CS 1112: Introduction To Programming

Tuples

Dr. Nada Basit

basit[*at*]Virginia[*dot*]edu

Spring 2024

Friendly Reminders

- Your ***safety*** and ***comfort*** is important!
  - If you choose to wear a mask you are welcome to do so
  - *We will interpret wearing a mask as being considerate and caring of others in the classroom (not that you are sick), and realize that some may choose to mask to remain distanced*
- Be an ***active*** participant in your learning!
  You're welcome and ***encouraged*** to ask questions during class!
- If you feel ***unwell***, or think you are, please stay home
  - *We will work with you!*
  - Get some rest ☺
  - View the recorded lectures – *please allow 24-48 hours to post*
  - *Contact us!*

Announcements

- **Quiz 5** is due by 11:00pm on 3/18 (***Tonight***)!
- **PA04** is due by 11:00pm on 3/20 (***Wednesday***)!
  - Submit on Gradescope: your .py file, and a PDF of your reflection
  - Please be mindful about submitting the right kind of files (.py file and .pdf file) as well as submitting the .py file that is named correctly (see assignment document for full details)
  - A note about submitting on Gradescope: you can submit an **UNLIMITED** number of times prior to the deadline. Look at the score you got, if you have some points taken off, that's ok, go back and fix your code and resubmit! Do this as often as you like BEFORE the assignment deadline. You cannot resubmit after the deadline.
  - ***REMEMBER ALSO***: You have a grace period of 24 hours to submit your PAs!
- **Exam 1**: if you notice grader-error, don't hesitate to let me or the TAs know!
  - Error corrections due by 11:00pm on Wednesday
  - Email me with any questions or concerns!

Announcements

- **SIGCSE Conference Travel!** I, along with many of my CS colleagues in the department, will be attending the Technical Symposium on Computer Science Education – ACM Special Interest Group on Computer Science Education (SIGCSE) in Portland, OR! I will be gone from Tuesday, 3/19 until Saturday, 3/23.

Please make note of the following:

- I will **NOT** be holding **my office hours** on Tuesday, March 19.
- **TA office hours will be the same** and will not be affected (see Office Hour Calendar on Canvas)
- **There will be class on Wednesday, March 20**, but it will be led by some of our wonderful TAs!
- **There will be class on Friday, March 22**, but it will be led by some of our wonderful TAs!
- *Please read the announcements on Canvas for any additional information*.
- *Due to the conference travel, I may be slow in responding to email but you will get a response!*

Properties of some collections

Type	**Stores**	**Syntax**
Range	ints	`range(3,7)`
String	characters	`"Hello", "abc 123"`
List	anything	`[1,2,3,6,"hello"]`
Tuple	anything	`(1,2,3,6,"hello")`

Today

Dictionary	key:value pairs	`{17:"hi", 29:"bye"}`
Set	anything	`{1,2,6,"hi"}`

Next Week

Tuples

- A Tuple is very similar to a List. However, one major difference is that **Lists** are *mutable* whereas **Tuples** are *immutable*!

```
my_tuple = (1, 2, 3, 4)

my_tuple = ('one', 'two', 'three', 'four')

my_tuple = (1, 'two', 3, 'four')

len(my_tuple) # gives the number of things in the tuple (in this case 4)

my_tuple[i] # gives the ith thing in the tuple (starts at 0)
```

Cannot use mutating methods…!

```
my_tuple = () # Empty tuple


my_tuple = ('Virginia', 'California')
my_tuple.append('Indiana') !!!!!

# Tuples are immutable – so, cannot append!
```

Can use indexing as we've seen before!

```
my_tuple = ('Purple', 'Green', 'Yellow')

print("First element:", my_tuple[0])

# First element: Purple
```

Can use indexing as we've seen before!

```
my_tuple = ('Purple', 'Green', 'Yellow')

print("Last element:", my_tuple[-1])

# Last element: Yellow
```

Some common **tuple** methods

```
my_tuple = ('Purple', 'Green', 'Yellow')

my_tuple.index('Yellow')
```

Some common **tuple** methods

```
my_tuple = ('Purple', 'Green', 'Yellow')

my_tuple.index('Yellow')

# Answer:  2
```

Some common **tuple** methods

```
my_tuple = (22, 45, 23, 78, 22, 22, 77)

my_tuple.count(22)
```

Some common **tuple** methods

```
my_tuple = (22, 45, 23, 78, 22, 22, 77)

my_tuple.count(22)

# Answer:  3
```

Some common **tuple** methods

There are **other methods** that can be used with tuples such as max() or sorted(). You're welcome to look up these and other functions to discover what they are.

Remember, none of these methods will mutate the elements in the tuple *(due to immutability!)*

Reminders

Python can access tuples **faster** than lists.

We can create a list from a tuple with type-casting

```
new_list = list(old_tuple)
```

Lists vs. Tuples

Reasons to use LISTS

- For flexibility of handling data: adding, deleting, changing (after list is created)
  - Mutability of Lists

Reasons to use TUPLES

- When using data that doesn't or shouldn't change
  - Immutability of Tuples
  - (Structure to loop through and access the data without modifying the data)
- Python can access tuples faster than lists *(better performance)*

Tuples are immutable and therefore do not have as many methods as a list. A lot of those list methods that we have seen involved mutating the values.
For tuples, we cannot append, remove, etc.
Otherwise, lists and tuples behave in pretty much the same way!

Lists vs. Tuples

This code …

```
# list vs. tuple comparison
my_list = [1, 2.7, 'wahoo', True]
my_tuple = (1, 2.7, 'wahoo', True)

for thing in my_list:
    print(thing)

for thing in my_tuple:
    print(thing)

print(my_list[2])
print(my_tuple[2])

print(my_list)
my_list[0] = 100
print(my_list)

print(my_tuple)
my_tuple[0] = 100  # Error - tuples are
immutable
print(my_tuple)
```

Produces this output

```
1
2.7
wahoo
True

1
2.7
wahoo
True

wahoo
wahoo

[1, 2.7, 'wahoo', True]
[100, 2.7, 'wahoo', True]


(1, 2.7, 'wahoo', True)
my_tuple[0] = 100
TypeError: 'tuple' object does not support
item assignment
```

Python Demonstration

Let's jump on PyCharm!

`tuples.py` - Lots of Tuple examples!

Python Demonstration

Let's jump on PyCharm!

Some live coding involving tuples!

In-Class "lab" Activity!

Activity for Today!

- In pairs or groups up to three work on the following activity.
- **tuples_ica1.py**
- *Practice writing solutions that utilize tuples*

Remember to check-in with a TA before leaving class today!

Reminder: CS Laptop Loaner Program

- This course requires students to have a **laptop**
- I realize that not everybody might have one (nor necessarily need one for their desired major / path…)
- If you do not have a laptop for any reason… ***not to worry!***
- The CS department's Systems staff has a notebook / laptop loaner program and will be able to loan you a notebook / laptop computer for the duration of the semester if you don't have one or if you cannot afford one.
  - Also available if your laptop is broken and under repair, we can arrange for you to receive a loaner laptop for a week or two until your own laptop is fixed

Interested? Link: https://www.cs.virginia.edu/wiki/doku.php?id=cs_laptop_loaner

I am happy to be your sponsor. Please let me know.