Mutability

Announcements

List Mutation

(Demo)

https://pythontutor.com/cp/composingprograms.html#code=s%20%3D%20%5B3,%205,%207,%209,%2011%5D%0Afor%20i%20in%20range%28len%28s%29%29%3A%0A%20%20%20%20s%5Bi%5D%20%3D%20s%5B-%28i%2B1%29%5D&cumulative=true&curInstr=0&mode=display&origin=composingprograms.js&py=3&rawInputLstJSON=%5B%5D

List Mutation

Create a new list:

List literal: `s = [1, 2, 3]`

List constructor: `t = list(s)`

List comprehension: `u = [x for x in s]`

+: `v = s + t`

Slicing: `w = t[1:]`

Modify a list:

```
s.append(4)

s[1] = 4

s.extend(t) # Modifies s but not t

s.remove(2) # Removes the first 2

s.pop() # Removes the last element
```

(Demo)

https://pythontutor.com/cp/composingprograms.html#code=s%20%3D%20%5B%5B1,%202%5D,%203%5D%0At%20%3D%20s%5B0%5D%0As%5B0%5D%5B1%5D%20%3D%204%0As.append%28t%29%0At.append%285%29%0Aprint%28s%29&cumulative=true&curInstr=0&mode=display&origin=composingprograms.js&py=3&rawInputLstJSON=%5B%5D

Sum more Fun (Building Lists with Append)

```
def sums(n, m):
    """Return lists that sum to n containing positive numbers up to m that
    have no adjacent repeats, for n > 0 and m > 0.

    >>> sums(5, 1)
    []
    >>> sums(5, 2)
    [[2, 1, 2]]
    >>> sums(5, 3)
    [[1, 3, 1], [2, 1, 2], [2, 3], [3, 2]]
    >>> sums(5, 5)
    [[1, 3, 1], [1, 4], [2, 1, 2], [2, 3], [3, 2], [4, 1], [5]]
    >>> sums(6, 3)
    [[1, 2, 1, 2], [1, 2, 3], [1, 3, 2], [2, 1, 2, 1], [2, 1, 3], [2, 3, 1], [3, 1, 2], [3, 2, 1]]
    """
    result = []
    for k in range(1, ___min(m + 1, n)___): # k is the first number of a list
        for rest in ___sums(n-k, m)___:
            if rest[0] != k:
                result.append(_[k] + rest_)  # build a list out of k and rest
    if n <= m:
        result.append([n])
    return result
```

What are all of the ways to build the rest of the list, now that we've used a 1? Figure out how to sum to 4

Start with a 1

https://pythontutor.com/cp/composingprograms.html#code=result%20%3D%20%5B%5B1%5D,%20%5B2%5D,%20%5B3%5D%5D%0Afor%20k%20in%20range%281,%204%29%3A%0A%20%20%20%20for%20s%20in%20result%3A%0A%20%20%20%20%20%20%20%20if%20s%5B-1%5D%20!%3D%20k%3A%0A%20%20%20%20%20%20%20%20%20%20%20%20s.append%28k%29%0A%20%20%20%20%20%20%20%20%20%20%20%20result.append%28s%29%0Aprint%28result%29&cumulative=true&curInstr=0&mode=display&origin=composingprograms.js&py=3&rawInputLstJSON=%5B%5D

Mutation and Identity

Identity Operators

Identity

`<exp0> is <exp1>`

evaluates to `True` if both `<exp0>` and `<exp1>` evaluate to the same object

Equality

`<exp0> == <exp1>`

evaluates to `True` if both `<exp0>` and `<exp1>` evaluate to equal values

Identical objects are always equal values

(Demo)

