

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

CS10 Fall 2026

TA: Nahee Jeong



Discussion 3: Domain/Range, Scoping, Iteration, Lists

Instructions:

- Open snap.berkeley.edu/run on your computer.
- Make a copy of this document and work on it with a peer or as a group! You're welcome to type or write your answers.
- You do not need to submit this worksheet. It's purely for your learning and won't be graded.

Optional Reading Questions:

What is an example of abstraction in your daily life? Using this example, give one reason why abstraction is helpful and another reason how it can do more harm than good.

Section I - Data-types, Domain, & Range

1. Write down the following for the blocks below: 'type', domain, and range:

<i>Block</i>	<i>Type</i>	<i>Domain</i>	<i>Range</i>
			
			
			
			

2. Now, suppose that $f(g(f(g(f(g(x))))))$ does not error for any x that is part of g 's domain. Which of the following statements must be True? Circle your answer(s).

- A. g 's range is a subset of f 's domain.
- B. g 's range is a subset of f 's range.
- C. g 's domain is a subset of f 's range.
- D. g 's domain is a subset of f 's domain.
- E. g 's range is equal to f 's domain.
- F. g 's range is equal to f 's range.
- G. g 's domain is equal to f 's domain.
- H. g 's domain is equal to f 's range.
- I. f 's range is a subset of g 's domain.
- J. f 's range is a subset of g 's range.
- K. f 's domain is a subset of g 's range.
- L. f 's domain is a subset of g 's domain.

3. Consider the following expression. Determine the data types of each function's input and output. If the function does not have an output, enter: none



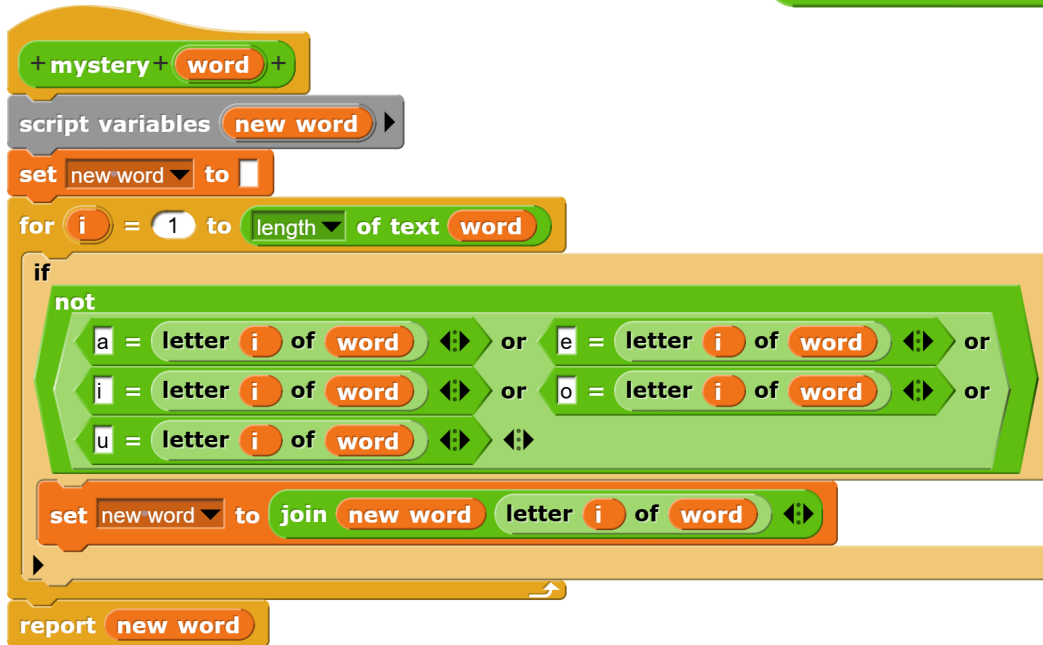
Function Name	Domain	Range
foo		
sqrt		
function		



