Environment Diagram Questions

You are not logged in.

If you are a current student, please Log In for full access to the web site.
Note that this link will take you to an external site (`https://shimmer.mit.edu`) to authenticate, and then you will be redirected back to this page.

Note

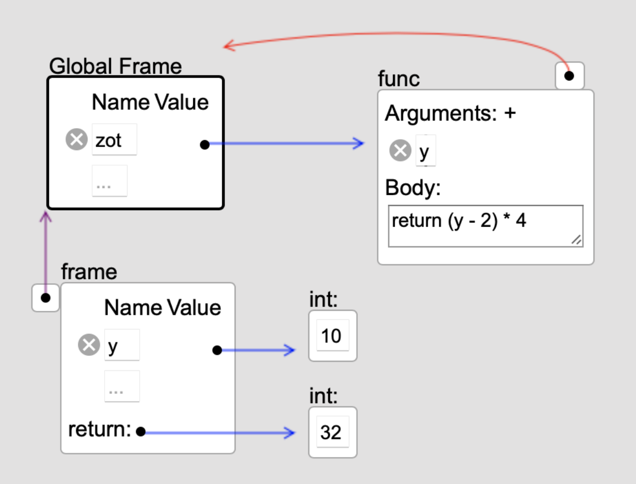
Note that these questions do not affect your grade in 6.009 in any way! They are presented here for two reasons: to provide some extra practice with environment diagrams leading up to the quiz, and to help test the tool used below (for drawing and testing the diagrams), which is being developed as part of an MEng thesis project.

Problem 0 - Diagramming Interface

In order to practice drawing environment diagrams, this question tasks you with using a drag-and-drop interface to construct a diagram.

The later questions on this page will ask you to develop diagrams on your own, but this question is designed to help you familiarize yourself with the interface itself.

Please use the interface below to re-create the following diagram. (note that the global frame has been provided)



