Name: Simran Sharma

When in 3 if x = x.sort() it will return None but if just x.sort() then it x = sorted list they returns sorted list

Code-Drawing Template

inplace

Code	Variables	Objects
x = [1,2,3,4] y = x x.pop(1)	x = [1,3,4] y = [1,3,4]	x ⟶ [1,3,4] y ⟶

Code	Variables	Objects
x = [1,4,3,2] y = x sorted(x)	x = [1,4,3,2] y = [1,4,3,2]	x ⟶ [1,4,3,2] y ⟶ sorted(x) needs to be reassigned to a new variable to be new list

x = x.sort() or just x.sort() means x = 'None' y = is new sorted list

Code	Variables	Objects
x = [1,4,3,2] y = x x = x.sort()	x = 'None' y = [1,2,3,4]	x ⟶ None y ⟶ [1,2,3,4]

Code	Variables	Objects
x = [3,14,42) z = {'boring #': [1,2,3], 'interesting': x} y = x y.append(1.618)	x = [3.14,42,1.618] y = [3.14,42,1.618] z = {'boring': [1,2,3], 'interesting': [3.14,42,1.618]}	x → [3.14,42,1.618] y ↗ z → "boring" | "[1,2,3]" "interesting" | " "

Code	Variables	Objects
x = [42,-1] y = (1,2,x) x.append(-2)	x = [42,-1] y = (1,2,x)	x → [42,-1,-2] y → (1,2,[42,-1,-2])

Exercise 2:

x = [1,4,3,2]
y = x
sorted(x) → But shows sorted list
↳ this leaves x and y as unsorted
x → [1,4,3,2] y = [1,4,3,2]

x = [1,4,3,2]
y = x → still pointing to old object
x = sorted(x) → creates new object
↳ gives
x = [1,2,3,4]
y = [1,4,3,2]

Name: Simrun Sharma

When in 3 if x = x.sort()
it will return None, but

Code	Variables	Objects
y['dino'] = ['big guy', 'lizard'] y['cat_name'].append('tiger')	x = {'dog': ('fido', 'waster'] y = {'dog_names': "fido, 'waster"}	X['dinosaur_names'] = ['big guy', 'super lizard'] X['cat_names'] = ['fluffy', 'bite-y', 'tiger']
Code	Variables	Objects
Code	Variables	Objects
Code	Variables	Objects
Code	Variables	Objects
Code	Variables	Objects
Code	Variables	Objects

'import copy
y = copy.deepcopy(x)
means x and y point to
two different objects