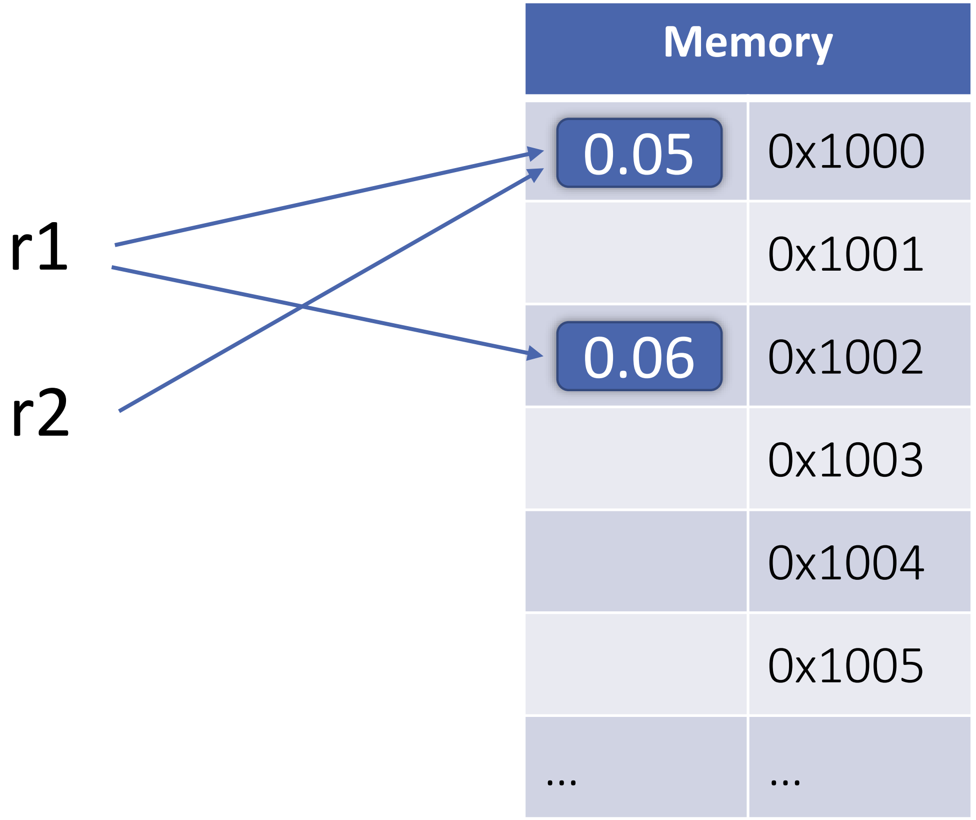
(Im-)Mutability of Integers/Floats



Memory	
0.05	0x1000
	0x1001
0.06	0x1002
	0x1003
	0x1004
	0x1005
…	…

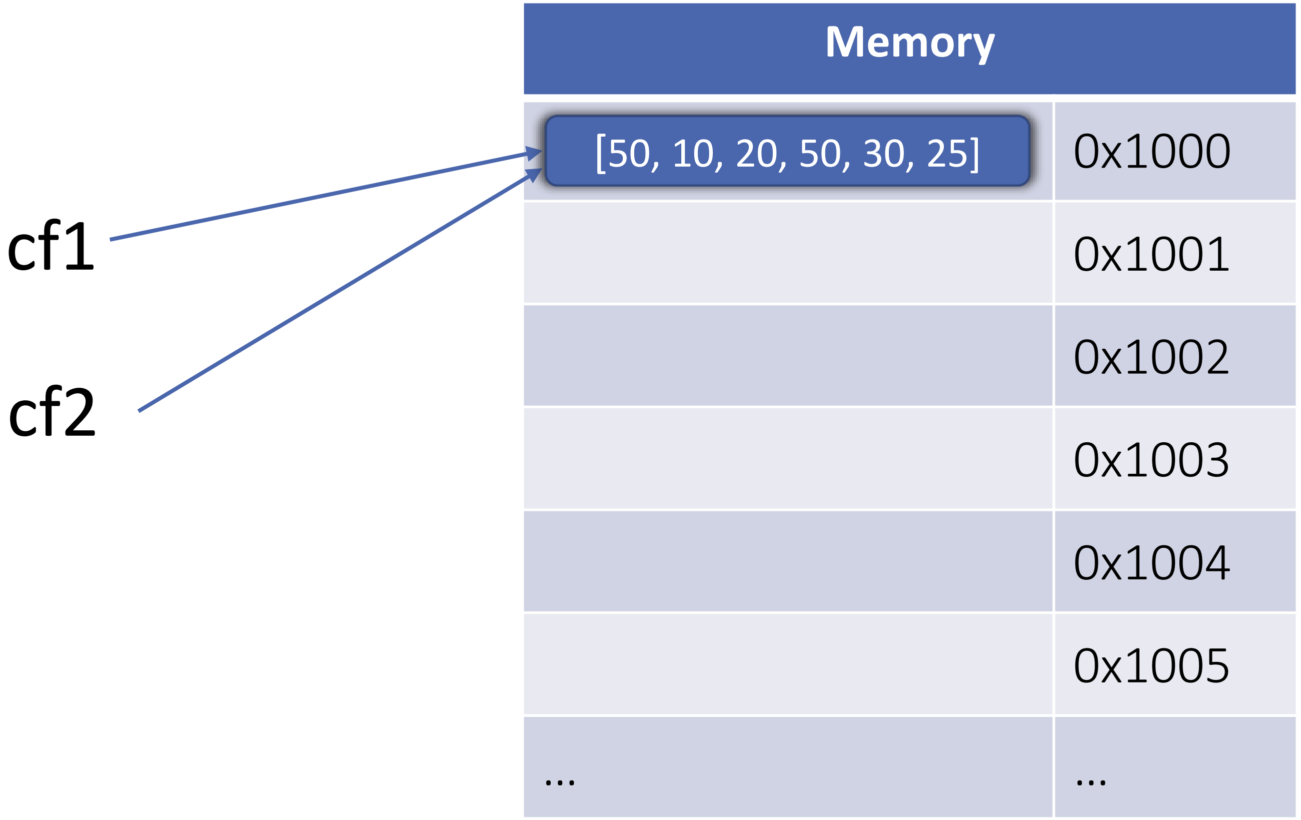
Code: r1 = 0.05

Code: r2 = r1

Code: r1 += 0.01

