UNIVERSITY OF CALIFORNIA, BERKELEY
Department of Electrical Engineering and Computer Sciences
Computer Science Division

CS10 Fall 2026 **TA: Nahee Jeong**



SOLUTIONS

Discussion 4: Iteration + Lists

For this discussion, I've turned many of the problems into runnable blocks/functions in Snap! You can find the Snap! File here: https://snap.berkeley.edu/snap/snap.html#present:Username=jedi_force&ProjectName=25Fa%20Disc-4%20Solutions

~~**Instructions:**~~

- ~~If you're attending this section in-person, please log into iClicker!~~
- ~~If you missed this discussion, fill out this entire worksheet, and upload it to the Gradescope assignment titled "Discussion 4" by next Discussion.~~

Group Activity / Question of the Day

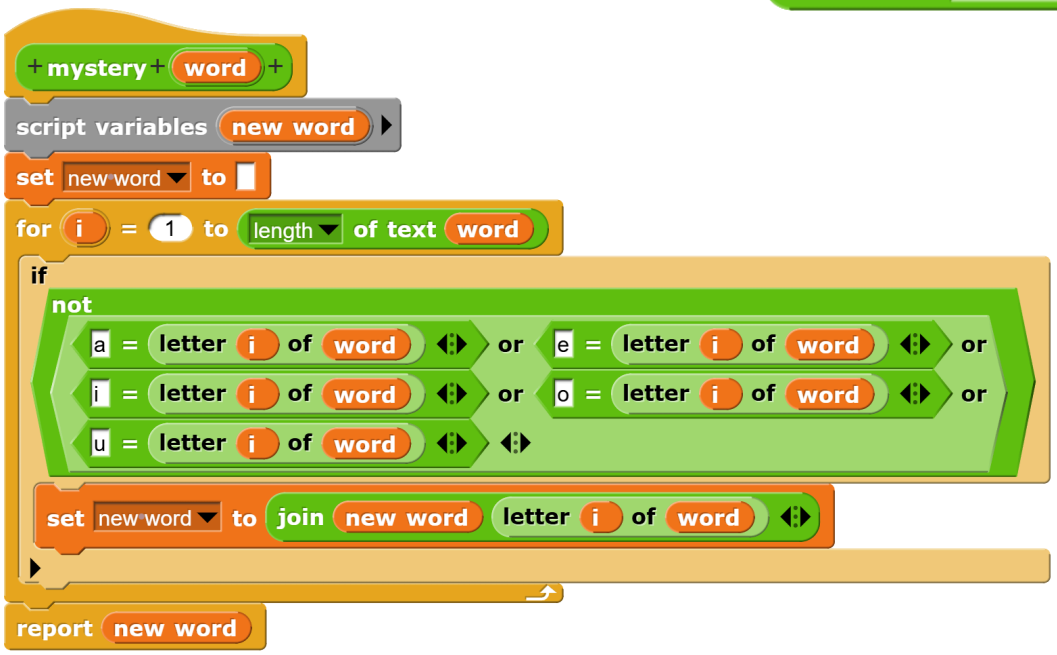
- Ask the person to you – in front / right / left / behind / – if they'd rather live in a house that's always a bit too hot, or a house that's always a bit too cold. Why? What's the average temperature in your hometown?

Required (Pages 2 - 6):

