CS 1340 Introduction to Computing Concepts

Instructor: Xinyi Ding

Aug 30 2019, Lecture 3

Agenda

- Agenda:
  - Quick review concepts from previous lecture
  - Sequence data types
  - Lists
  - Common operations on lists

Variables and Data Types

- Variables
  - containers for storing data values.
  - created the moment you first assign a value to it.

- Common data types
  - Strings: *'this is a string'*
  - Integer: *39*
  - Float: *3.1415926*
  - Boolean: *Ture, False*

Sequence types

- When you have millions of items to store

```
employee_1_name = "Alice"
employee_2_name = "Bob"
employee_3_name = "Charlie"
employee_4_name = "David"
employee_5_name = "Eric"

...
```

- Use Sequence types
  - Tuples

```
employee_names = ("Alice", "Bob", "Charlie", "David", "Eric")
```

  - Lists (most often used)

```
employee_names = ["Alice", "Bob", "Charlie", "David", "Eric"]
```

Sequence types

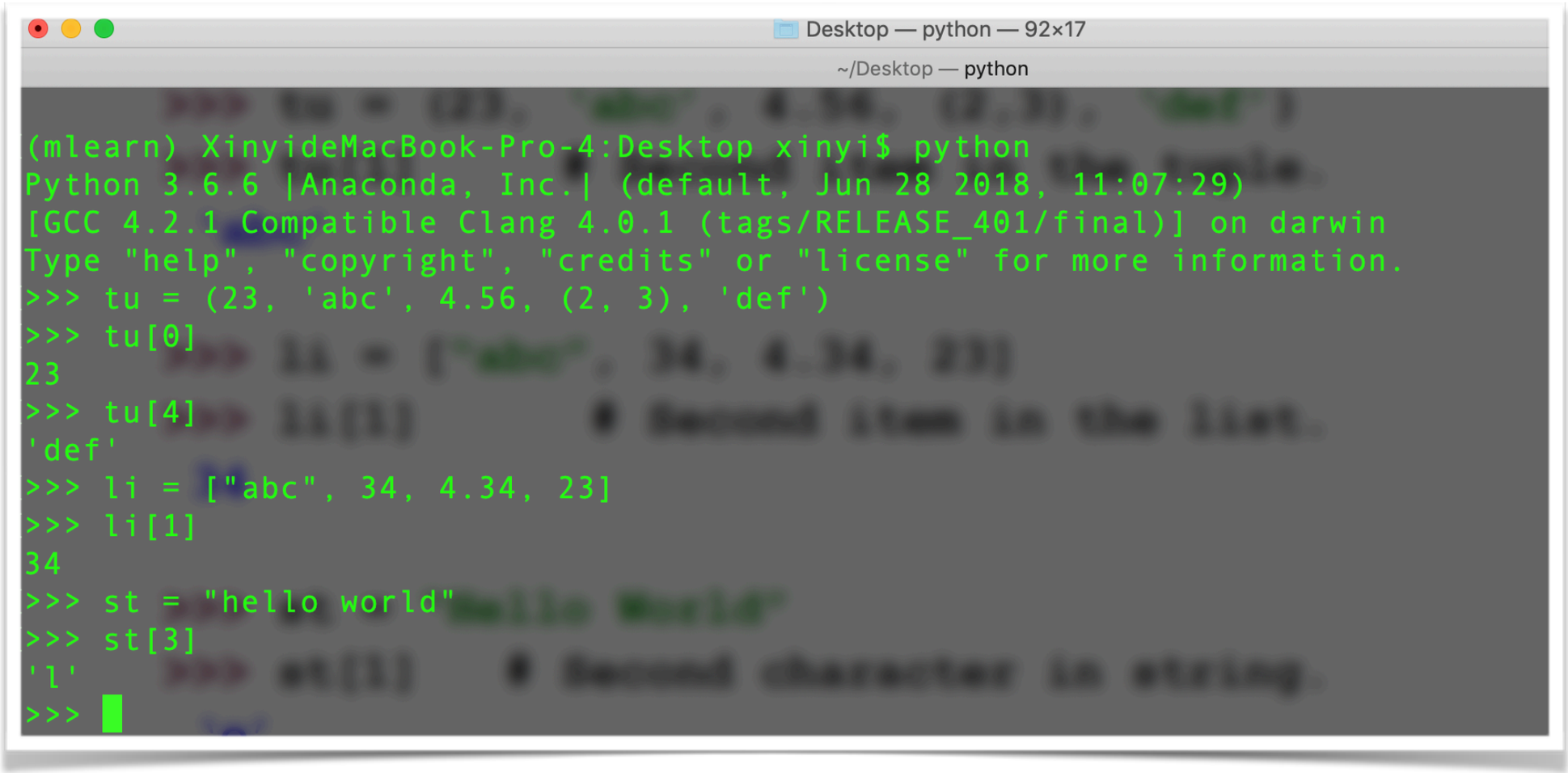
- Tuple
  - A simple **immutable** ordered sequence of items
    - Immutable: a tuple cannot be modified once created....
    - Items can be of **mixed types**, including collection types
- List
  - **Mutable** ordered sequence of items of **mixed types**
- String
  - **Immutable**
  - Conceptually very much like a tuple

