

CS 152: Lists and Tuples

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 6 (zybooks) – you should have already done that 😊
- Lab 04
- Reading 7 (zybooks) - you should have already done that 😊
- Lab 05
- Reading 8 (zybooks) - you should have already done that 😊

We can
do hard
things

Recall Activity

- What is the difference between a List and a Tuple? Explain using your own words.
- Write your answer to our attendance activity for today's class.

Lists

- Fundamental to Python
 - List of items, any item.
 - mutable (can be modified)

```
machete = [4, 5, 2, 3, 6, 7, 8, 9]
machete_names = ["A New Hope", "The Empire Strikes Back", "Attack of the Clones",
                 "Revenge of the Sith", "Return of the Jedi", "The Force Awakens",
                 "The Last Jedi", "The Rise of Skywalker"]
title = machete_names[2]
episode = machete[2]
print(f"Star Wars {episode}: {title}", episode, title)
```

Prints - Star Wars 2: Attack of the Clones

List Functions

Operation	Description
<code>len(list)</code>	Find the length of the list.
<code>list1 + list2</code>	Produce a new list by concatenating list2 to the end of list1.
<code>min(list)</code>	Find the element in list with the smallest value. All elements must be of the same type.
<code>max(list)</code>	Find the element in list with the largest value. All elements must be of the same type.
<code>sum(list)</code>	Find the sum of all elements of a list (numbers only).
<code>list.index(val)</code>	Find the index of the first element in list whose value matches val.
<code>list.count(val)</code>	Count the number of occurrences of the value val in list.

List Functions

```
glacier_size = [0.53, 0.34, 0.59, 0.2, 0.28, 0.13, 0.25, 5.5, 3.25, 0.84, 0.5, 0.75]  
missing_glaciers = [0.19, 0.04, 0.08, 0.29, 0.22, 0.07, 0.12]
```

```
print(min(glacier_size)) # 0.13  
print(max(glacier_size)) # 5.5  
print(glacier_size.index(5.5)) #7  
glacier_size[7] = "Five.Five"  
print(glacier_size[7]) # Five.Five
```

```
combined = glacier_size[7:9] + missing_glaciers[0:3]  
print(combined) # ["Five.Five", 3.25, 0.19, 0.04, 0.08]
```

List Functions

Adding elements to a list:

- `list.append(value)`: Adds value to the end of list. Ex: `my_list.append('abc')`

Removing elements from a list:

- `list.pop(i)`: Removes the element at index `i` from list. Ex: `my_list.pop(1)`
- `list.remove(v)`: Removes the first element whose value is `v`. Ex: `my_list.remove('abc')`

```
my_list = [10, 'bw']
print(my_list)

my_list.append('abc')
print('After append:', my_list)

my_list.pop(1)
print('After pop:', my_list)

my_list.remove('abc')
print('After remove:', my_list)
```

Tuples

- Behaves similar to a list but is immutable – once created the tuple's elements cannot be changed

```
white_house_coordinates = (38.8977, 77.0366)
print('Coordinates:', white_house_coordinates)
print('Tuple length:', len(white_house_coordinates))

# Access tuples via index
print('\nLatitude:', white_house_coordinates[0], 'north')
print('Longitude:', white_house_coordinates[1], 'west\n')

# Error. Tuples are immutable
white_house_coordinates[1] = 50
```


Named Tuples

- Allows the programmer to define a new simple data type that consists of named attributes

```
from collections import namedtuple

Car = namedtuple('Car', ['make', 'model', 'price', 'horsepower', 'seats']) # Create the named tuple

chevy_blazer = Car('Chevrolet', 'Blazer', 32000, 275, 8) # Use the named tuple to describe a car
chevy_impala = Car('Chevrolet', 'Impala', 37495, 305, 5) # Use the named tuple to describe a different car

print(chevy_blazer)
print(chevy_impala)
```

```
Car(make='Chevrolet', model='Blazer', price=32000, horsepower=275, seats=8)
Car(make='Chevrolet', model='Impala', price=37495, horsepower=305, seats=5)
```

List Activities

- Write a Python function to multiply all the items in a list.
- Write a Python function to calculate the average of the elements in a list.
- Write a Python function to calculate the number of values that are greater than the average of the elements in a list.
- Given a list, write a Python function to swap first and last element of the list.
- Given a list, write a Python function to invert the list.
- Write a Python function to find the max elements in a list and its position.
- Write a Python function to print the numbers of a specified list after removing even numbers from it.