Integers/Floats are immutable Objects!

Mutability of Lists



Memory	
[50, 10, 20, 50, 30, 25]	0x1000
	0x1001
	0x1002
	0x1003
	0x1004
	0x1005
…	…

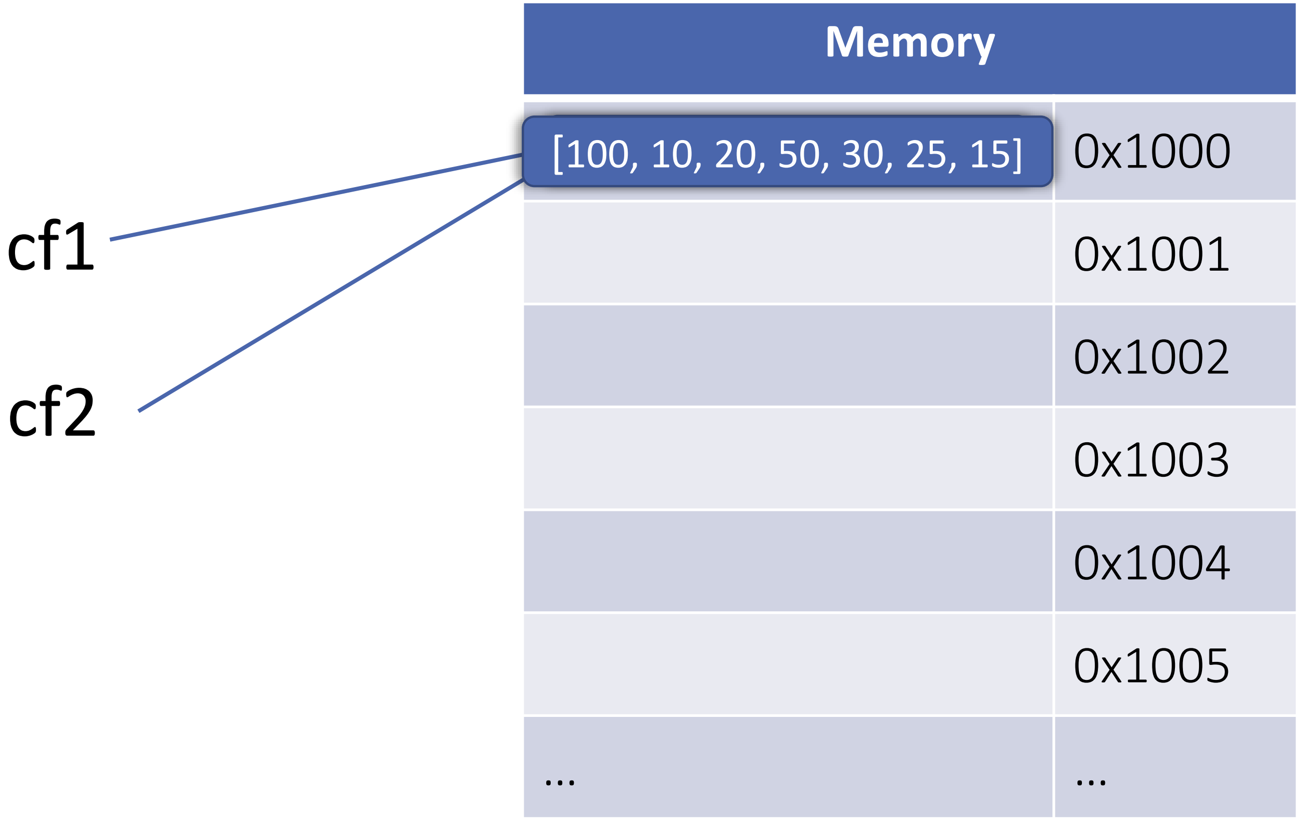
Code: cf1 = [100, 10, 20, 50, 30, 25]