Sequence types

- The three sequence types (tuples, strings, and lists) share much of the same syntax and functionality.
  - Tuples are defined using parentheses(and commas).
    - *tu = (23, 'abc', 4.56, (2, 3), 'def')*
  - Lists are defined using square brackets(and commas).
    - *li = ["abc", 34, 4.34, 39]*
- Strings are defined using quotes (", ' , or """).
  - *st = "Hello world"*
  - *st = 'Hello world'*
  - *st = """ This is a multi-line string that uses triple quotes """*

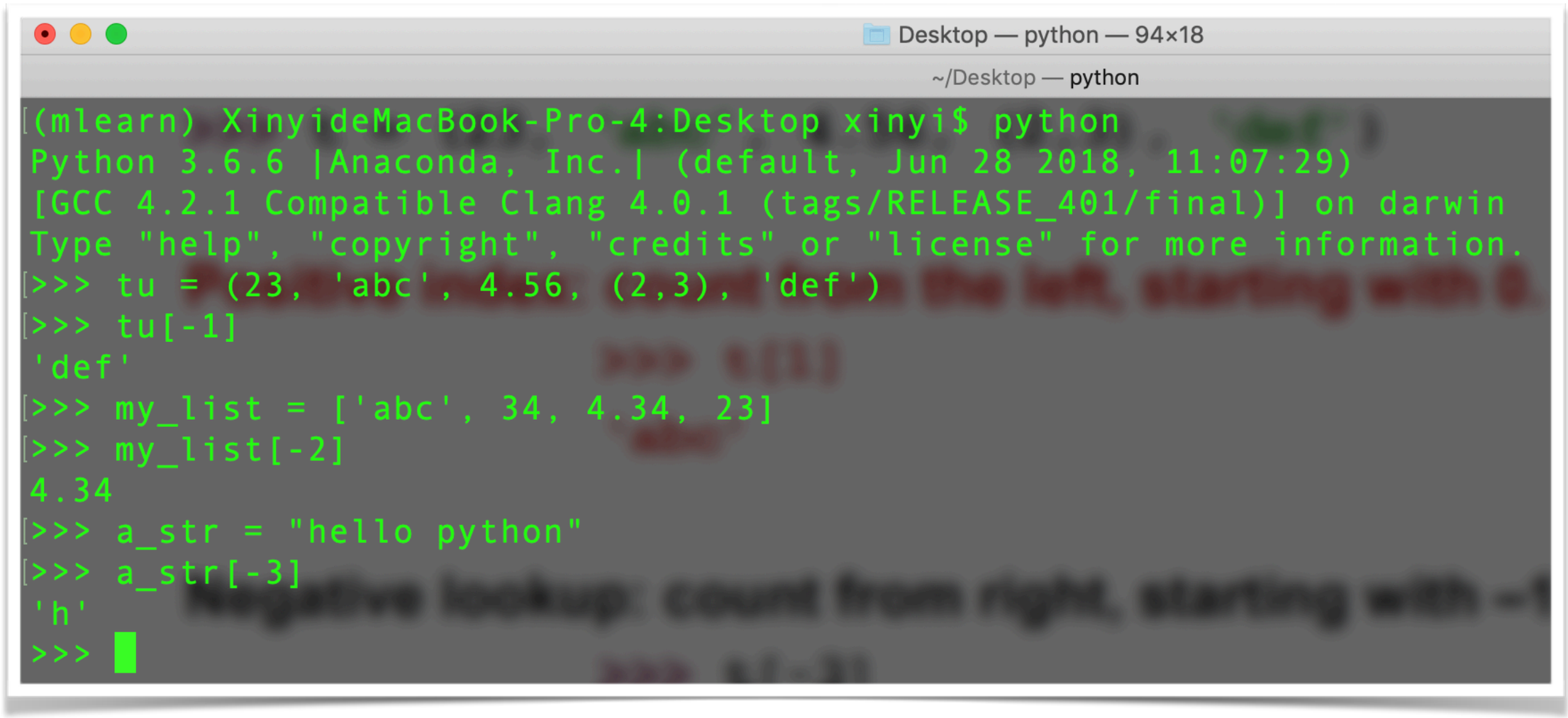
Sequence types

- We can access individual members of a tuple, list, or string using square bracket notation.
- **Note that all are 0 based...**



Negative indices

- Positive index: count from the left, **starting with 0**
- Negative lookup: count from the right, **starting with -1**



Index Errors

- Index Errors
  - when you have three items in your list, but you ask for the fourth element (Python starts indexing at 0, not 1)

```
>>> employee_names = ["Alice", "Bob", "Charlie", "David", "Eric"]
>>> employee_names[5]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
IndexError: list index out of range
>>>
```

- A little bit summary, we have seen:
  - *syntax errors, variable undefined errors, index errors.*

Slicing: Return a copy of a subset

