Put True (T) or False (F) in every cell, based on characteristics of each type.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Data Type** | **Mutable?** | **Pre-installed?** | **Builtin?** | **Create New Types?** | **Named Attributes?** |
| list |  |  |  |  |  |
| tuple |  |  |  |  |  |
| namedtuple |  |  |  |  |  |
| recordclass |  |  |  |  |  |

(done for you)

**x**

**y**

**z**

1

2

3

1

2

3

4

x = [1, 2, 3]

y = [1, 2, 3]

z = x

z.append(4)

(draw)

nums1 = [1,2]

nums2 = nums1

x = nums2.pop(1)

(draw)

x = [1, 2]

y = [3]

z = x + y

y.append(4)

(draw)

people = {"alice":30, "bob":25}

x = people

y = people["bob"]

x["alice"] = 31

y = 26

(draw)

def f(items):

return items.pop(0)

nums = [1,2,3]

nums.append(f(nums))

Remember to import copy for these in Python Tutor!

x = [2,1]

(draw)

y = copy.copy(y)

y.sort()

(draw)

def biggest(items):

items = copy.copy(items)

items.sort()

return items[-1]

nums = [3,9,6]

x = biggest(nums)

(draw)

team1 = [

{"name":"A", "age":7}

]

team2 = copy.copy(team1)

team2.append(

{"name":"B", "age":9}

)

team2[0]["age"] = 8

x = team1[0]["age"]

(draw)

Same as above, but with

copy.deepcopy(...) instead

of copy.copy(...).

(draw)

orig = [1,[2,[3,4]]]

x = orig

y = copy.copy(orig)

z = copy.deepcopy(orig)