Code: cf2 = cf1

Code: cf1[0] = 50

Lists are mutable Objects!

Changing / Mutating Lists “In Place” vs…



Memory	
[100, 10, 20, 50, 30, 25, 15]	0x1000
	0x1001
	0x1002
	0x1003
	0x1004
	0x1005
…	…

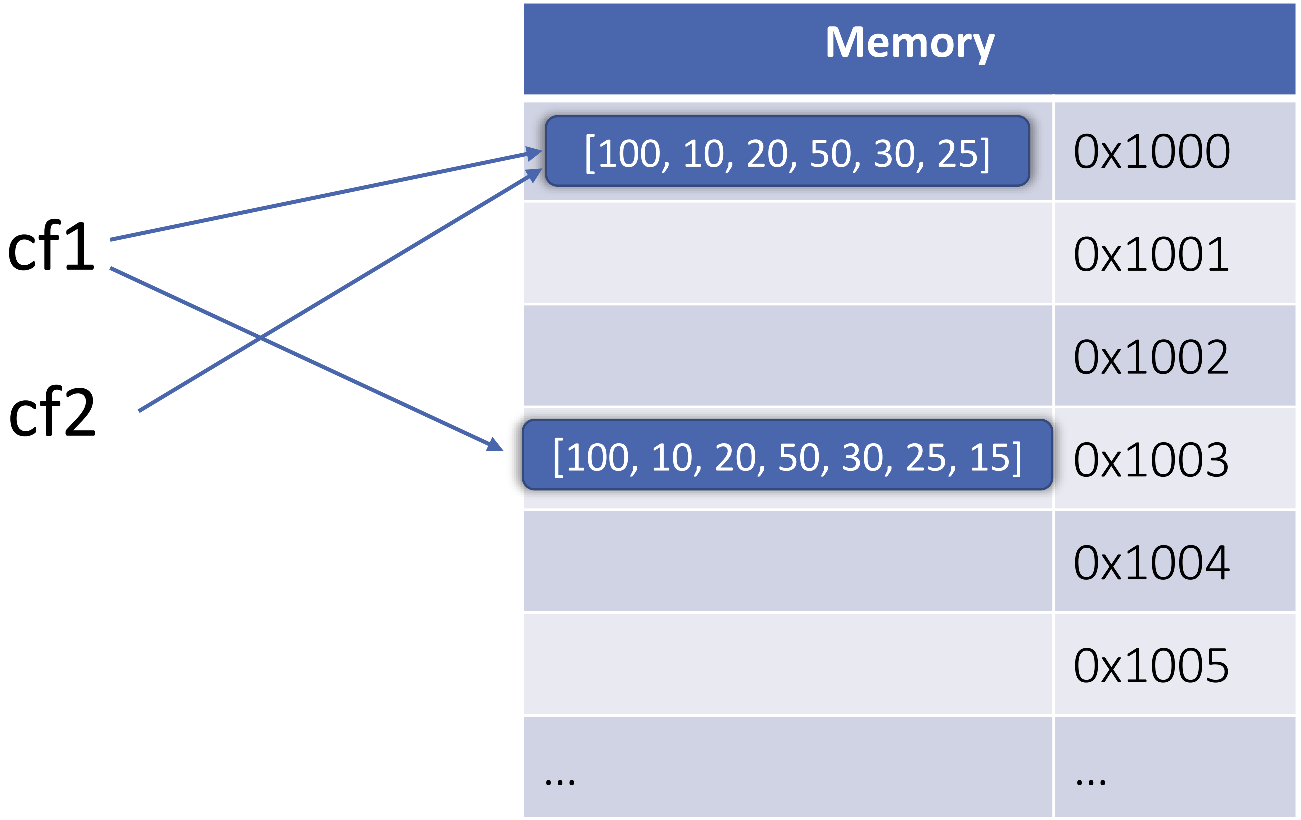
Code: cf1 = [100, 10, 20, 50, 30, 25]