Section I - Iteration

---

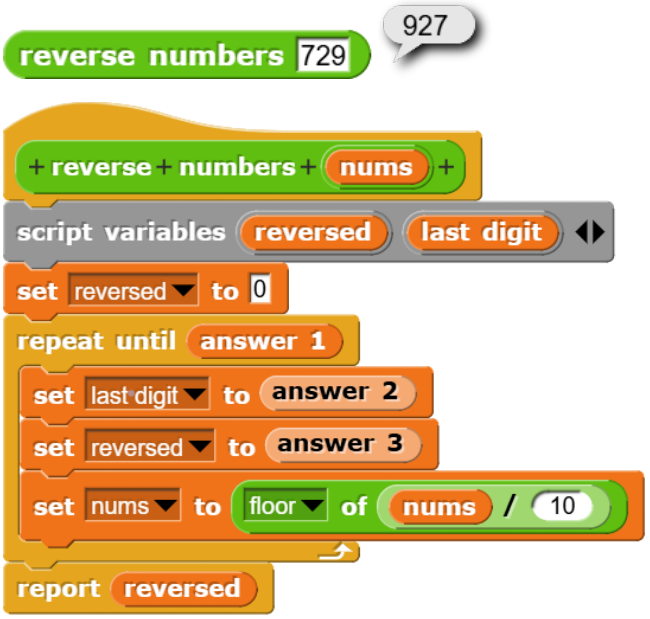
1. What will the following function return, for the input: 



It will output → tmts

For a walkthrough and explanation, please see the Fall 2025 slides
Link: [Fa25 | Discussion 4: Iteration + Lists]

1. Complete the function *`reverse numbers (nums)`* that takes a positive integer *nums* and returns its digits in reverse order. Use the repeat until block and mathematical operations—do not use string manipulation or the join block.



   a. What should `answer 1` be?

      nums = 0

      We don't need to keep iterating once we reach 0 because there's nothing

left to reverse in the “nums” variable

b. What should `answer 2` be?

Nums mod 10

This finds the last digit of a number.

So 729 mod 10 is 9

72 mod 10 is 2

And 7 mod 10 is 7

c. What should `answer 3` be?

(reversed * 10) + last digit

The first time we do this in the first iteration:

Reversed = 0

So the equation would be

(0 * 10) + 9 = 9

In the next iteration, we get

(9 * 10) + 2 = 92

In the last iteration, we get

(92 * 10) + 7 = 927

Section II - Lists

---

1. Select all the ways I can find the smallest number in a list of numbers:

- 

- 

- 

- 

- 



This works because we are sorting the list – and the default is ascending order, so after sorting we get: 1, 2, 3, 4, 8
Then, we ask for the first item of that list, which is 1 – which is the smallest



This expression works because combine is finding the aggregate minimum value from the list. Remember, combine will work by passing the aggregate to the left blank, and the new item to right blank

The first time it's run we get → empty min 3 → 3
Then → 3 min 8 → 3
Then → 3 min 4 → 3
Then → 3 min 2 → 2
Then → 2 min 1 → 1

And since there are no more items in the list, we return 1



This works because "min" is a hyperblock. When we pass the list into the black arrows of the "min" block, it becomes "minimum". It actually works the same way "combine" works.



This DOES NOT work because, "find first" can only have predicate functions passed in. However, "min" is a reporter. So this, actually doesn't return anything – it's just an empty value

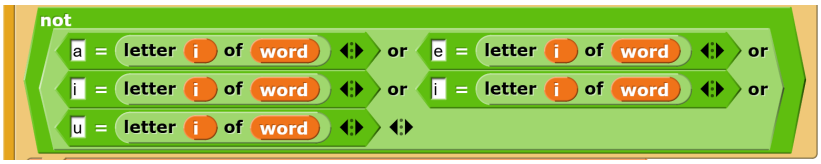


This DOES NOT work, because each value gets passed into the empty slots of "min". But since we have two empty slots, the same value gets passed twice. So instead of finding the minimum value in the list, this expression is doing the following:

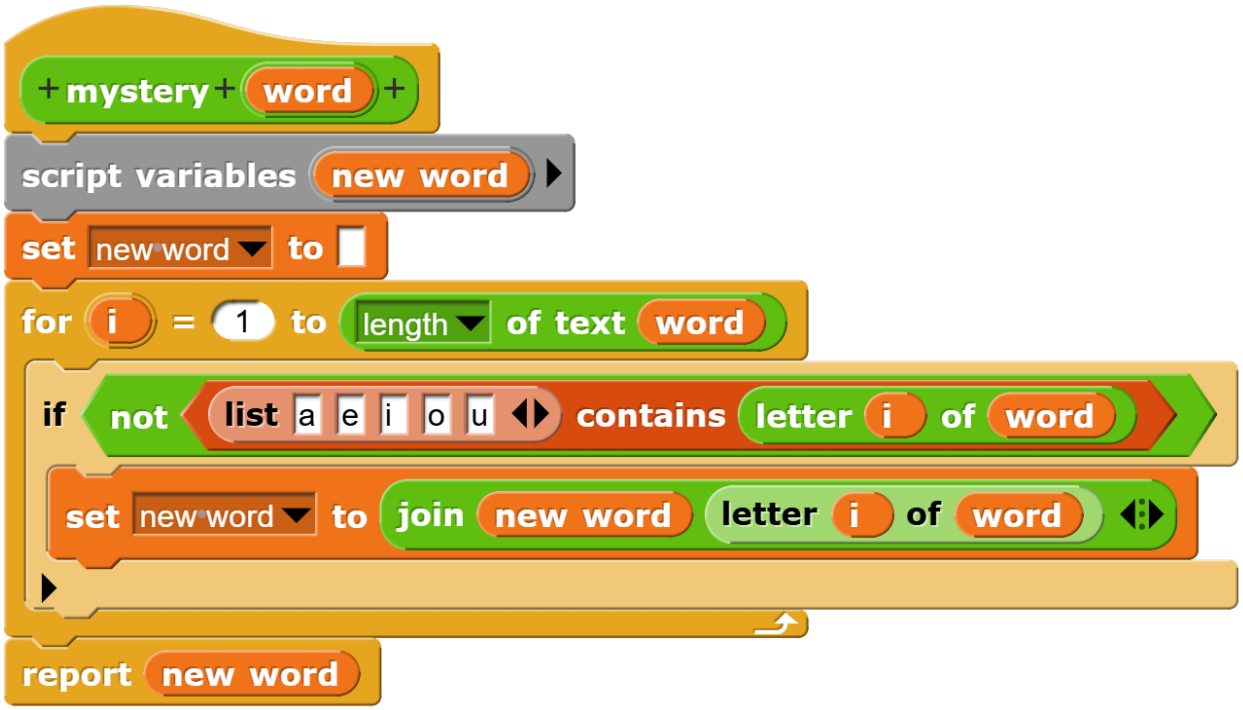
3 min 3 ⟶ 3
8 min 8 ⟶ 8
4 min 4 ⟶ 4
2 min 2 ⟶ 2
1 min 1 ⟶ 1

And it outputs the same list that we inputted: [3, 8, 4, 2, 1]

2. Using the problem from Section. Iteration, Question 1, is there an easier way to write this problem using lists. Specifically, how can we rewrite the if condition with less blocks using list operators?



Here is one way to do this: use the "contains" block and make the vowels into a list!



3. Write a `for loop` that goes through a list of numbers and doubles each item from the list. You will just need to fill out the inside of the `for loop` for this

problem. Fill out the rest of the script for each problem:

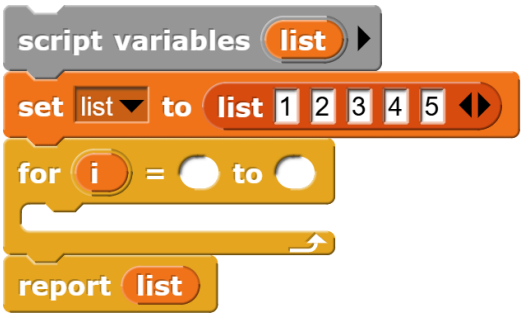
a. For this example, you will not be mutating the original list. Instead, you will be returning "`new list`".



Solution



b. Now, you will do the same problem, but you will need to mutate the list using different blocks. You should not create any more script variables, and you should be returning the original list.