i frame int float str bool function frame int float str bool function

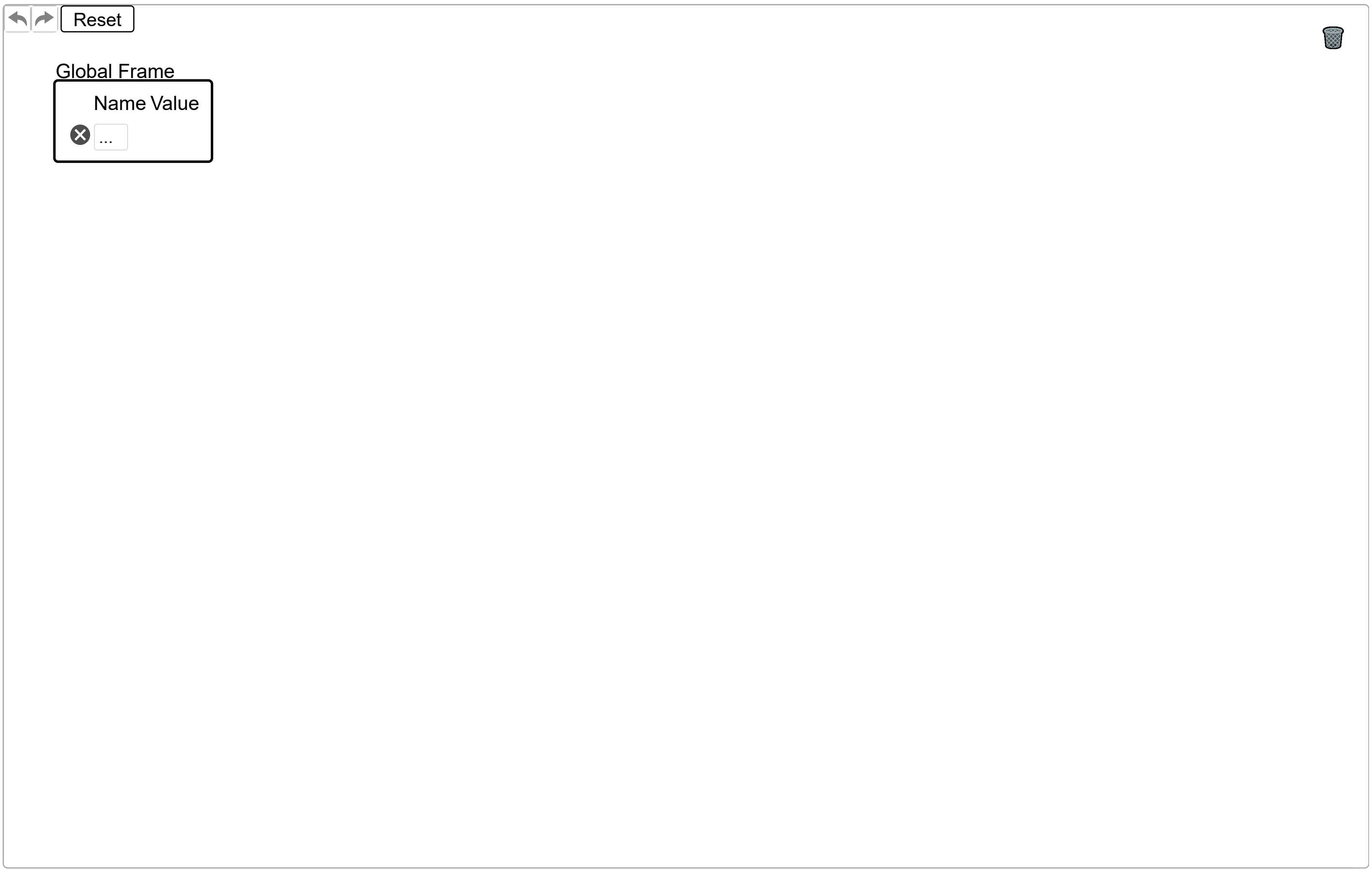
Problem 1 - Function Definition

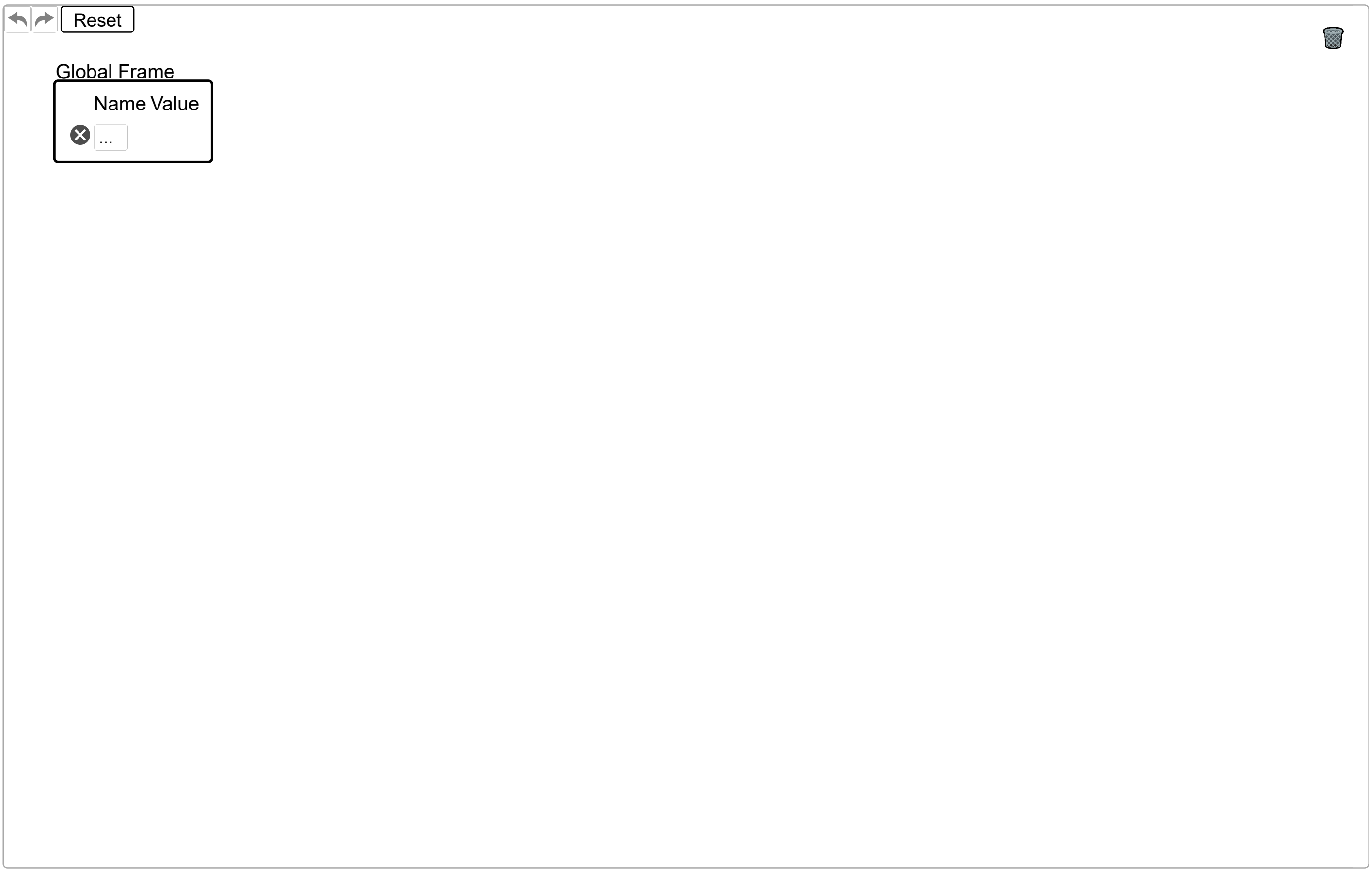
Create a diagram representing the result of executing the following code, consisting of a single function definition. Remember that a function object must contain the argument names, the function's body, and a pointer to the frame in which it was defined (its "enclosing frame").

```
def foo(n, k):
    d = 2
    return k/(n+k)
```

