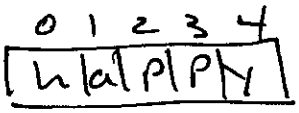
CSE 20 11-13-20

---

lab 3 : ext. 1 day

more on strings (chap 8):

strings & lists can be manipulated using slices:

s = 'happy' s → 

s[n:m] # substring from index n (inclusive) to index m (exclusive).

s[1:4] # 'app'

s[2:5] # 'ppy'

s[2:] # 'ppy

so

s[n:] # from n (incl) to end

also

s[0:3] # 'hap'

s[:3] # 'hap'

so

s[:m] # beginning of str to s[m-1]

also

s[:] # a copy of s

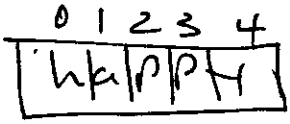
<u>comparing strings</u> : t, s

comparison ops: t==s, t!=s, t<s,
(alpha order) t<=s, t>s, t>=s

Note:

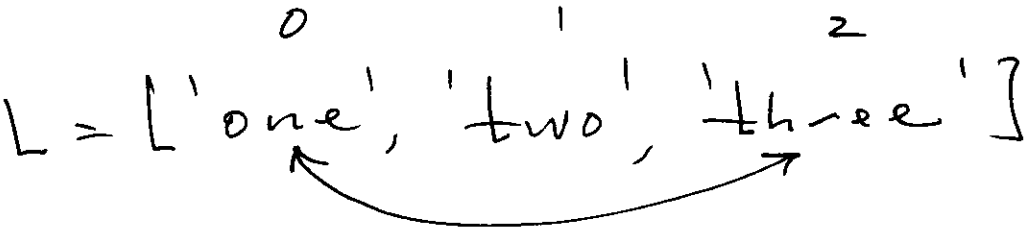
- strings are immutable (i.e. cannot be changed.
- lists are mutable.

Ex. s = 'happy'



s[1] = 'o' # error

Can alter lists



temp = L[0]
L[0] = L[2]
L[2] = temp

The in operator tests for membership.

obj1 in obj2

obj2 must be a collection list, strings, ...

True if obj1 is mem. of obj2

False if " is not mem. of obj2

Ex. S = 'happy' L = ['one', 'two', 'three']

'p' in S # True

'q' in S # False

'one' in L # True

S in L # False

The not in operator:

'p' not in s # False

'q' not in s # True

Same as: not ('p' in s)

Ex. RemoveVowels.py

Ex. FindInString.py

note: str has a fcn called find()

Exercise

write a function

myFind(s,t)

that emulates s.find(t)

built in to str type.

split() fcn in str

```
r = 'one two three four'
L = r.split()
```

join() fcn in str

```
t = ' '.join(L)
u = ':'.join(L)
```

also: upper(), lower(), title(), swapcase().

format() function

Ex. `'2+3={0}, 2*3={1}'.format(2+3, 2*3)`

0 = 0th arg. 1 = 1st arg.