**IS201 Fundamentals of Computing**

**H0P02 Lists, Tuples**

10/10/2020 Developed by Kim Nguyen

Center for Information Assurance (CIAE) @City University of Seattle (CityU)

**Before You Start**

* The directory path shown in screenshots may be different from yours.
* Some steps are not explained in the tutorial**.** If you are not sure what to do:
  1. Consult the resources listed below.
  2. If you cannot solve the problem after a few tries, ask a TA for help.

**Learning Outcomes**

Students will be able to:

* Understand the List Data Type and Tuples in Python
* Write a Python program with functions that can be performed in Lists and Tuples

**Resources**

Matthes, E. (2019). [Python Crash Course: A Hands-On, Project-Based Introduction to Programming, 2nd Edition](https://login.proxy.cityu.edu/sso/skillport?context=146803). No Starch Press. (ISBN 9781593279288)

**Preparation**

1. In Visual Studio Code, open the private repository generated when you accepted the HOP02 assignment (If you cannot find that repository in your machine, you might have not cloned the repo, if so, please do before proceeding).

A screenshot of a cell phone

Description automatically generated

Open the terminal from the VSCode by hitting the control + ~ key, navigate into Module 1 folder using the following command:

>>> cd Module 2

**Lists**

The list is a most versatile datatype available in Python which can be written as a list of comma-separated values (items) between square brackets. Important thing about a list is that items in a list need not be of the same type.

Creating a list is as simple as putting different comma-separated values between square brackets.

**Creating, accessing and updating values**

1. Under Module2 create a file called **List\_basic.py** and type the following code

A black sign with white text

Description automatically generated

To access values in lists, use the square brackets for slicing along with the index or indices to obtain value available at that index. Python considers the first item in a list to be at position 0, not position 1.

2)Type the following in the terminal to check the output of the above code

**>>>** python3 List\_basic.py

A close up of a black background

Description automatically generated

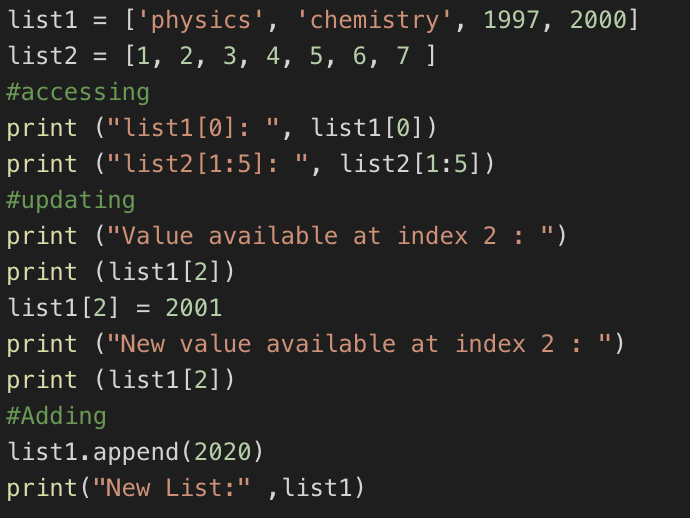
**Challenge 1: Explain the code above by commenting each line of code**

**Adding elements**

**Append**

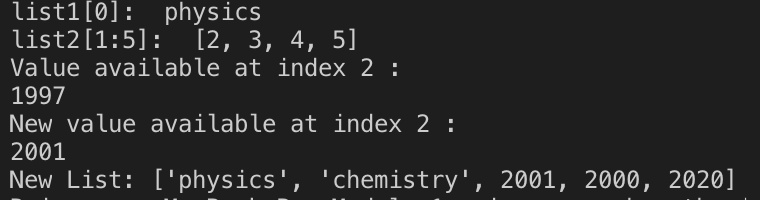
The simplest way to add a new element to a list is to *append* the item to the list. When you append an item to a list, the new element is added to the end of the list. The append() method makes it easy to build lists dynamically.

1. In the same file **List\_basic.py** add the below code



1. Type the following in the terminal to check the output of the above code

**>>>** python3 List\_basic.py



**Inserting elements**

You can add a new element at any position in your list by using the insert() method. You do this by specifying the index of the new element and the value of the new item.

1. In the same file **List\_basic.py** add the below code

A close up of text on a black background

Description automatically generated

1. Type the following in the terminal to check the output of the above code

**>>>** python3 List\_basic.py

A close up of a logo

Description automatically generated

**Removing element**

There are 3 ways to remove a element from a list.

1. If you know the position of the item you want to remove from a list, you can use the del statement.
2. The pop() method removes the last item in a list, but it lets you work with that item after removing it. The term *pop* comes from thinking of a list as a stack of items and popping one item off the top of the stack. In this analogy, the top of a stack corresponds to the end of a list. You can use pop() to remove an item from any position in a list by including the index of the item you want to remove in parentheses.
3. If you only know the value of the item you want to remove, you can use the remove() method.
4. Under Module2 create a file **List\_remove.py** and type the following code.

A picture containing drawing

Description automatically generated

1. Type the following in the terminal to check the output of the above code

>>> python3 List\_remove.py



The del used at index 1 deletes the value Yamaha. You can no longer access the value that was removed from the list after the del statement is used.

