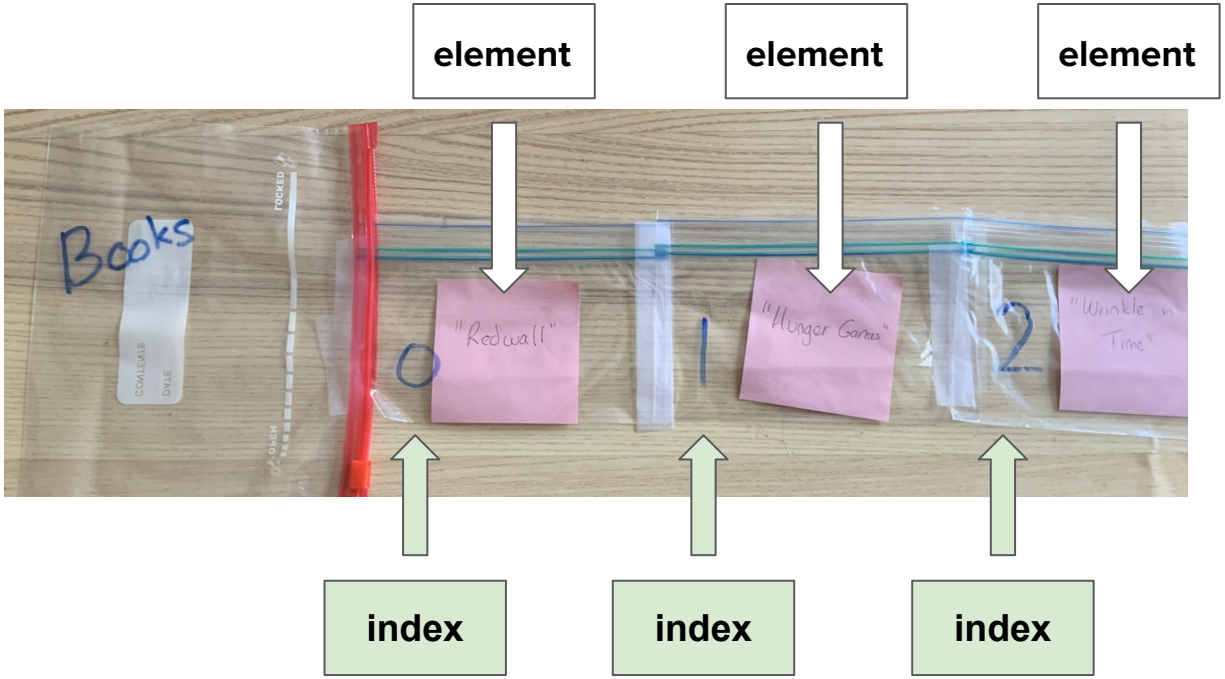
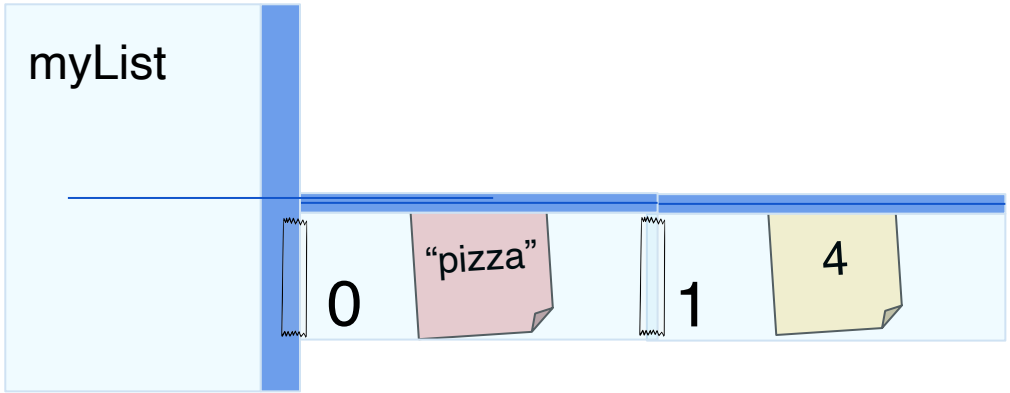
Creating, Accessing, and Updating: Arrays/Lists