Function Name	Domain	Range
mystery		
all but first of		
foo		
function		

Section II - Iteration

1. What will the following function return for the input: **mystery** tomatoes

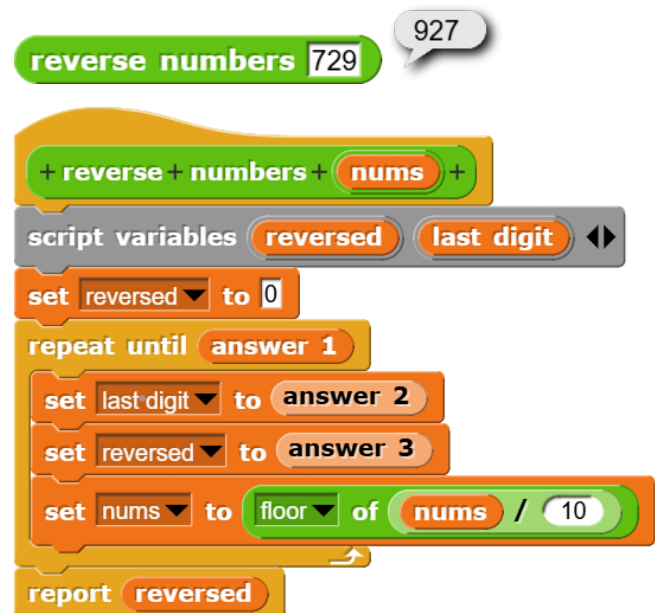


1. Complete the function *reverse* *numbers* (*nums*) that takes a positive integer *nums* and returns its digits in reverse order. Use the repeat until block and mathematical operations—do not use string manipulation or the join block.

a. What should answer 1 be?

b. What should answer 2 be?

c. What should answer 3 be?

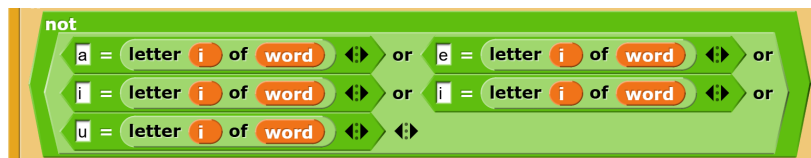


Section III - Lists

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

-
-
-
-
-

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?



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".



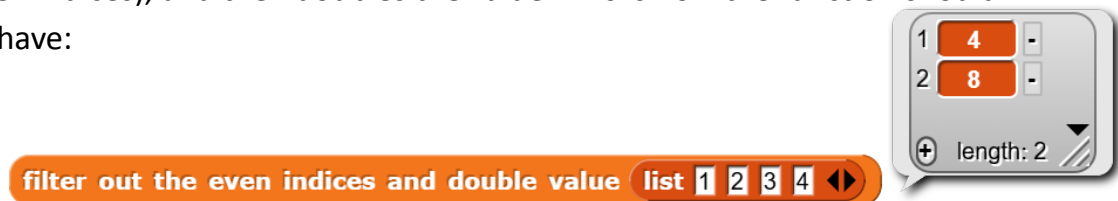
- b. Now, you will do the same problem, but you will need to mutate the list using different blocks. You should not create any more script variables, and you should be returning the original list.



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

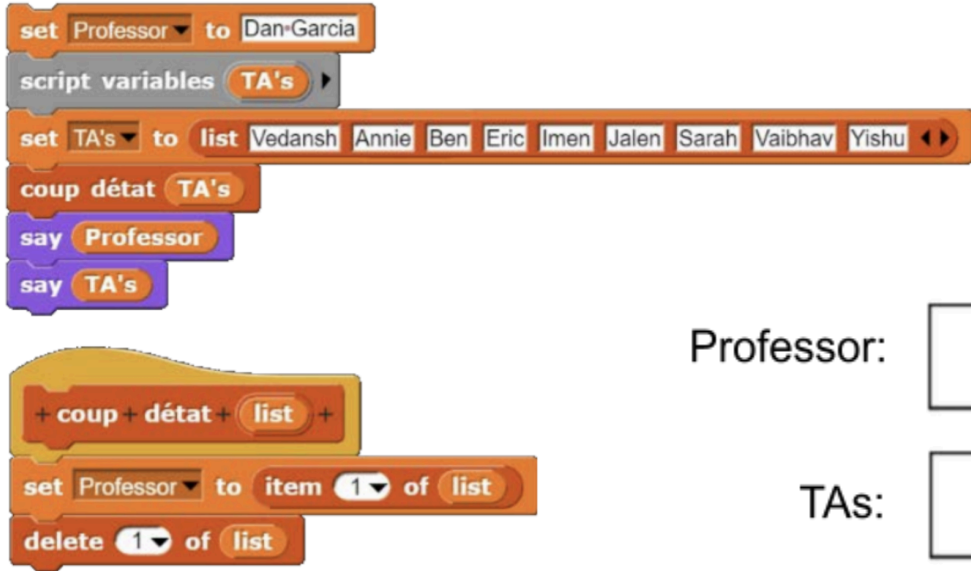


5. Using iteration, write a function that filters out the odd indices (only keep the even indices), and then doubles the value. This is how the function should behave:



Section III - Mutability vs. Immutability

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



The script consists of the following blocks:

- `set Professor to Dan-Garcia`
- `script variables TA's`
- `set TA's to list Vedansh Annie Ben Eric Imen Jalen Sarah Vaibhav Yishu`
- `coup d'état TA's`
- `say Professor`
- `say TA's`
- `+ coup + d'état + list +`
- `set Professor to item 1 of list`
- `delete 1 of list`

Professor:

TA's:

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



The script consists of the following blocks:

- `script variables a b`
- `set a to list 1 2 3 4`
- `set b to a`
- `replace item 1 of b with cat`
- `report a`



The script consists of the following blocks:

- `script variables a b`
- `set a to list 1 2 3 4`
- `set b to a`
- `replace item 1 of a with cat`
- `report b`

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



The script consists of the following blocks:

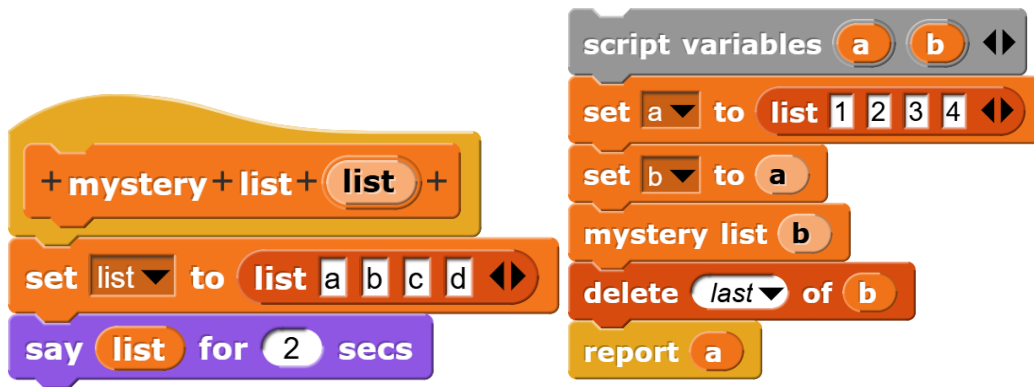
- `+ mystery + list + list +`
- `script variables new item`
- `add dog to list`
- `set new item to item 1 of list`
- `set new item to turtle`
- `say list for 2 secs`



The script consists of the following blocks:

- `script variables a b`
- `set a to list 1 2 3 4`
- `set b to a`
- `mystery list b`
- `report a`

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



Optional Extra Practice:

Optional Section I - Lists

Extra Practice

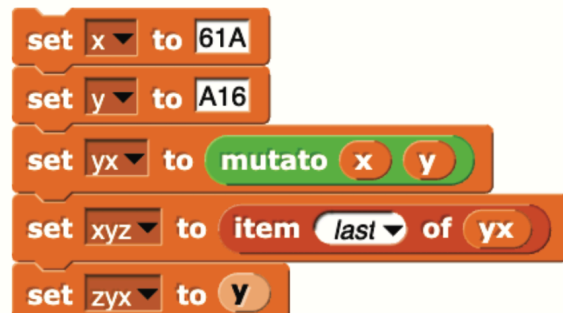
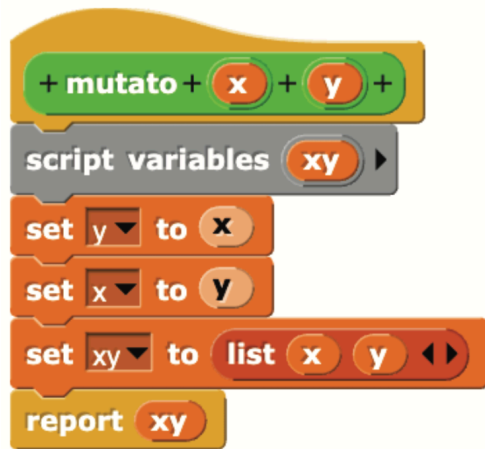
1. You are given a list of pets. Each pet has a name, species, and age. For example, here are the first 3 pets.



b. 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