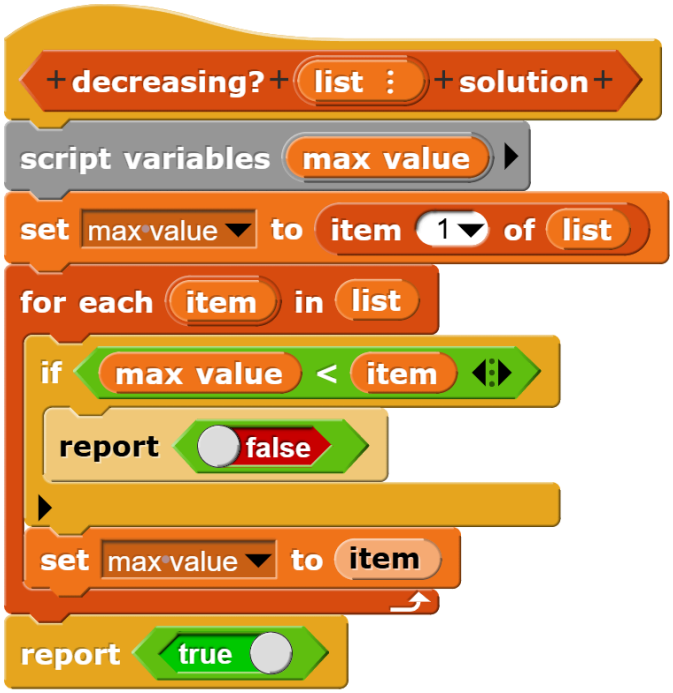
Solution



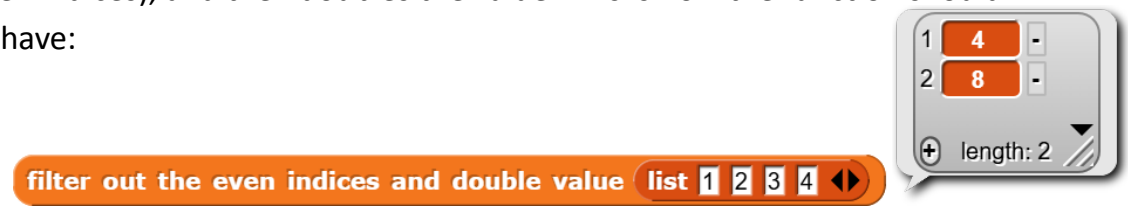
4. Write a function called `decreasing?(list)` which takes in a list of numbers and outputs a boolean. The function should return `true` if all the numbers in the list are in descending order and `false` otherwise



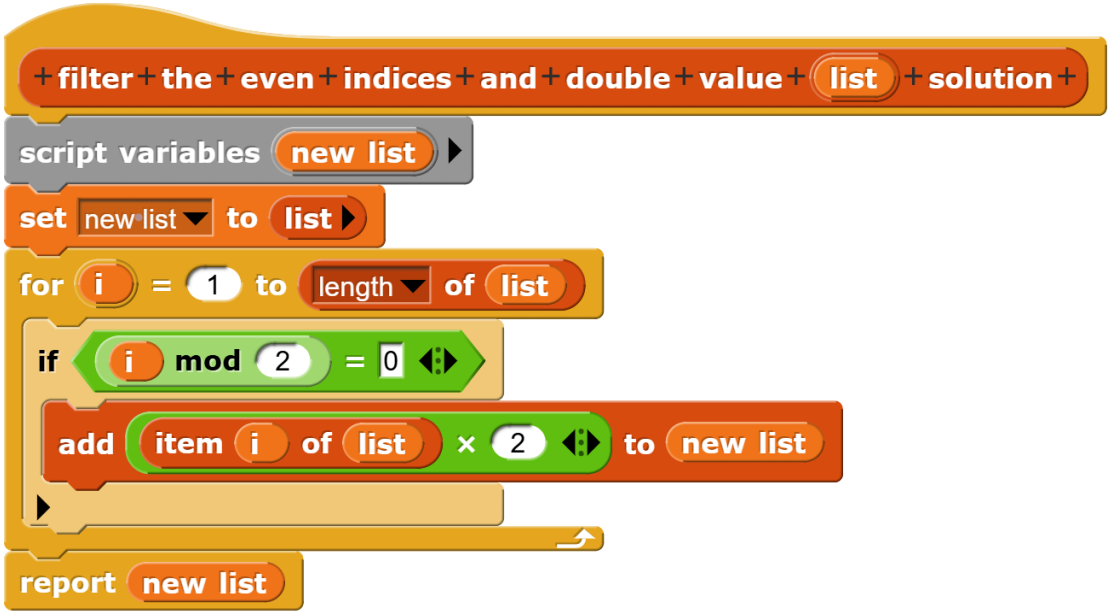
We use the "max value" variable to keep track of the last item. The "max value" (last item) would always be greater than or equal to the current item. If we find one value for "max value" that is not greater than or equal to than we can report False immediately and not check the rest – because this means our list is NOT decreasing.