Code: cf2 = cf1

Code: cf1.append(15)

List was modified/changed.

…Variable Re-Assignment



Memory	
[100, 10, 20, 50, 30, 25]	0x1000
	0x1001
	0x1002
[100, 10, 20, 50, 30, 25, 15]	0x1003
	0x1004
	0x1005
…	…

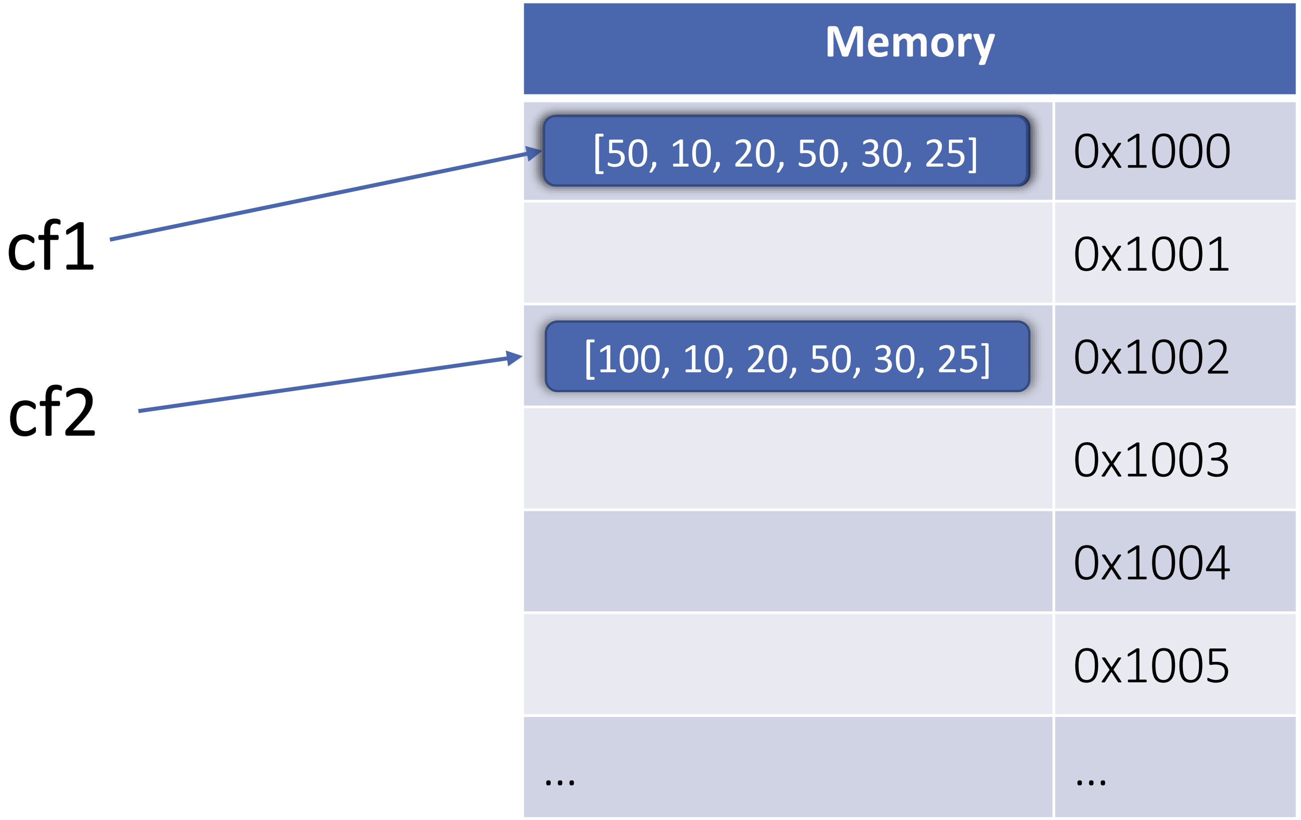
Code: cf1 = [100, 10, 20, 50, 30, 25]

Code: cf2 = cf1

Code: cf1 += [15]

Original List wasn´t modified/changed.

