

Par: ext... 3 days

lab4: ext. 1 day

Ex. MatrixAdd.py

Exercise

consider a list to represent a vector

$[2, 5, 3]$ represents $(2, 5, 3)$

or $2i + 5j + 3k$

write functions that

- add 2 lists (as vectors)
 - multiply 2 lists (componentwise multiplication)
 - scalar multiply number x list
- $\text{scalarMult}(3, [1, 4, 7]) \neq [3, 12, 21]$

• dot Product: list $\times$ list $\rightarrow$ number

$$\text{dot}([1, 4, 7], [2, -3, 1]) \# 1 \cdot 2 + 4(-3) + 7 \cdot 1$$

$$\# \boxed{-3}$$

chapter 11 : Dictionaries

A Dictionary is similar to a list

$L = []$ or $L = \text{list}()$ $\#$ empty list

$D = \{ \}$ or $D = \text{dict}()$ $\#$ empty dictionary.

Both lists and Dictionaries are containers (like tuples), operators

$\left\{ \begin{array}{l} \text{in} \\ \text{not in} \end{array} \right.$

work as in lists (and tuples).

The difference is that in a dictionary indices don't have to be ints. The indices in a dictionary (now called keys) can be any immutable type

$D['foo'] = 'bar'$

$D['happy'] = 'sad'$

$D[6] = 'blah'$

$D[(1,2)] = [1,2,3]$

Both lists & dictionaries can be thought of as mappings.

0 1 2

$L = [1, 2, 3]$

<u>index</u>		<u>value</u>
0	→	1
1	→	2
2	→	3

$D = \{ 'uno': 'one', 'dos': 'two', 'tres': 'three' \}$

<u>key</u>	<u>value</u>
uno	→ one
dos	→ two
tres	→ three

Print(L) # [1, 2, 3]

Print(D) # { 'uno': 'one', 'dos': 'two', 'tres': 'three' }

for key in D:

 Print(key, D[key])

#end for

output :

uno one

dos two

tres three

Ex CountCharacters.py

countChar('happy')

returns

{ 'h': 1, 'a': 1, 'p': 2, 'y': 1 }