5. Using iteration, write a function that filters out the odd indices (only keep the even indices), and then doubles the value. This is how the function should behave:

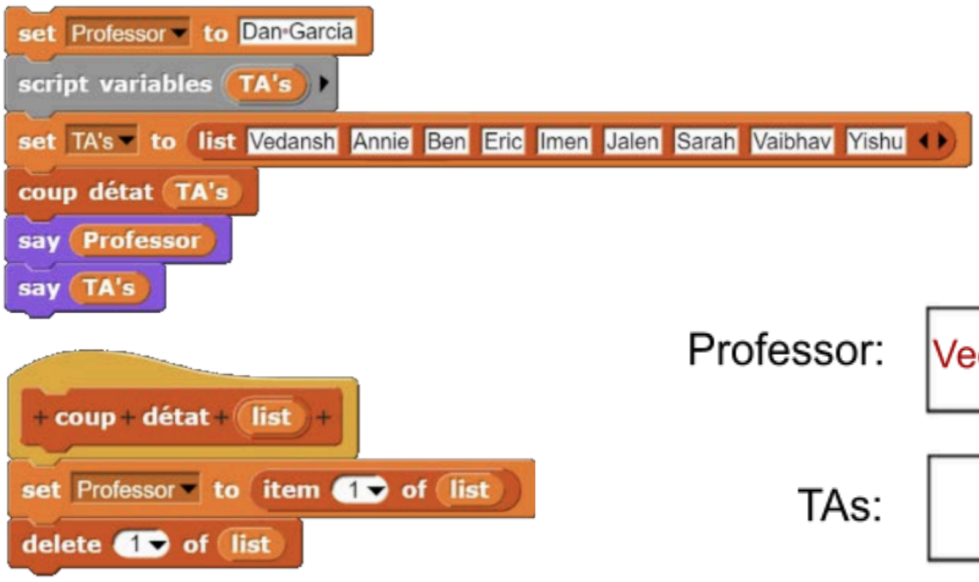


For this solution, we only add items to "new list" if the index/position is an even number



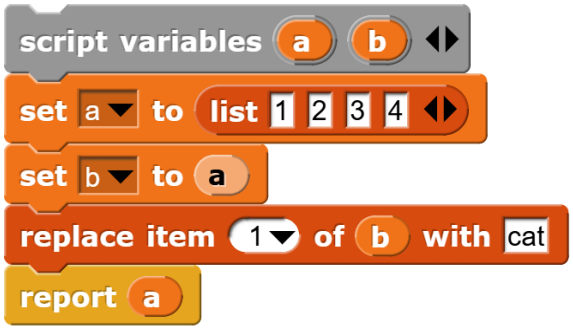
Section II - Mutability vs. Immutability

1. Assume we create a global variable named `Professor` and then run the script:



TAs becomes → [Annie, Ben, Eric, Imen, Jalen, Sarah, Vaibhav, Yishu]