Creating Copies (non-advisable Solution)



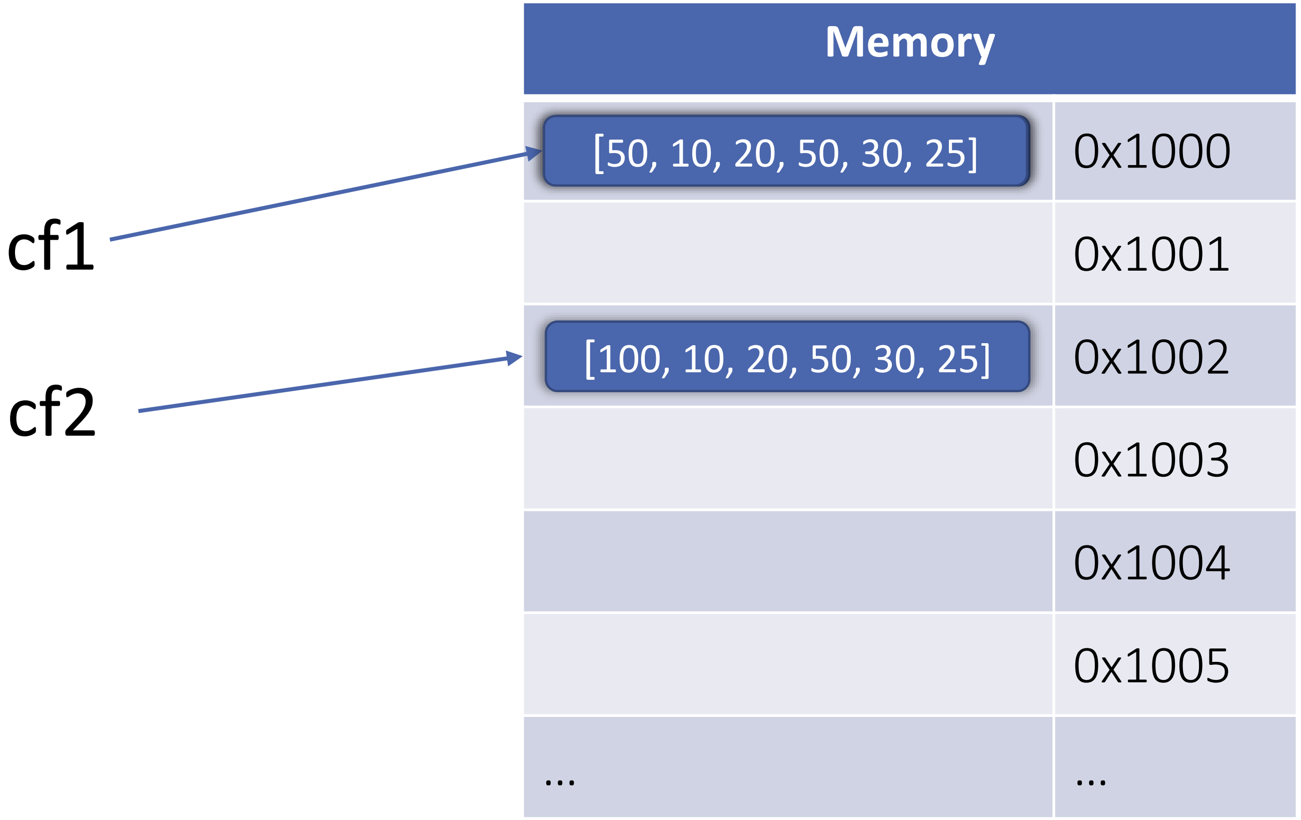
Memory	
[50, 10, 20, 50, 30, 25]	0x1000
	0x1001
[100, 10, 20, 50, 30, 25]	0x1002
	0x1003
	0x1004
	0x1005
…	…

Code: cf1 = [100, 10, 20, 50, 30, 25]