i frame int float str bool None tuple list function frame int float str bool None tuple list function

Problem 2 - Calling Functions





When calling a function, we follow several steps:

1. Evaluate the function to be called, and its arguments (in order)
2. Create a new frame for the function call, with the function's enclosing frame as its parent
3. Bind the parameters of the function to the given arguments in this new frame
4. Execute the body of the function in this new frame.

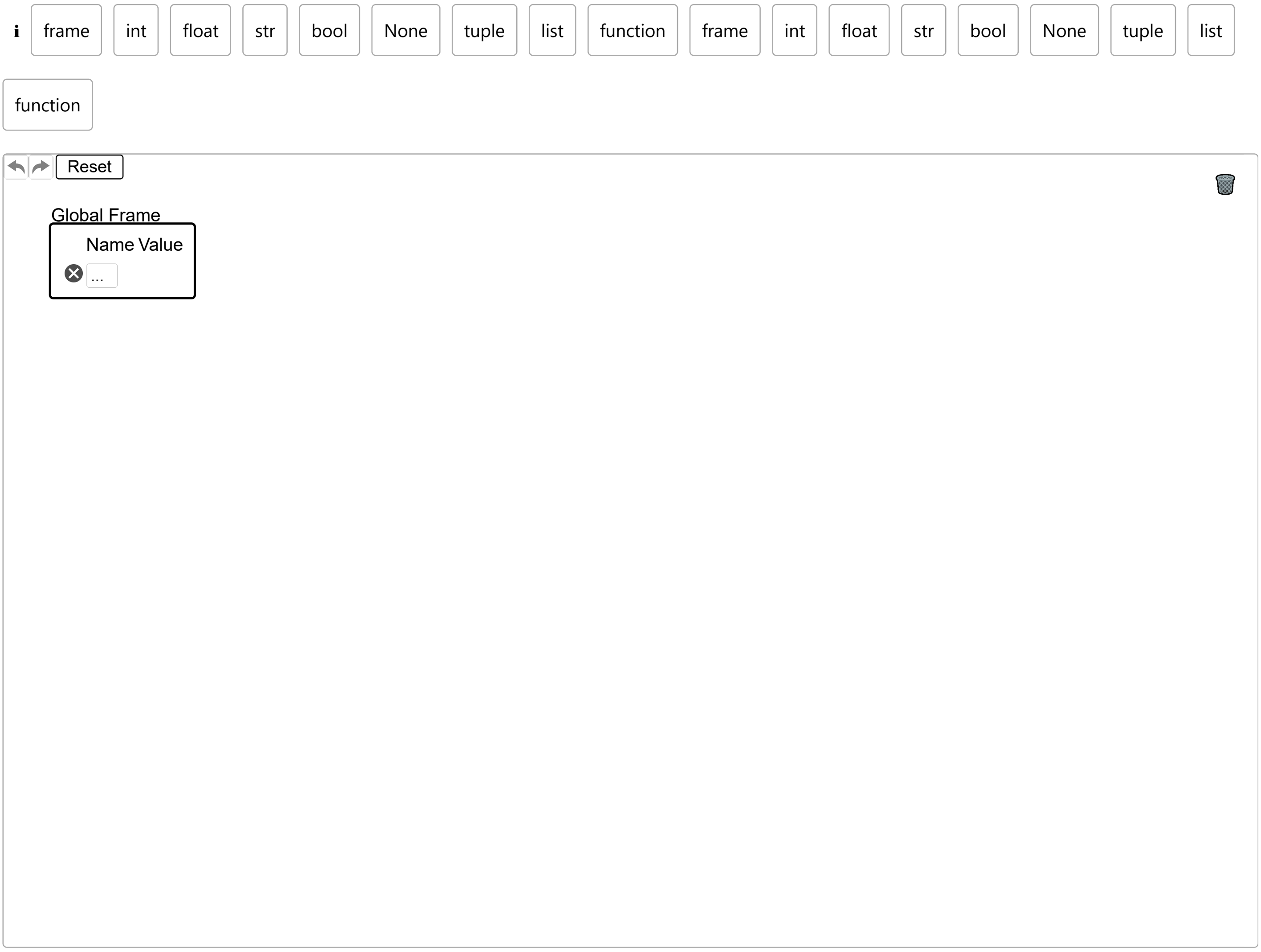
This process is demonstrated through the examples from the recitation 0 materials.

Note that in this interface, we specify each frame's parent with a line from the dot at the top of each frame to its parent. Frames can also have a return value. We will diagram the return value of the function using the `return` pointer at the bottom of the frame. If the `return` pointer isn't set, this corresponds to a function returning `None`.

▶ Show/Hide Hints

Using the interface below, diagram the state of the following program right before the call to `bar(2)` finishes.

```
def bar(x):
    return x**3

bar(2)
```

