

LECTURE 2

PYTHON TOUR PART I

TYPES, CONTROL STRUCTURES, I/O

MCS 275 Spring 2023

David Dumas

LECTURE 2: PYTHON TOUR

Reminders and announcements:

- Read the [syllabus](#).
- Discord open (invite link on Blackboard).
- Homework 1 will be due Noon on Wed 18 January.
Not posted yet.

PLAN FOR TODAY

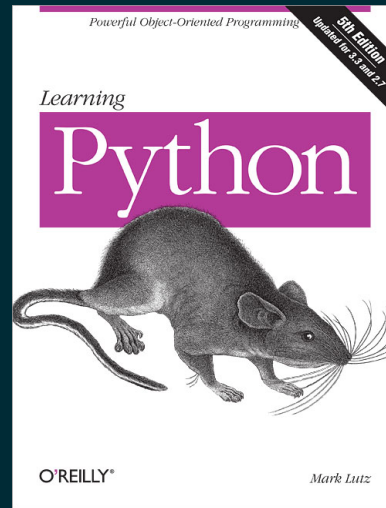
Start our quick tour of Python, summarizing some material I think you saw in a previous course^{*}.

I'll indicate where you can find more detail in optional texts and the [online MCS 260 materials from my Fall 2021 course](#).

^{*} If I mention things today that are completely new to you, please let me know afterward.

TEXTBOOK NOTE

The most comprehensive optional text is



Learning Python, 5ed, by Mark Lutz

Written in 2013, so it discusses Python 3 and Python 2. Since then, Python 2 has been phased out.

NOTES FOR SELF STUDY

I'll do most examples as live coding today.

Options to study this outside of lecture:

- These slides: Main points summarized succinctly.
- Notebooks written in class today (posted later).
- [All the MCS 260 lecture slides](#): Much more detailed (perhaps *too* detailed)

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](#).

* Notebooks are a third way we'll discuss in this course.

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 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 7](#).

IF-ELIF-ELSE

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: # elif abbreviates "else, if"  
    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 6](#).

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](#) and [Lec 14](#).

REFERENCES

- Later, I'll post the Python tour notebook I worked on in lecture.
- Today's slides referenced chapters from *Lutz* (Learning Python 5ed).
 - UIC students can access the online book for free, but login is required. Instructions on Blackboard.
- [MCS 260 Fall 2021 home page](#) has slide presentations, sample code, and other resources for review.

REVISION HISTORY

- 2023-01-10 Initial publication
- 2023-01-11 Fixed year

