

chap 10 : lists

⁰ ¹ ²
L = ['one', 'two', 'three']

L[0] # 'one'

a list index can be any int expression, but not float.

L[0.0] # error.

To traverse a list:

```
for i in range(len(L)):
    print(L[i])
```

more Pythonic:

```
for x in L:
```

```
    print(x)
```

in & not in operators:

'one' in L $\neq$ True

'happy' in L $\neq$ False

'happy' not in L $\neq$ True.

we can concatenate lists with + :

Ex. A = [1, 2, 3]

B = [4, 5, 6]

C = A + B $\neq$ [1, 2, 3, 4, 5, 6]

Recall: $3 * \text{'happy'} \neq \text{'happyhappyhappy'}$

same with lists

$$D = 3 * A \neq [1, 2, 3, 1, 2, 3, 1, 2, 3]$$

0 1 2 3 4 5 6 7 8

$$E = A * 3 \neq \text{same}$$

Slices: work on lists

$$D[3:7] \neq [1, 2, 3, 1]$$

$$D[:7] \neq [1, 2, 3, 1, 2, 3, 1]$$

$$D[3:] \neq [1, 2, 3, 1, 2, 3]$$

$$D[:] \neq \text{copy of } D.$$

Recall: lists are mutable.

$$ID[2] = 5 \neq [1, 2, 5, 1, 2, 3, 1, 2, 3]$$

we can alter $\mathcal{D}$ by using slices

$$\triangleright [3:7] = []$$

now $D \rightarrow$ $\begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 5 & 2 & 3 \end{bmatrix}$
 $D[1:4]$

$$\triangleright [1:4] = ['one', 'two']$$

now $\rightarrow$ it $\left[\begin{matrix} 0 & 1 & 2 & 3 \\ 1, & \text{'one'}, & \text{'two'}, & 3 \end{matrix} \right]$

del operator:

del ID[1] # [1, 'two', 3]

E = 3*[1, 2, 3] # [1, 2, 3, 1, 2, 3, 1, 2, 3]

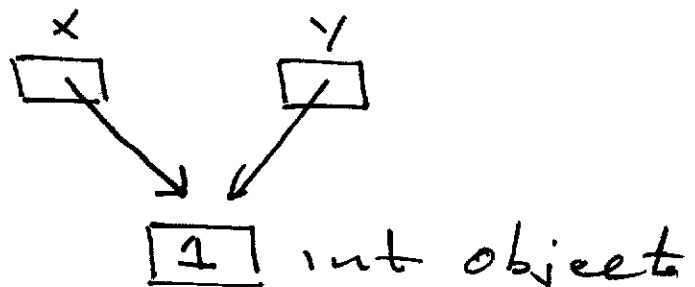
del E[1:5] # [1, 3, 1, 2, 3]

Objects

The is operator determines if two objects are one and the same

x = 1

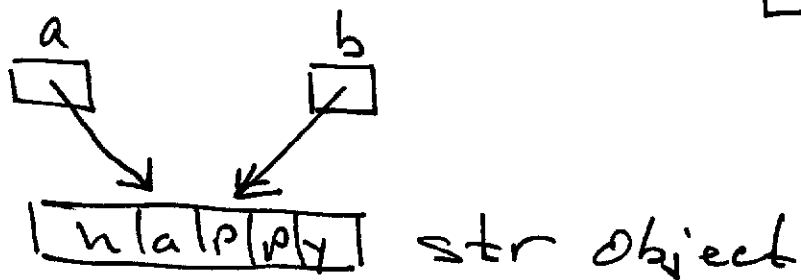
y = 1



x is y # True

a = 'happy'

b = 'happy'



