

LECTURE 2

PYTHON TOUR PART I

TYPES, CONTROL STRUCTURES, I/O

MCS 275 Spring 2021

Emily Dumas

LECTURE 2: PYTHON TOUR

Course bulletins:

- Read the [syllabus](#) completely, ask if unclear.
- Discord open (link in the zoom chat or Blackboard).
- Worksheet 1 available now. Try to finish it this week; may not finish in discussion.
- Quiz 1 schedule adjustment due to MLK holiday:
Posted at Noon on Tue Jan 19, deadline Noon on Wed Jan 20.

PLAN FOR TODAY

Start our quick tour of Python, summarizing some material I think you saw in a previous course^{*}.

I'll indicate where you can find more detailed coverage in Lutz (textbook) or in the [online MCS 260 materials from my Fall 2020 course](#).

^{*} If I mention things today that are completely new to you, I'd like to know. Email or send a message in discord.

TEXTBOOK NOTE

Learning Python, 5ed was written in 2013 and so it discusses both Python 3 (our focus in MCS 275) as well as an older language (Python 2) that is no longer supported and which we won't discuss or use at all.

NOTES FOR SELF STUDY

I'll do most examples as live coding today.

For study outside of lecture, I wrote a larger collection of examples that roughly follow the same outline:

[MCS 275 Python tour](#)

SCRIPTS AND REPL

Two ways to run Python code:

- One statement at a time, in **interactive mode**, also known as the **REPL** (read-eval-print loop)
- A whole file at a time, in **script mode**

See Lutz, Chapter 3 or MCS 260 [Lec 2](#).

VARIABLES AND TYPES

Create new vars by assignment, `name = value`

Dynamically typed: No need to specify the type of a variable, nor for it to remain the same.

Basic types include: `int`, `float`, `boolean`, `string`, `None`

See Lutz, Chapters 4-6 and MCS 260 [Lec 2](#) and [Lec 3](#).

LISTS AND DICTS

Lists are mutable ordered collections of elements, accessible by integer index.

```
[260, 275, "hello", True, None, None, -1.5]
```

Dictionaries (dicts) are mutable key-value mappings. Index like lists, but use key instead of position.

```
{ "name": "Stinger", "age": 403,  
  "species": "space wasp", "hostile": True }
```

See Lutz, Chapter 8 and MCS 260 [Lec 5](#) and [Lec 10](#).

STRINGS

Strings support some list-like features, such as indexing and slicing.

Lists have useful methods such as `.lower()`, `.startswith(...)`, `.format(...)`, and [many more](#).

See Lutz, Chapter 7 and MCS 260 [Lec 4](#), [Lec 11](#), [Lec 13](#).

IF-ELSE-ELIF

If statement (or **conditional**) runs a block of code only if a condition is True. Elif/else allow chained tests.

```
if GREAT:
    RUNS_IF_GREAT_IS_TRUE
elif OKAY:
    RUNS_IF_OKAY_IS_TRUE_AND_GREAT_IS_FALSE
else:
    RUNS_OTHERWISE
```

Non-boolean conditions are coerced: empty list, empty dict, empty string, None, and zero map to False.

See Lutz, Chapter 12 and MCS 260 [Lec 6](#) and [Lec 18](#).

LOOPS

While: Keep going until a condition becomes False

```
while CONDITION:  
    STUFF_TO_DO # should modify things in the condition
```

For: Take items (list elements, dict keys) out, one at a time, and do something with each.

```
for ITEM in CONTAINER:  
    STUFF_TO_DO # should use the ITEM
```

See Lutz, Chapter 13 and MCS 260 [Lec 7](#).

FILES

`open(filename, mode, ...)` opens a file and returns a **file object**. Mode string selects reading ("r"), writing ("w"), ...

Methods of the file object perform input/output (I/O).

Read/write text to text files ("t" in mode), bytes to binary files ("b" in mode).

`.close()` a file when finished.

The basics are in Lutz, Chapter 9 and MCS 260 [Lec 13](#).

REFERENCES

- The [MCS 275 Python tour](#) is an expanded written version of the live coding examples from today's lecture.
- Today's slides referenced chapters from Lutz (Learning Python 5ed).
 - Free access to online book for UIC students; see course web page.
- [MCS 260 Fall 2020 home page](#) has slide presentations, sample code, and other resources for review.

REVISION HISTORY

- 2021-01-14 Fixed book edition typo
- 2021-01-13 Added link to written Python tour; small text revisions
- 2021-01-12 Initial publication