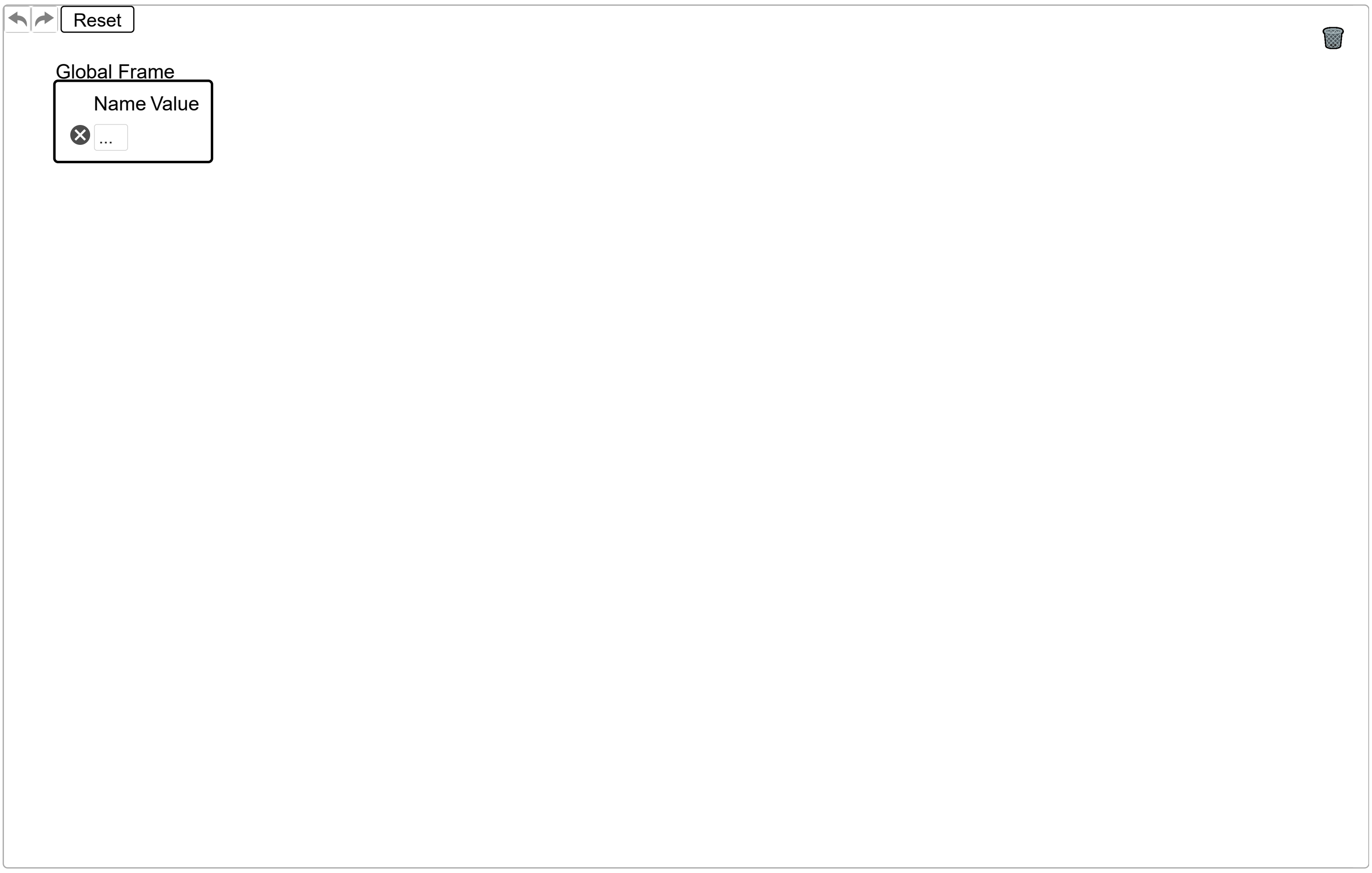


Problem 3 - Diagramming Function Calls

For this problem, diagram the result of executing the following code. Include each of the frames generated by function calls. For this problem, there should be three total frames: the global frame (already provided), one generated by a call to `foo` and one generated by a call to `bar`.

```
def foo(y):
    return bar(y+2)

def bar(x):
    return x*3

foo(10)
```

i frame int float str bool None tuple list function frame int float str bool None tuple list function



Problem 4 - Nested Functions

This problem explores what happens when a function is defined inside of another function. Use the interface below to diagram the state of the program just before the call to `add3(4)` returns.