- Return a copy of the container with a subset of the original members. Start copying at the first index, and stop copying before the second index.

```
>>> employee_names = ["Alice", "Bob", "Charlie", "David", "Eric"]
>>> employee_names[1:4]
['Bob', 'Charlie', 'David']
>>> employee_names[0:4]
['Alice', 'Bob', 'Charlie', 'David']
>>> employee_names[:4]
['Alice', 'Bob', 'Charlie', 'David']
>>> employee_names[4:]
['Eric']
```

Lists

- **Mutable** ordered sequence of items of **mixed types**
- Use square brackets []
- Individual elements in the list are separated by commas
- Indexing starts from 0, not 1

```
>>> bicycles = ['trek', 'cannondale', 'redline', 'specialized']
>>> print(bicycles)
['trek', 'cannondale', 'redline', 'specialized']
>>>
```

Accessing Elements in a List

```
>>> bicycles = ['trek', 'cannondale', 'redline', 'specialized']
>>> print(bicycles)
['trek', 'cannondale', 'redline', 'specialized']
>>> print(bicycles[0])
trek
>>> print(bicycles[1])
cannondale
>>> print(bicycles[-1])
specialized
>>>
```

Using individual element in a List

```
>>> my_first_biclye = bicycles[0]
>>> message = "My first bicyle was a " + my_first_biclye
>>> print(message)
My first bicyle was a trek
>>> message = "My first bicyle was a " + bicycles[0]
>>> print(message)
My first bicyle was a trek
>>> print("My first bicyle was a " + bicycles[0])
My first bicyle was a trek
>>>
```

