CS 152: More Lists

CS 152: Python for STEM

Colorado State University
Computer Science Department
Slides Originally Created by Albert Lionelle and Updated by Marcia Moraes

Colorado State University

Weekly Announcements!

TODO Reminders:

- Reading 12 (zybooks) – you should have already done that ☺
- Lab 08
- Reading 13 (zybooks)) – you should have already done that ☺
- Lab 08
- Coding Exam 2
- Exam2

List Slicing

Operation	Description	Example code	Example output
my_list[start:end]	Get a list from start to end (minus 1).	`my_list = [5, 10, 20]` `print(my_list[0:2])`	`[5, 10]`
my_list[start:end:stride]	Get a list of every stride element from start to end (minus 1).	`my_list = [5, 10, 20, 40, 80]` `print(my_list[0:5:3])`	`[5, 40]`
my_list[start:]	Get a list from start to end of the list.	`my_list = [5, 10, 20, 40, 80]` `print(my_list[2:])`	`[20, 40, 80]`
my_list[:end]	Get a list from beginning of list to end (minus 1).	`my_list = [5, 10, 20, 40, 80]` `print(my_list[:4])`	`[5, 10, 20, 40]`
my_list[:]	Get a copy of the list.	`my_list = [5, 10, 20, 40, 80]` `print(my_list[:])`	`[5, 10, 20, 40, 80]`

List Slicing

- A position of -1 refers to the last element of the list

```
election_years = [1992, 1996, 2000, 2004, 2008]
print(election_years[0:-1])  # Every year except the last
print(election_years[0:-3])  # Every year except the last three
print(election_years[-3:-1])  # The third and second to last years
```

```
[1992, 1996, 2000, 2004]
[1992, 1996]
[2000, 2004]
```

Modifying List During Interaction

```
my_list = [3.2, 5.0, 16.5, 12.25]

for i in range(len(my_list)):
    my_list[ i ] += 5
```

- List comprehension

```
new_list = [expression for loop_variable_name in iterable]
```

```
my_list = [10, 20, 30]
list_plus_5 = [(i + 5) for i in my_list]

print('New list contains:', list_plus_5)
```

```
New list contains: [15, 25, 35]
```

Sorting List

- Function sort() – sort the own list

```
my_list = [ 150, 47, 500, -37, 0]
my_list.sort()
```

- Function sorted() – creates a new sorted list

```
numbers = [int(i) for i in input('Enter numbers: ').split()]

sorted_numbers = sorted(numbers)

print('\nOriginal numbers:', numbers)
print('Sorted numbers:', sorted_numbers)
```

```
Enter numbers: -5 5 -100 23 4 5
Original numbers: [-5, 5, -100, 23, 4, 5]
Sorted numbers: [-100, -5, 4, 5, 5, 23]
```

Multidimensional Lists

- List nesting

```
my_list = [[10, 20], [30, 40]]
print('First nested list:', my_list[0])
print('Second nested list:', my_list[1])
print('Element 0 of first nested list:', my_list[0][0])
```

```
First nested list: [10, 20]
Second nested list: [30, 40]
Element 0 of first nested list: 10
```

Reading Values for a Two-Dimensional List

```
def readingTwoDimensionalList():
    matrix = [[0,0],[0,0]]
    for i in range(2):
        for j in range(2):
            print(f'Enter a number for [{i}][{j}]:')
            matrix[i][j] = int(input())

    return matrix
```

Writing Values for a Two-Dimensional List

```
def printTwoDimensionalList(matrix):
    for i in range(2):
        for j in range(2):
            print(f'{matrix[i][j]} ', end='')
        print()
```

Printing a Multidimensional List

```
def printTableFormat(multiList):
    for row_index, row in enumerate(multiList):
        for column_index, item in enumerate(row):
            print(f'{item:<8.2f} ', end='')
        print()


multiList = [[10.5, 4.1, 3.3], [2.6, 4], [10, 20, 30]]
printTableFormat(multiList)
```

Coding Activity

- With a Peer:
  - Write a Python function that receives a multidimensional list and prints the sum of each row and the total sum of the multidimensional list.
  - Write a Python function that returns the principal diagonal of a square matrix.
  - Write a Python function that returns the max element of the principal diagonal of a square matrix.
  - Write a Python function that receives a multidimensional list as a parameter and returns a list containing only the even number.