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

CS10 Summer 2024

TAs: Wen Cao & Naveen Nathan

Discussion-3: Domain / Range, Iteration, Scope, Lists

Instructions

- If you're attending this section in-person, at the end of section, you'll be given a secret phrase. Please enter it here: https://tinyurl.com/cs10su24
- If you missed this discussion, fill out this entire worksheet, and upload it to the Gradescope assignment titled "Discussion-3" by next Discussion.
- Please open up snap.berkeley.edu/run on your computer.
- Ask the person to your right / left how much caffeine they consume per week. For reference: a cup of coffee has ~90mg of caffeine.

Q1. Data-types, Domain, & Range

First, your TA will do a mini-lecture on data-types, domain, and range. As you watch the presentation, you'll be able to answer the following questions.

Questions:

1. What's a data-type? Why is it useful to know the data-type of a variable?

2. What does domain mean in the context of data-types? How do we figure out the domain of a Snap*!* block?

3. What does range mean in the context of data-types? How do we figure out the range of a Snap*!* Block?

4. Write down the data-type(s) that corresponds to the following input shapes in Snap*!*:

Round shape, 

Wide rectangle, 

Hexagon, 

2 lines (hamburger), 

Gray ring, 

Narrow rectangle, 

5. What's a command block? What's its shape? What (if anything) can we say about the domain / range of a command block?

6. What's a reporter block? What's its shape? What (if anything) can we say about the domain / range of a reporter block?

7. What's a predicate block? What's its shape? What (if anything) can we say about the domain / range of a predicate block?

8. Write down the types of the following blocks:







9. Fill in the blanks:



Domain of `contains` : _____

Range of `contains` : _____



Domain of `foo` : _____

Range of `foo` : _____

Data Type of `var` : _____

10. 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?

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.

Q2. Scoping

First, your TA will do a mini-lecture on scoping. As you watch the presentation, you'll be able to answer the following questions.

Problems:

1. What does *scope* mean in the context of variables? What's the difference between a local variable and a global variable?

2. What's the scope of a script variable? An input variable?

3. In the following scripts, determine what's *said* by the sprite. If an error is raised, write "ERROR."

A.

1. Consider the following script. `Instructor` and `Professor` are global variables:



What does Snap*!* say first?

i. ◯ Vedansh ii. ◯ Dan iii. ◯ Snap*!* says nothing iv. ◯ An ERROR occurs

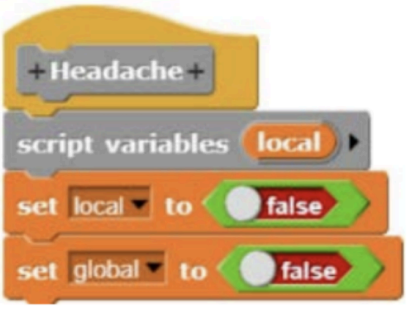
What does Snap*!* say second?

i. ◯ Vedansh ii. ◯ Dan iii. ◯ Snap*!* says nothing iv. ◯ An ERROR occurs

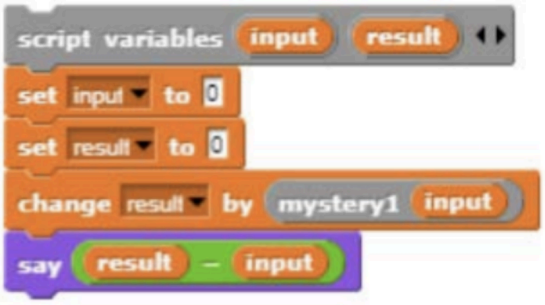
B.