1. Add the below code in the same file

A screenshot of a cell phone

Description automatically generated

1. Type the following in the terminal to check the output of the above code

>>>python3 List\_remove.py

A picture containing drawing

Description automatically generated

At pop(), a value from the list and store that value in the variable popped\_motorcycle. We print the list to show that a value has been removed from the list. Then we print the popped value to prove that we still have access to the value that was removed.

The output shows that the value ‘suzuki’ was removed from the end of the list and is now assigned to the variable popped\_motorcycle.

1. Add the below code in the same file

A picture containing food, drawing

Description automatically generated

1. Type the following in the terminal to check the output of the above code

>>>python3 List\_remove.py



The code removes “ducati” from the list. You can use the remove() method to work with a value that's being removed from a list.

**Using Loop in lists**

1. Let’s use for loop to print out all the element of a list. Create a file **replaceNegative.py** and enter the following code:

Text

Description automatically generated

The range() function causes Python to start counting at 0 and it stops when it reaches the length of the list. The len() function calculates the length of the list. The i consist of the range index and it is used for navigating throughout the list. If the value in the list is less than 0, that is negative value then that value is converted to absolute value and returned back to list.

Run the code:

>>> python3 replaceNegative.py

A picture containing meter

Description automatically generated

**Challenge 2: Edit the program to replace any negative numbers in the list with positive ones. Expected result:**



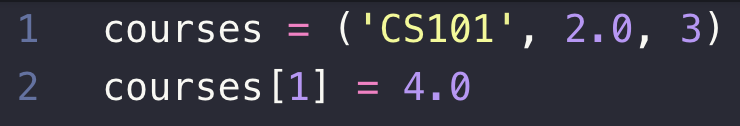
*Hint: you can use abs() function to turn negative numbers to positive*

**Tuples**

A tuple is a sequence of immutable Python objects. The tuple data type is similar to list, except tuples are typed with parentheses and cannot be modified, appended, or removed. The following is how the tuples looks like.

tup1 = ('physics', 'chemistry', 1997, 2000);

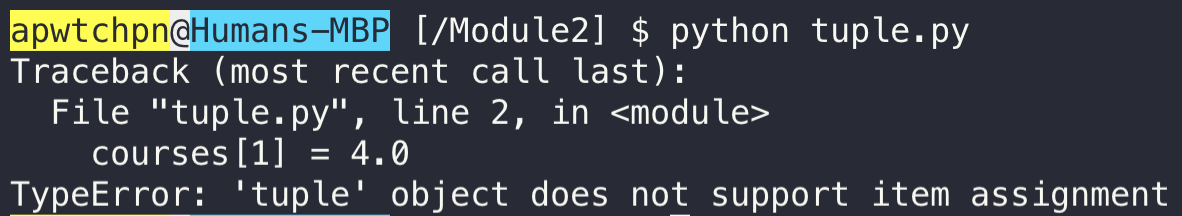
1. Create a**tuple.py** file and define a tuple with multiple data types and try to modify the grade by assigning 4.0 value to item at index 1 which is 2.0



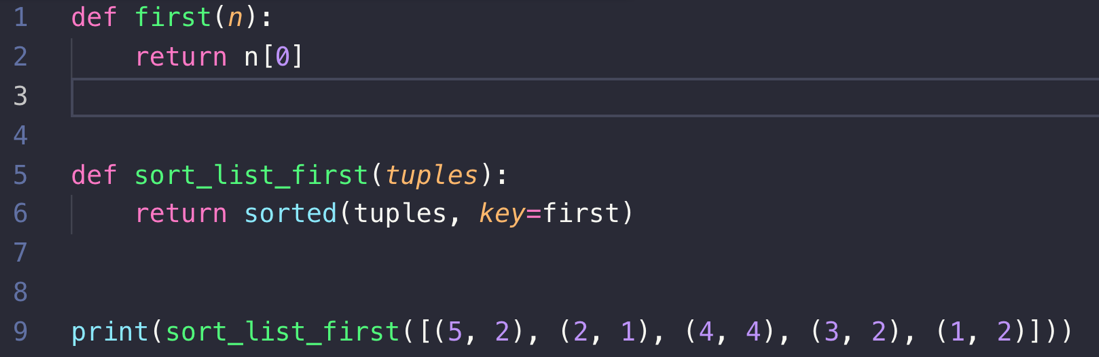
Note that you will get error message like because Tuples can’t be modified, **unlike** List

1. Type the following in the terminal to check the output of the above code

>>> python3 tuple.py



1. Create a **sortTuple.py** with the following code to sort tuples by its first value.



1. Type the following in the terminal to check the output of the above code

>>> python3 sortTuple.py

Output will be sorted.



“def” is the function and it will be explained in the later chapters. the sort function takes in a keyword argument called key. What key does is it provides a way to specify a function that returns what you would like your items sorted by. The function gets an "invisible" argument passed to it that represents an item in the list and returns a value that you would like to be the item's "key" for sorting.

**Push your work to GitHub**

Open the terminal from the VSCode by hitting the “control” + “~” key and type the following command:

>>> git add .

>>> git commit -m “Submission for Module 2 – Your GitHub Username”

>>> git push origin master