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.

```
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

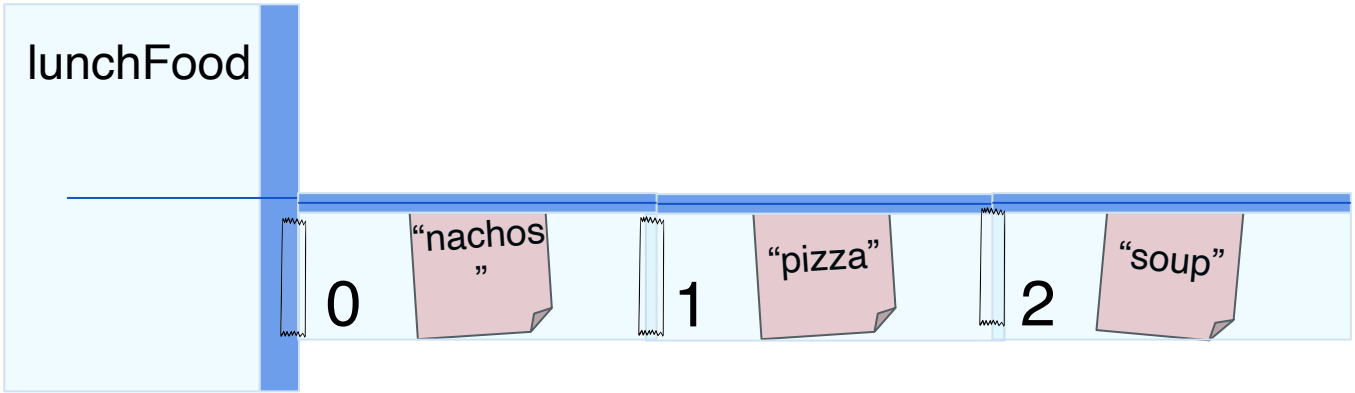


`myList[1]`

- 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?



myList[1] = “hello”

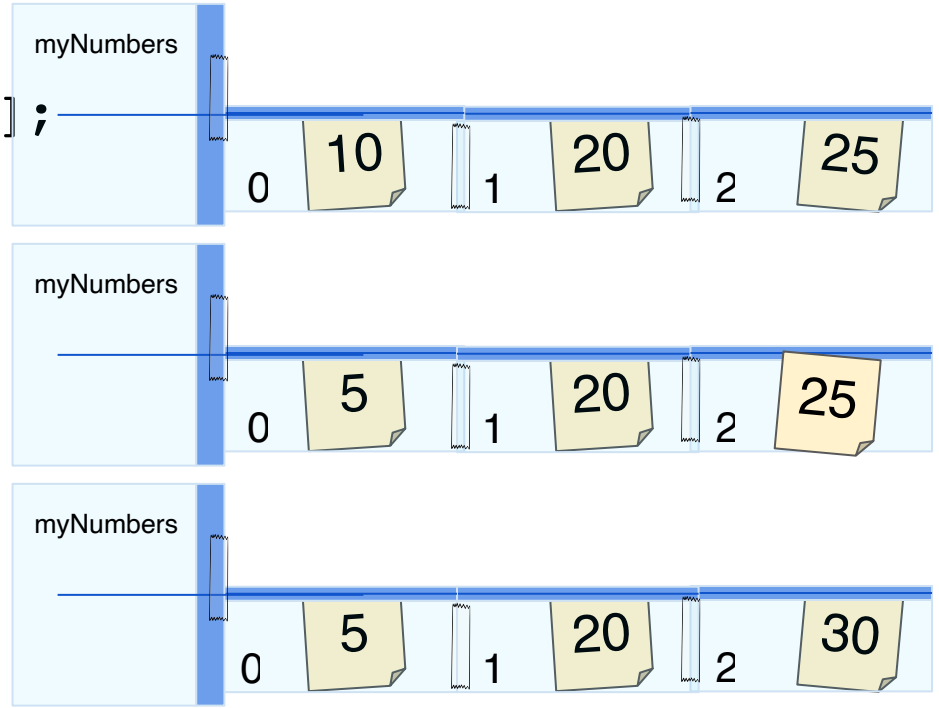
Assigns the value on the right to the index

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;
```

