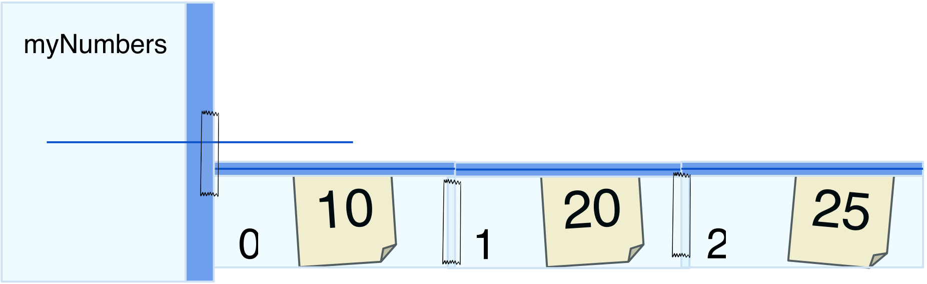
What does lunchFood[3] access?



Lists and Expressions

myNumbers[1]

evaluates to 20



myNumbers[0] + myNumbers[2]

10 25

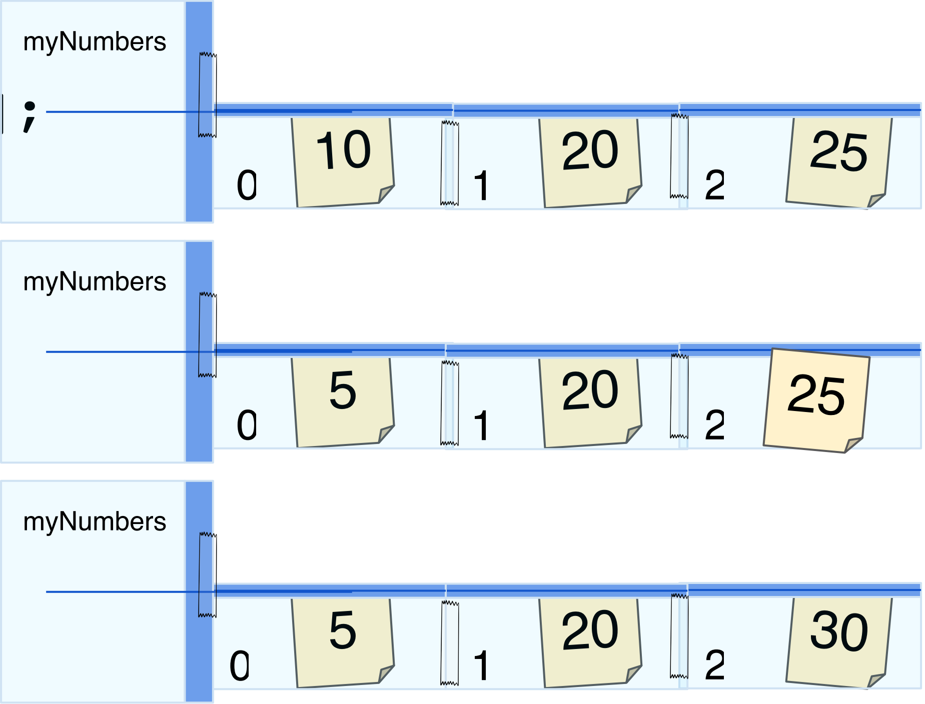
evaluates to 35

Do This: Write an instruction to create a list of numbers. Write three expressions that include accesses to the list you created. Use the examples above for inspiration

`myList[1] = “hello”`

Assigns the value on the right to the index
Just like variable assignment, the old value is thrown away and replaced.

```
00 var myNumbers = [10,20,25];
01 myNumbers[0] = 5;
02 myNumbers[2] = 30;
```

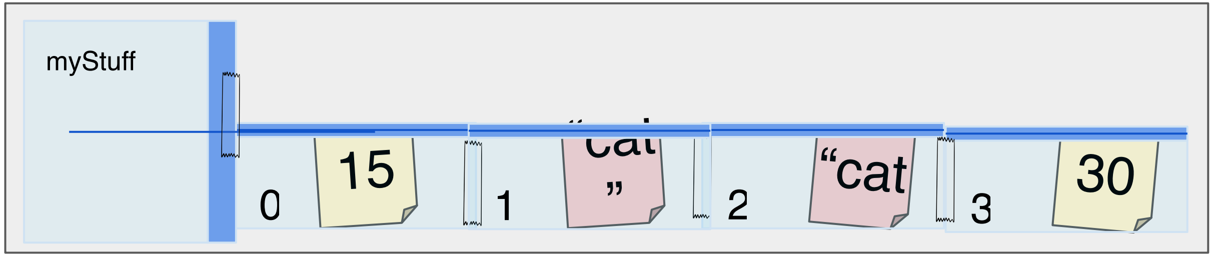


Do This: Think about what the list will contain after line 02 runs.

Do This:

Simulate a run of this program.

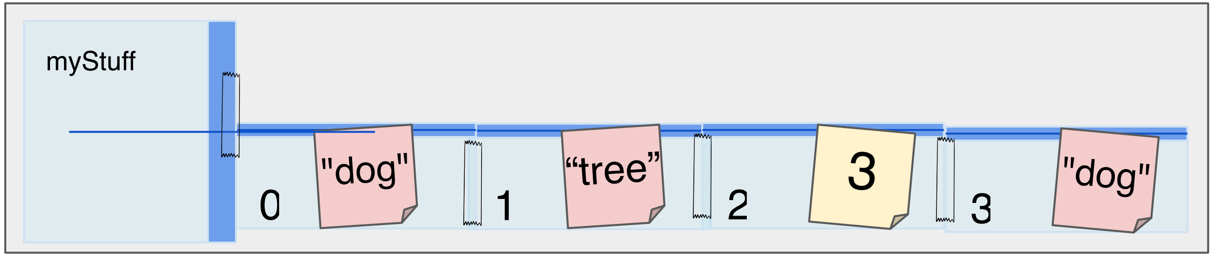
```
var myStuff = [20, “hat”, “pow”, 5];
myStuff[1] = “cat”;
myStuff[2] = myStuff[1];
myStuff[0] = myStuff[3] + 10;
myStuff[3] = myStuff[0] + myStuff[0];
```



Do This:

You can use expressions in the place of the list index.
Simulate.a run of this program.

```
var myStuff = [“dog”,”cat”,3,10];
myStuff[2-1] = “tree”;
myStuff[myStuff[2]] = myStuff[0];
```


myStuff
0
"dog"
1
“tree”
2
3
3
"dog"