Code: cf2 = [100, 10, 20, 50, 30, 25]

Code: cf1[0] = 50

Creating Copies (non-advisable Solution)



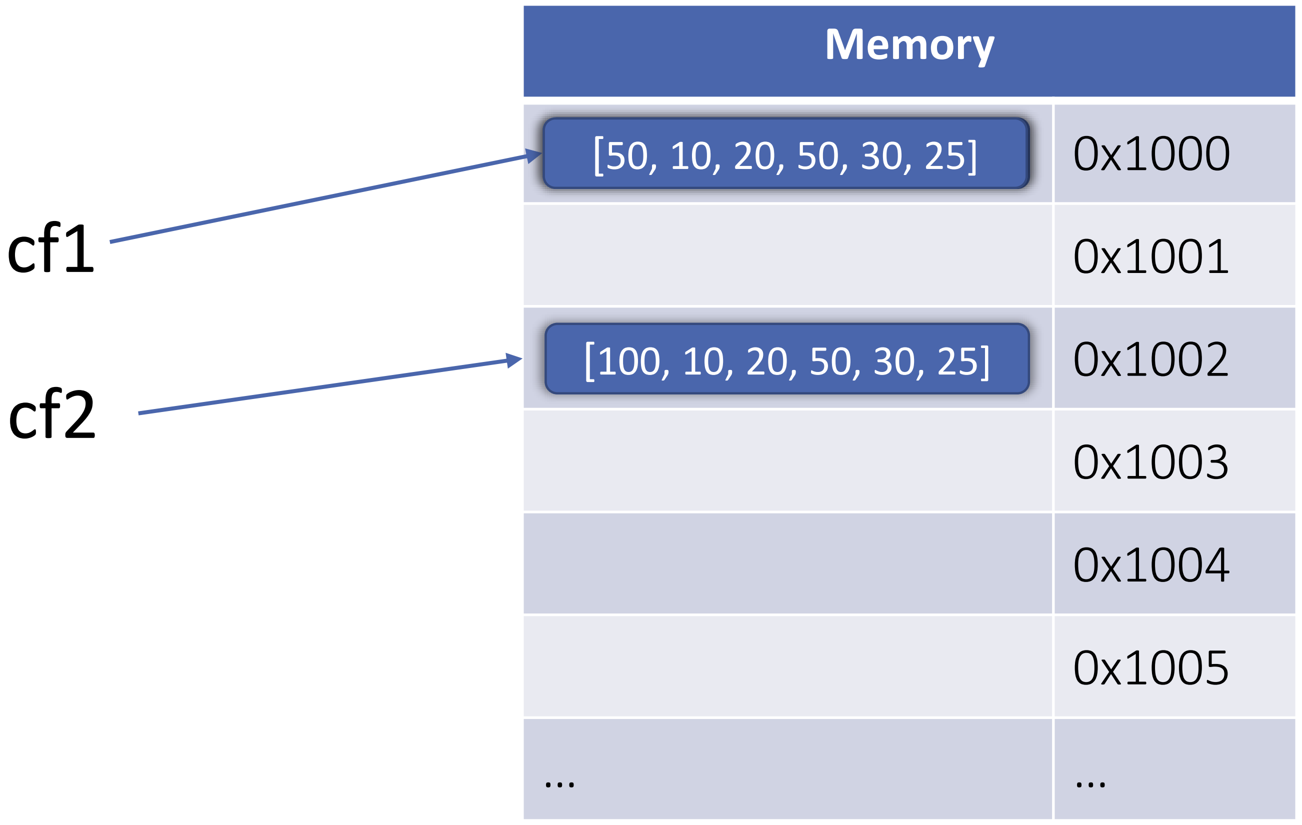
Memory	
[50, 10, 20, 50, 30, 25]	0x1000
	0x1001
[100, 10, 20, 50, 30, 25]	0x1002
	0x1003
	0x1004
	0x1005
…	…

Code: cf1 = [100, 10, 20, 50, 30, 25]

Code: cf2 = cf1[:]

Code: cf1[0] = 50

Creating Copies (advisable Solution)



Memory	
[50, 10, 20, 50, 30, 25]	0x1000
	0x1001
[100, 10, 20, 50, 30, 25]	0x1002
	0x1003
	0x1004
	0x1005
…	…

Code: cf1 = [100, 10, 20, 50, 30, 25]

Code: cf2 = cf1.copy()

Code: cf1[0] = 50