

loops, or y for yes and N for No
 Put True (T) or False (F) in every cell, based on characteristics of each type. do you need to run pip install ... ?
 do you need to use import ... ?

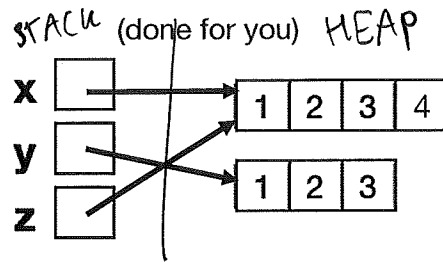
1

Data Type	Mutable?	Pre-installed?	Builtin?	Create New Types?	Named Attributes?
list	y	y	T	N	N
tuple	N	y	T	N	N
namedtuple	N	y	F	y	y, eg

page

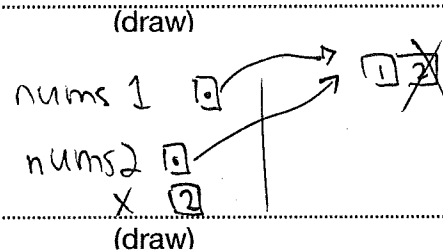
2

```
x = [1, 2, 3]
y = [1, 2, 3]
z = x
z.append(4)
```



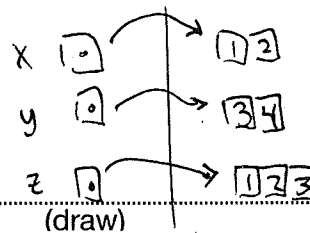
3

```
nums1 = [1, 2]
nums2 = nums1
x = nums2.pop(1)
```



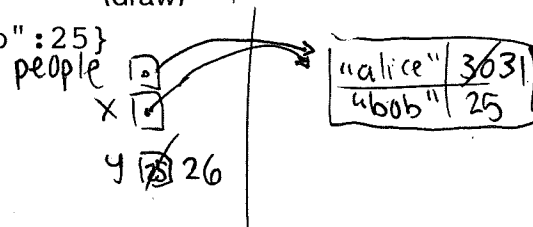
4

```
x = [1, 2]
y = [3] + always creates new instance
z = x + y
y.append(4)
```



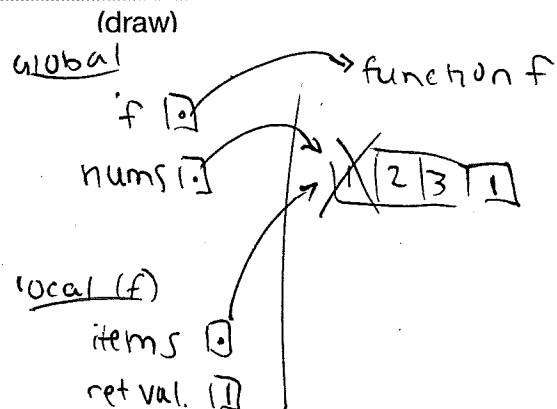
5

```
people = {"alice": 30, "bob": 25}
x = people
y = people["bob"]
x["alice"] = 31
y = 26
```



6

```
def f(items):
    return items.pop(0)
nums = [1, 2, 3]
nums.append(f(nums))
```

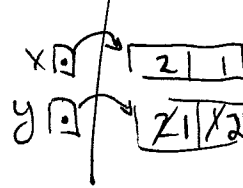


Remember to import copy for these in Python Tutor!

7

```
x = [2,1]
y = copy.copy(y)
y.sort()
```

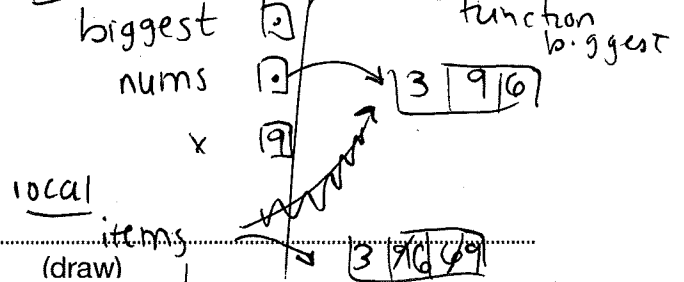
(draw)



8

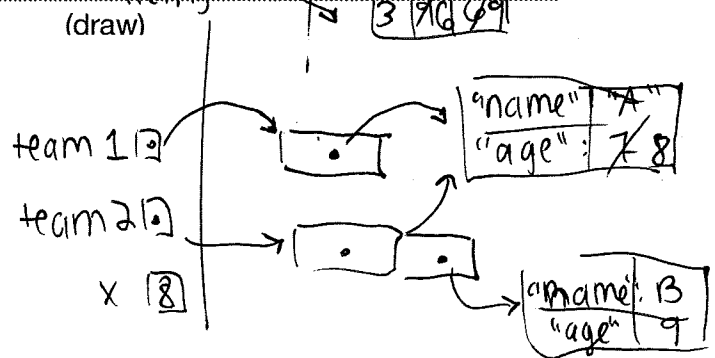
```
def biggest(items):
    items = copy.copy(items)
    items.sort()
    return items[-1]
nums = [3,9,6]
x = biggest(nums)
```

(draw)



9

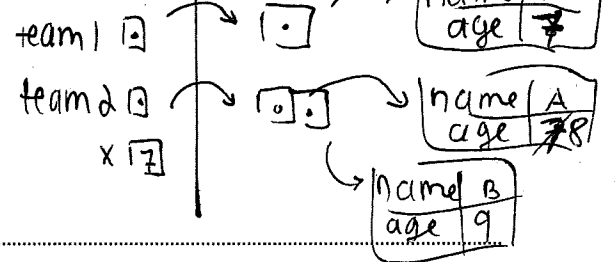
```
team1 = [
    {"name": "A", "age": 7}
]
team2 = copy.copy(team1)
team2.append(
    {"name": "B", "age": 9}
)
team2[0]["age"] = 8
x = team1[0]["age"]
```



10

Same as above, but with
copy.deepcopy(...) instead
of copy.copy(...).

(draw)



11

```
orig = [1,[2,[3,4]]]
x = orig
y = copy.copy(orig)
z = copy.deepcopy(orig)
```

(draw)