Demo

Changing, Adding, and Removing Elements

- Modifying elements in a List
  - To change an element, use the name of the list followed by the index of the element you want to change, and then provide the new value you want that item to have

```
>>> motorcycles = ['honda', 'yamaha', 'suzuki']
>>> print(motorcycles)
['honda', 'yamaha', 'suzuki']
>>> motorcycles[0] = 'mazda'
>>> print(motorcycles)
['mazda', 'yamaha', 'suzuki']
>>>
```

Changing, Adding, and Removing Elements

- Adding elements to a List
  - Appending elements to the end of a list
  - Inserting elements into a list at any position
  - The extend method

```
>>> motorcycles = ['honda', 'yamaha', 'suzuki']
>>> motorcycles.append('bmw')
>>> print(motorcycles)
['honda', 'yamaha', 'suzuki', 'bmw']
>>> motorcycles.insert(0, 'tesla')
>>> print(motorcycles)
['tesla', 'honda', 'yamaha', 'suzuki', 'bmw']
>>> motorcycles.insert(1, 'jeep')
>>> print(motorcycles)
['tesla', 'jeep', 'honda', 'yamaha', 'suzuki', 'bmw']
>>> l1 = [1, 2, 3]
>>> l2 = [4, 5, 6]
>>> l1.extend(l2)
>>> print(l1)
[1, 2, 3, 4, 5, 6]
>>> l3 = [7, 8, 9, 10]
>>> print(l1 + l3)
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
>>>
```

Changing, Adding, and Removing Elements

- Removing elements from a list
  - Use the del statement
  - Use the pop() method
  - Removing an item by value

```
>>> motorcycles = ['honda', 'mazda', 'yamaha', 'suzuki', 'bmw']
>>> del motorcycles[0]
>>> print(motorcycles)
['mazda', 'yamaha', 'suzuki', 'bmw']
>>> motorcycles.pop()
'bmw'
>>> print(motorcycles)
['mazda', 'yamaha', 'suzuki']
>>> motorcycles.pop(1)
'yamaha'
>>> print(motorcycles)
['mazda', 'suzuki']
>>> motorcycles = ['honda', 'mazda', 'yamaha', 'suzuki', 'bmw']
>>> print(motorcycles)
['honda', 'mazda', 'yamaha', 'suzuki', 'bmw']
>>> motorcycles.remove('honda')
>>> print(motorcycles)
['mazda', 'yamaha', 'suzuki', 'bmw']
>>>
```

Other useful operators on lists

- The 'in' operator
- The + operator
- Get the index of an element
- Count some specific element in a list

```
>>> motorcycles = ['honda', 'mazda', 'suzuki', 'bmw', 'jeep']
>>> 'honda' in motorcycles
True
>>> 'tesla' in motorcycles
False
>>> motorcycles.index('bmw')
3
>>> motorcycles.count('jeep')
1
>>> motorcycles.append('jeep')
>>> print(motorcycles)
['honda', 'mazda', 'suzuki', 'bmw', 'jeep', 'jeep']
>>> motorcycles.count('jeep')
2
```

Organizing a list

- Sort a list
- Reverse order
- Finding the length of a list

```
>>> numbers = [8, 3, 6, 2, 1, 5]
>>> numbers.sort()
>>> print(numbers)
[1, 2, 3, 5, 6, 8]
>>> numbers.reverse()
>>> print(numbers)
[8, 6, 5, 3, 2, 1]
>>> len(numbers)
6
```

More resources : https://docs.python.org/3/tutorial/datastructures.html

Some useful operations on Strings

- Convert to upper case
- Convert to lower case
- Remove whitespaces

```
>>> str_test = "hello python"
>>> str_test.upper()
'HELLO PYTHON'
>>> print(str_test)
hello python
>>> str_test2 = "HELLO WORLD"
>>> str_test2.lower()
'hello world'
>>> str_whitespace = " hello world "
>>> str_whitespace.strip()
'hello world'
>>> str_whitespace
' hello world '
```

More resources : https://docs.python.org/3.7/library/string.html

Next Week

- Dictionary
- Control statement, if/else for loops, etc