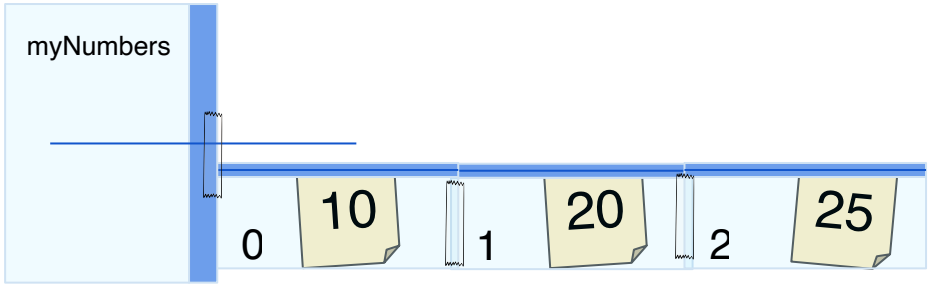


Do This: Think about what the list will contain after line 02 runs.

Lists and Expressions

myNumbers[1]

evaluates to 20



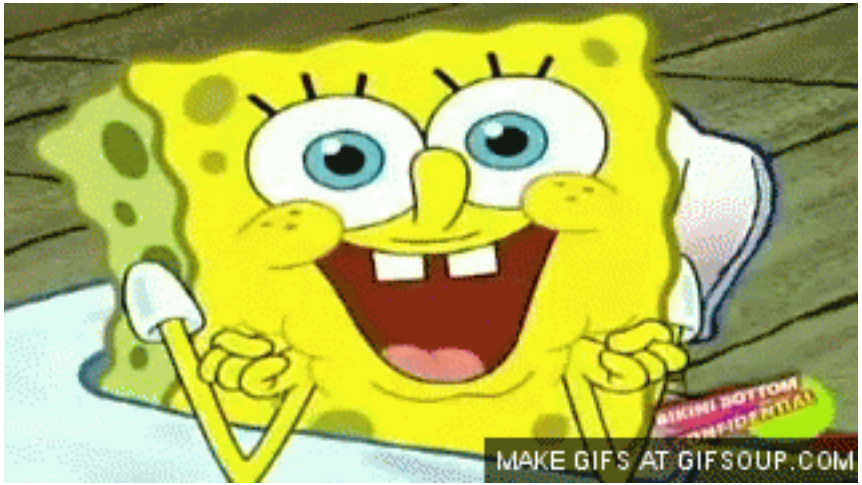
myNumbers[0] + myNumbers[2]

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

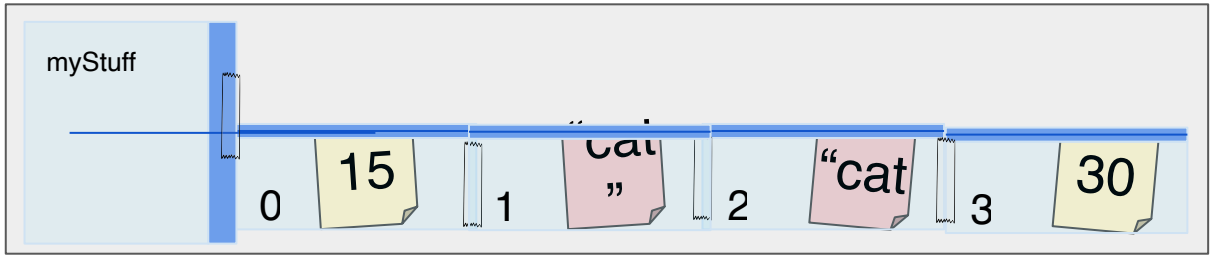
(Breathe!)



Do This:

Simulate a run of this program.

```
00 var myStuff = [20, “hat”, “pow”, 5];
01 myStuff[1] = “cat”;
02 myStuff[2] = myStuff[1];
03 myStuff[0] = myStuff[3] + 10;
04 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.

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