2. For each script below, determine what will be reported / said:

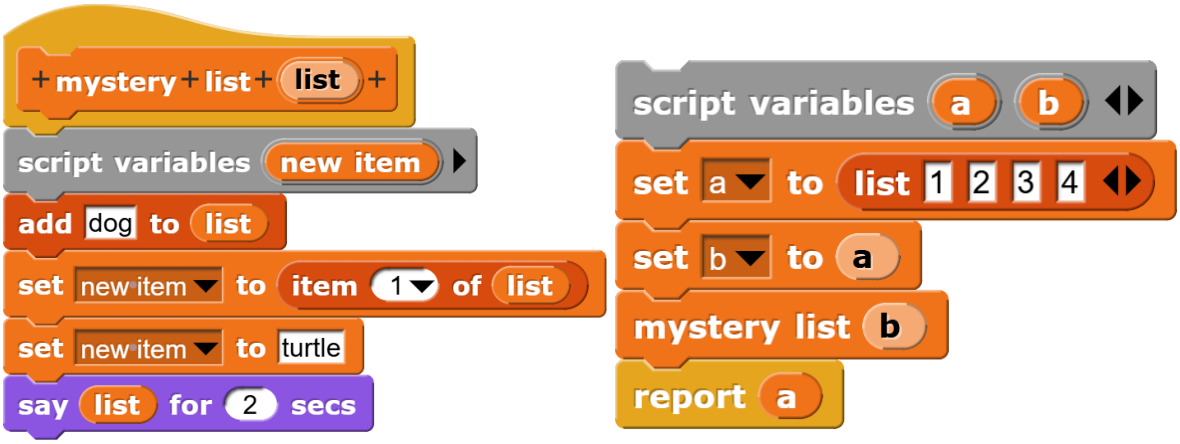


[cat, 2, 3, 4]



[cat, 2, 3, 4]

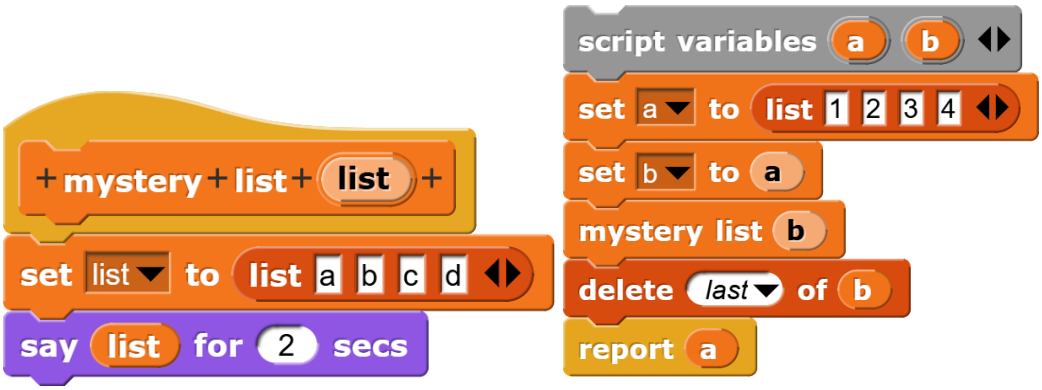
3. Using the function, determine what the scripts will output/say



Say list → [1, 2, 3, 4, dog]

Report a → [1, 2, 3, 4, dog]

4. Using the function, determine what the scripts will output/say



Say list → [a, b, c, d]

Report a → [1, 2, 3]

Optional Extra Practice:

Optional Section I - Lists

Extra Practice

pets

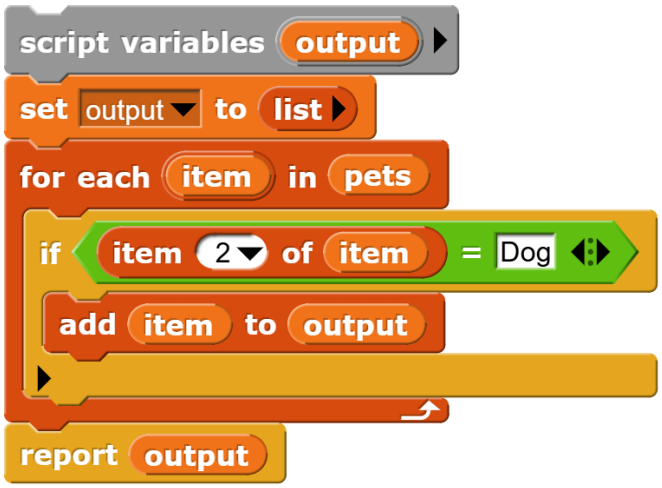
1. You are given a list of pets. Each pet has a name, species, and age. For exan here are the first 3 pets.

3	A	B	C
1	Aaron	Dog	3
2	Murtz	Cat	11
3	Pat	Dog	1

a. Return a list of pets who are dogs.

b. Return the age of the oldest pet.

To return a list of pets who are dogs



2	A	B	C
1	Aaron	Dog	3
2	Pat	Dog	1

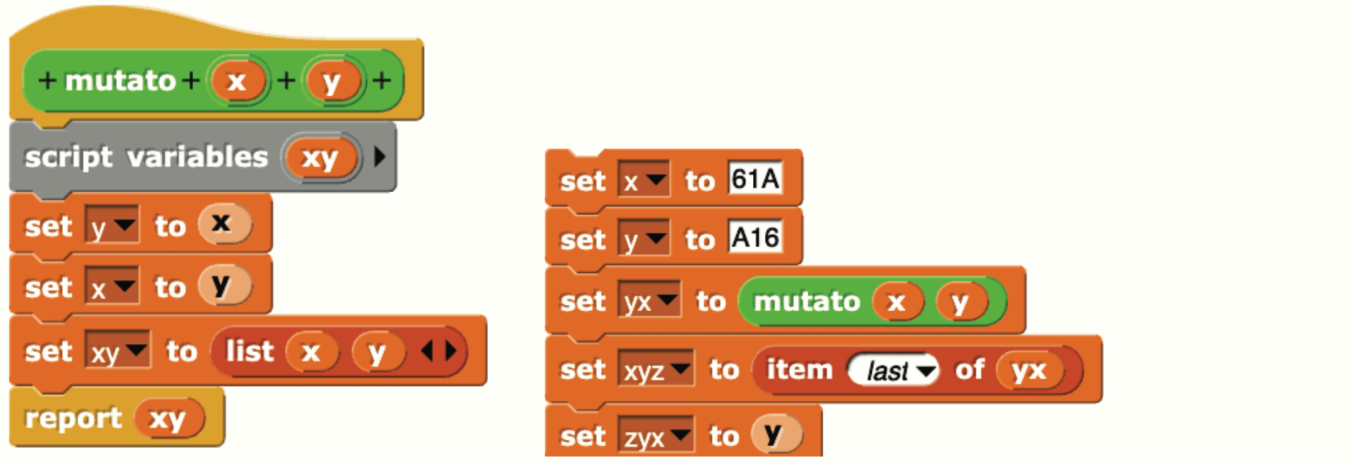
To return the age of the oldest pet



Optional Section II - Mutability vs. Immutability

---

5. List the values of the global variables: `x, y, yx, xyz, zyx` after the following scripts are executed. Assume `mutato` is defined before the script on the right is executed. If an error is caused, mark the "`ERROR`" bubble instead.



x: ____ y: ____ yx: ____ xyz: ____ zyx: ____

◯ ERROR

X = 61A

Y = A16

Yx = [A16, 61A]

Xyz = 61A
Zyx = A16

When we run “mutato”, “mutato” creates it’s own variables named “x” and “y” which are separate / different from the “x” and “y” that are global variables. Since scope will always first look at the nearest variables, when we set “x” and “y” inside of “mutato”, we only change that variables that exist in “mutato”, and we don’t actually reassign the global variables “x” and “y”