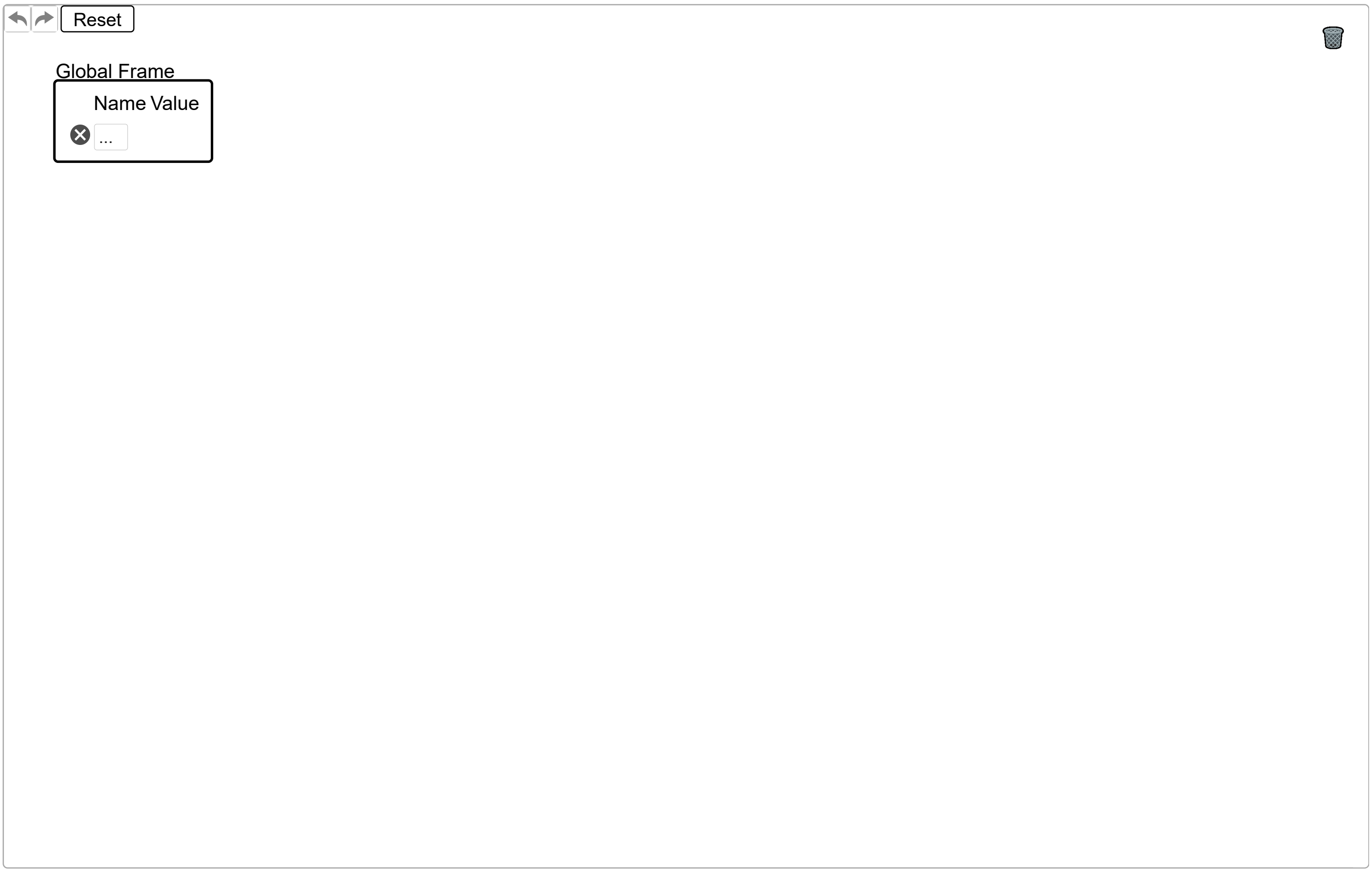
```
def outer(x):
    def inner(y):
        return x+y
    return inner

add3 = outer(3)
add3(4)
```

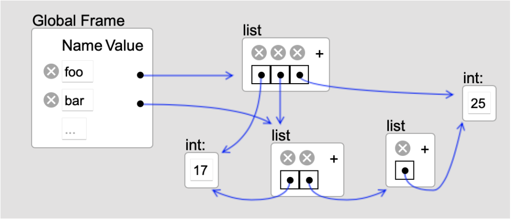
Hint: for this problem, there should be three total frames: the global frame, one generated by `outer(3)` and one generated by `add3(4)`.

i frame int float str bool None tuple list function frame int float str bool None tuple list function

