

# CSC 101

Today

- List Comprehensions

- map
- filter

- Test Cases for Assignment #3

list = [7, 9, 2]

~~List[7, 9, 2]~~

---

class Point:

- -  
[ Point(2, 3), Point(4, 7) ] point objects

[ (2, 3), (4, 7) ] tuples  
↑  
tuple

Math

(1, 2)  
(1, 2)

# Tuples

- compound data

$(2, 3)$  tuple

$t = (2, 3)$

$t[0]$

triple =  $(3, \text{True}, "s")$

representation

|   |   |
|---|---|
| 0 | 1 |
| 2 | 3 |

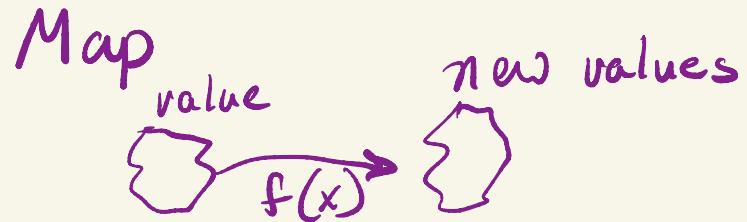
import typing

from typing import Tuple

triple : Tuple[int, bool, str] = (2, True, "S")

x = triple[1]

$L1 = [7, 9, 2]$



$[f(x) \text{ for } x \text{ in list}]$

$L2 = [x + 2 \text{ for } x \text{ in } L1]$

$L2 = \underset{\text{map}}{\underbrace{[9, 11, 4]}}$

Diagram showing the mapping process:

- Original list:  $[7, 9, 2]$
- Mapping function:  $x + 2$
- Resulting list:  $[9, 11, 4]$

Annotations:

- $x = 7$  above 9
- $x = 9$  above 11
- $x = 2$  above 4
- $x + 2$  written vertically below the mapping arrow

pattern = " X O X  
          O X O  
      "  
      "

[ [ 'X', 'O', 'X' ],  
  [ 'O', 'X', 'O' ] ]

Math       $x = y + 3$

Code

known  
 $y = 27$

$x = y + 3$

$x = 5$

$y = 27$

## List Comprehension

Computes for a list of  
Points the translation of  
each 1 position in the  
positive x direction



---

points = [Point(2, 9, 3), Point(4, 1, 0)]  
dir = Vector(1, 0, 0)

new-points = [ dir ↑ for point in points ]  
[Vector(1, 0, 0), Vector(1, 0, 0)]

&gt;

points = [Point(2, 9, 3), Point(4, 1, 0)]

dir = Vector(1, 0, 0)

new-points = [translate-point(point, dir)

for point in points]

Result

new-points → [Point(3, 9, 3),  
Point(5, 1, 0)]

ticket-price

given age returns price

---

$$\text{ages} = [9, 14, 27, 3]$$

prices = [ticket-price(age) for age in ages]

$$[12, 15, \overset{\leftarrow}{15}, \overset{\rightarrow}{12}]$$

under 10 prices

ages = [9, 14, 27, 3]

prices = [ (age, ticket-price(age))  
for age in ages ]

[ (9, 12), (14, 15), (27, 15),  
(3, 12) ]

`ages = [9, 14, 27, 3]`

`prices = [`  $\text{age, ticket\_price(age)}$   
for  $\text{age}$  in  $\text{ages}$  `]`

---

`prices = [`  $\text{age} = \text{ages}[0]$ ,  $\text{ticket\_price}(\text{ages}[0])$ ,  
 $\text{age} = \text{ages}[1]$ ,  $\text{ticket\_price}(\text{ages}[1])$ ,  
 $\text{age} = \text{ages}[2]$ ,  $\text{ticket\_price}(\text{ages}[2])$ ,  
 $\text{age} = \text{ages}[3]$ ,  $\text{ticket\_price}(\text{ages}[3])$ ,  
;

## Filter

values = [2, 3, 1, 17, 4]  
filter even values  
[2, 4]

evens = [x for x in values  
if is-even(x)]

evens = [x for x in values  
if  $x \% 2 == 0$ ]