Spring 2023 Midterm 2 Question 1

pop() removes the last element and returns it

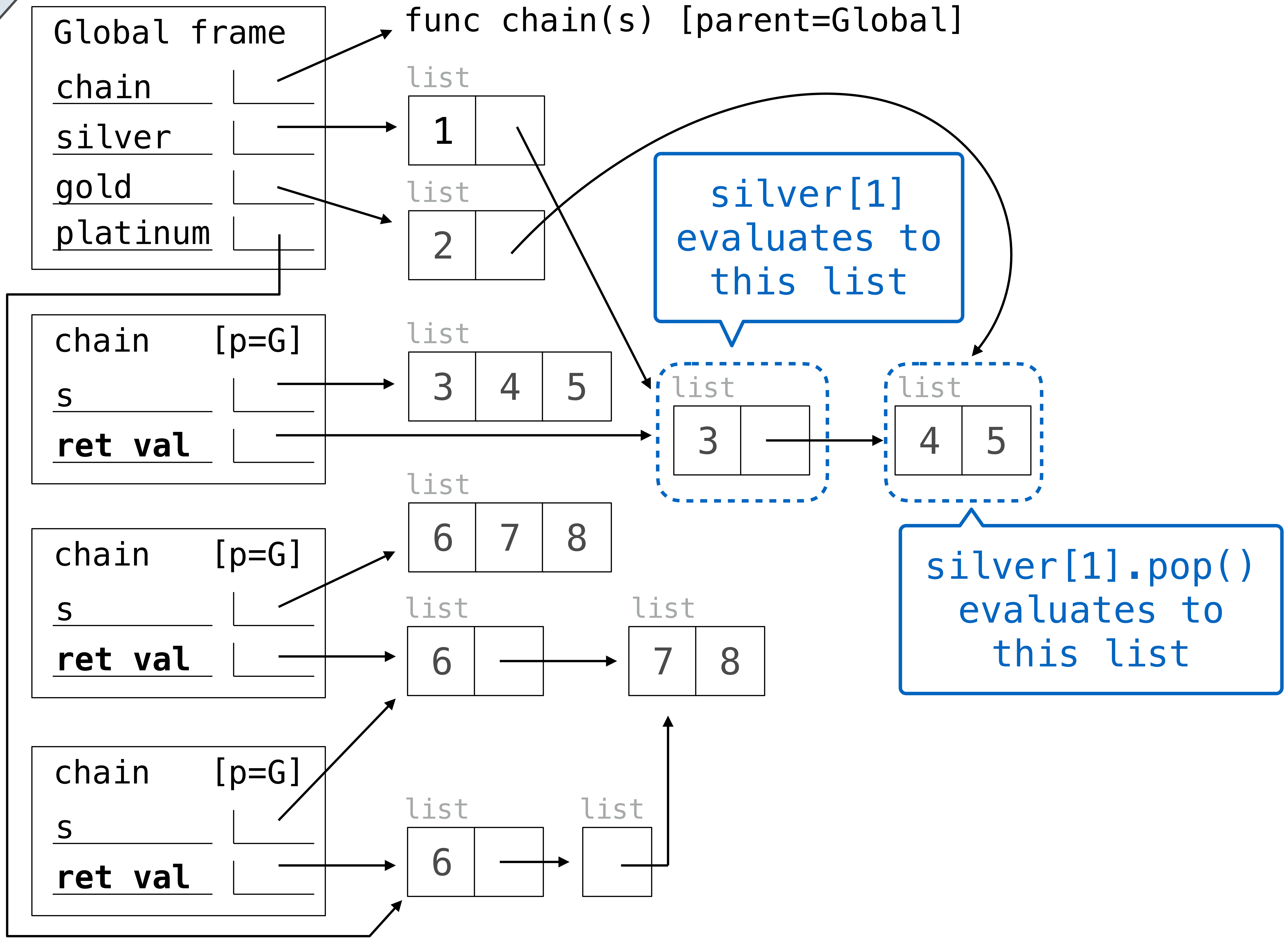
```
def chain(s):
    return [s[0], s[1:]]
silver = [2, chain([3, 4, 5])]
gold = [silver[0], silver[1].pop()]
silver[0] = 1
platinum = chain(chain([6, 7, 8]))
```

Reminder: s.pop() removes and returns the last item in list s.

```
>>> silver
[1, [3]]

>>> gold
[2, [4, 5]]

>>> platinum
[6, [[7, 8]]]
```



silver[1] evaluates to this list

silver[1].pop() evaluates to this list

https://pythontutor.com/cp/composingprograms.html#code=def%20chain%28s%29%3A%0A%20%20%20%20return%20%5Bs%5B0%5D,%20s%5B1%3A%5D%5D%0Asilver%20%3D%20%5B2,%20chain%28%5B3,%204,%205%5D%29%5D%0Agold%20%3D%20%5Bsilver%5B0%5D,%20silver%5B1%5D.pop%28%29%5D%0Asilver%5B0%5D%20%3D%201%0Aplatinum%20%3D%20chain%28chain%28%5B6,%207,%208%5D%29%29%0A&cumulative=true&curInstr=0&mode=display&origin=composingprograms.js&py=3&rawInputLstJSON=%5B%5D

Mutation and Names

If multiple names refer to the same mutable object (directly or indirectly), then a change to that object is reflected in the value of all of these names.

What numbers are printed (and how many of them)?

```
s = [2, 7, [1, 8]]
t = s[2]
t.append([2])
e = s + t
t[2].append(8)
print(e)
```

https://pythontutor.com/cp/composingprograms.html#code=s%20%3D%20%5B2,%207,%20%5B1,%208%5D%5D%0At%20%3D%20s%5B2%5D%0At.append%28%5B2%5D%29%0Ae%20%3D%20s%20%2B%20t%0At%5B2%5D.append%288%29%0Aprint%28e%29%0A&cumulative=true&curInstr=6&mode=display&origin=composingprograms.js&py=3&rawInputLstJSON=%5B%5D