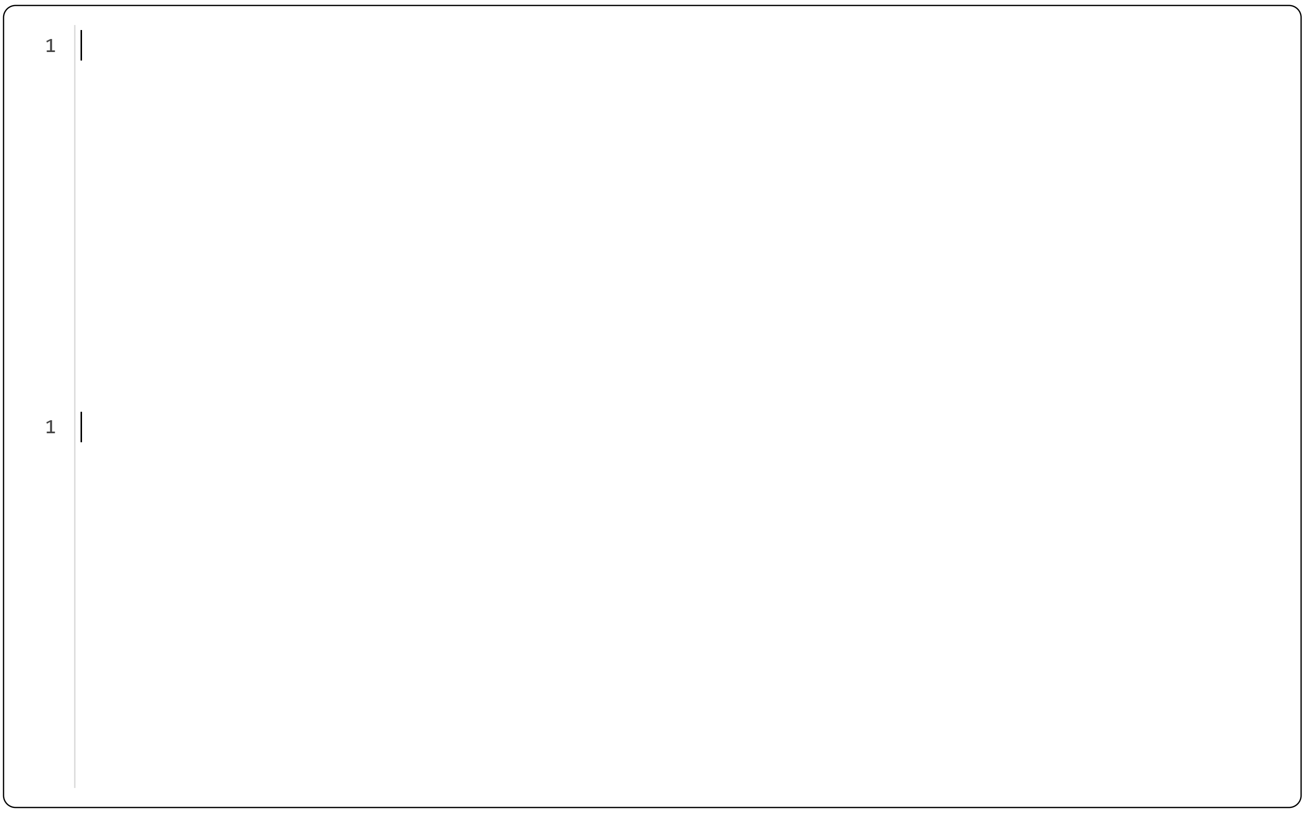
Problem 5 - Code to Generate a Diagram

Consider the following diagram:





In the box below, enter any valid Python program that, when executed, would lead to the diagram above.



Survey

For the diagrams above, on a scale of 1 [very not confident] to 5 [very confident], how confident were you that you knew what diagram to draw? --

On a scale of 1 [very difficult] to 5 [very easy], how easy did you find using the interface to depict your diagrams? --

On a scale of 1 [very unhelpful] to 5 [very helpful], how helpful did you find automated hints while constructing your diagrams? (N/A if not applicable) --

Please provide any comments or suggestions about the diagraming question tool.