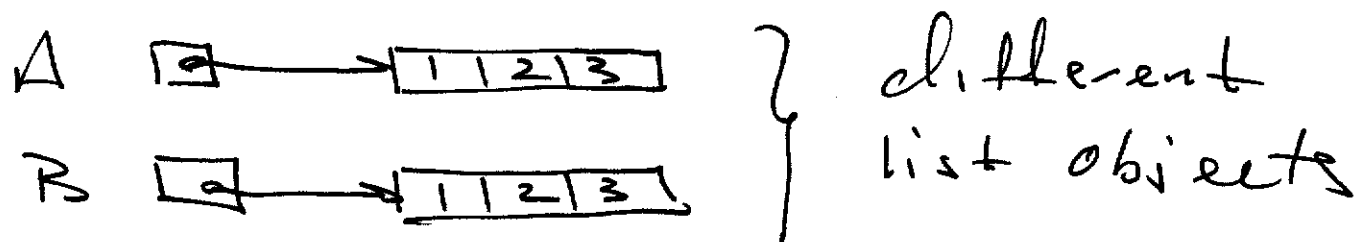
a is b $\neq$ True

A = [1, 2, 3]

B = [1, 2, 3]

A == B $\neq$ True

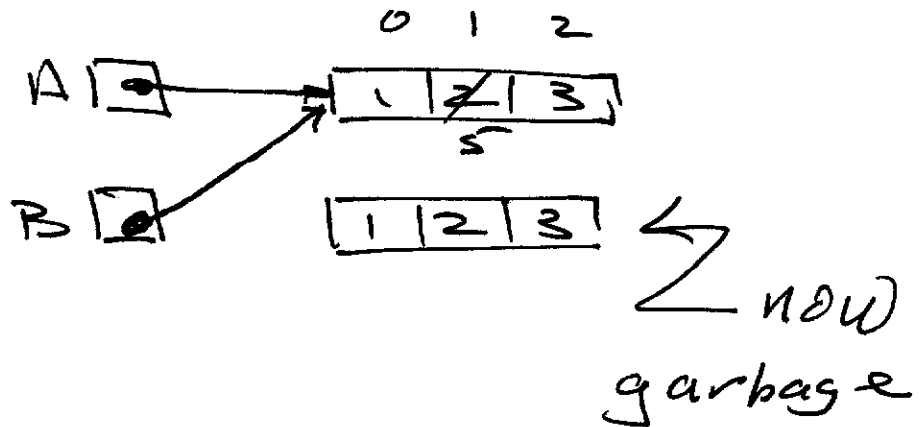
A is B $\neq$ False



Recall: list is mutable
 str is immutable
 int " "

The assignment operator = creates an alias

B = A



A[1] = 5

Print(B) $\neq$ [1, 5, 3]

Thus

- == compares the contents or data stored in objects
- is compares the addresses of objects.

note:

for any variables u, v !

if u is v is true,
then $u == v$ is true

But the converse may be false.

we could have $u == v$ true
but u is v false.

To clone an object (i.e. a copy with same data.)

$$A = [1, 2, 3]$$

$$B = \text{list}(A) \neq \text{copy}$$

$$B = A[:] \neq \text{also a copy}$$

Ex. ReverseList.py

ReverseSequence.py

Built-in list funcs

- append() ✓
- insert() ✓
- count() ✓
- reverse() ✓
- Pop() ✓
- remove() ✓
- index() ✓

Suppose we want to square everything in a list.

Say $L = [1, 2, 3, 4]$

for i in range(len(L)):

$L[i] = L[i]**2$

#end

now: $L = [1, 4, 9, 16]$

or

for (i, x) in enumerate(L)

$L[i] = x**2$

#end.

or (list comprehension)

$$L = [x ** 2 \text{ for } x \text{ in } L]$$

Ex. $L = [3, 2, 1, 4, 8, 5, 9, 6, 7]$

$$L = [x ** 2 \text{ for } x \text{ in } L \text{ if } x \% 2 == 0]$$

Nested lists

matrices:

$$M = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

3×3
 $\nearrow$ $\nwarrow$
 #rows #columns

in Python:

$[[1, 2, 3], [4, 5, 6], [7, 8, 9]]$

Ex. MatrixAdd.py

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}_{2 \times 2} + \begin{pmatrix} e & f \\ g & h \end{pmatrix}_{2 \times 2} = \begin{pmatrix} a+e & b+f \\ c+g & d+h \end{pmatrix}$$