# 9/9 CSP(L) notes:

### S:

* Only one partner needs to turn in Labs
* Take turns for each problem with typing
* If cant figure out problem put pass back in so it could load into python
* When upload to gradescope make sure to select partners name
* **Tuple no comma is not tuple**

Lists: [1, 3, 5] [0.01, true, false,’my house’]

[] empty list of elements

[(1)] one element containing a list of 1 element

* Lists differ semantically from tuples in that they are mutable. **Strings and tuples are immutable.**
* Strings tuples and lists share many operations(like + and \*) includijng indexing and other explicit operations
* >>> len(“this is a test”) 14
* >>> “this is a test”[11] “e”
* >>> len(1, 2, 3) 3
* (1, 2, 3)[2] 3 #zero indexing
* len([]) 0
* [][0] #error list of 0 elements
* >>> [1,3,5,7] [-2] 5 #index from the right sade, -1 first element
* >>> [1, 3, 5, 7] [1: -2] [3] #returns a sublist of one element
* >>> “this is a test”[4:8] #starts 4 but doesnt include or go past 8 ‘is’
* “This is a test” [4:] ‘is a test’
* >>> “this is a test”[:8] ‘this is’
* Strings are immutable ordered sequences of characters
* Tuples are immutable ordered sequences of arbitrary items including other tuples,strings, lists
* Lists are mutable ordered sequences of arbitiarty items including other lists tuples and strings
* All sequence typers share indexing and length operations
* >>> "famous"[3] + "forge"[5//2]

'or'

* (‘R’, + (“tail”, “frame”) \* len([-7, 8, 2][:2])
* R a f, ‘tail’ frame’ tail frame
* >>> (False, False, True, True) [-3:3] (false, True) #tuple
* ‘p’ in “once upon a time”[3:11][-6:6][:2] “e upon a“ True #boolean
* “once upon a time”[3:11] “e upon a”
* “once upon a time”[-6:6] “upon”

### Ranges:

* A range is an implicit representation of an immutable sequence of integers
* >>> range(5) range(0,5)
* >>> range(5)[3] 3
* >>>range(1,11, 2) range(1,11, 2)
* >>> range(1, 11, 2)[-2:5][-1] 9
* >>> list(tuple(range(0,0)))
* In general range(start, stop, step): range(start, stop) assumes step is 1 while range(stop) assumes start is 0
* >>> range(3, 30, 3) range(3, 30, 3)
* len(range(3,30,3)) 9
* Slicing can be even more complicated consider S=’0123456789’
* >>. S(1:8:2] “1357”
* S[:] ‘0123456789’ #copy
* S[::-1] #reversedcopy
* S[1::-1] ‘10’ #from 1 down reversed
* S[:-3:-1] ‘98’

### 