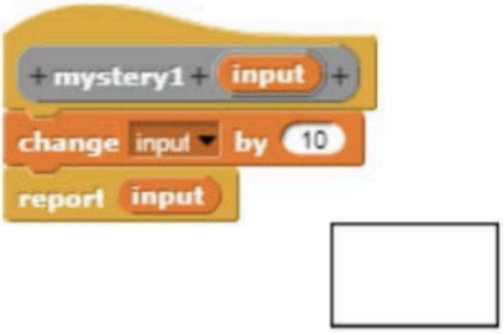
Assume we create a global variable named `global` (and no other global variables) and then run the script below:



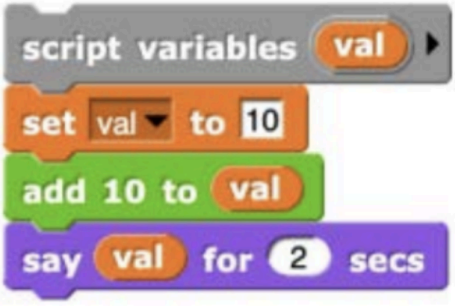


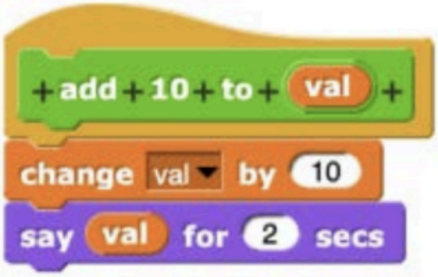
C.





D.





Q3. Lists

Problems:

1. A *list* is a data-structure: it is a container that allows us to store a collection of values. The following words describe some properties of lists. Next to each word, write down what it means:

A. Lists are *iterable*:

B. Lists are *ordered* and *indexable*:

C. Lists are *mutable*:

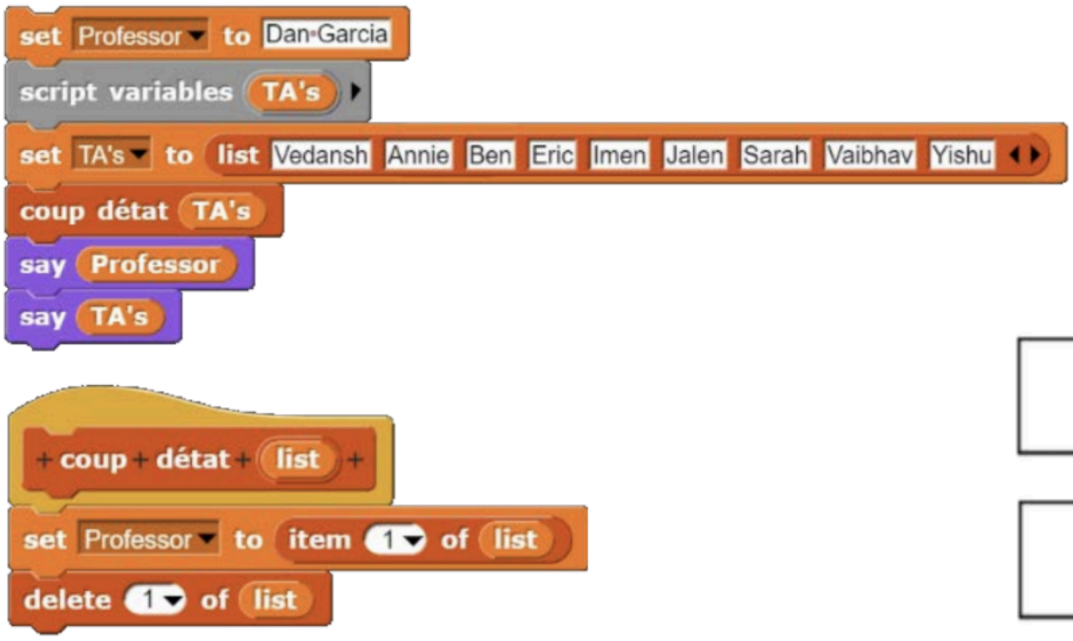
2. The *mutability* of lists is very useful, and important to understand. The TA will now talk about mutability. Pay attention to the mini-lecture, and try to answer the following questions:

A. What are the built-in blocks in Snap*!* that mutate lists?

B. What are some other data-types we've seen in Snap*!* that are mutable?

C. What would Snap*!* say?

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



D.

E.

| **10.** What is said first, second, and third? If there's an error, write a capital 'E.'



First: Second: Third: