Python Fundamentals

Wednesday: Dictionaries, Files, and Writing Functions

Colby Witherup Wood

NU IT Research Computing Services

About this bootcamp

Cameras are **not** required during these lectures.

During lectures, ask questions in the Zoom chat. If you know the answers, feel free to answer questions in the chat, too.

If my internet goes out during a lecture, that means everyone gets a 10-minute break! (Has never happened, but it's good to have a plan.)

About this bootcamp

From 3 pm to 4 pm Central I will be back here (same Zoom channel) to answer any questions about the homework or quiz.

You can also email me questions at colby.witherup@northwestern.edu and I will do my best to answer as soon as I can.

Plan for the week – **6 basic objects**

LEARN PYTHON

Monday

- Objects
  - **integers**
  - **floats**
  - **booleans**
  - **strings**
- Functions
  - basic operators
  - convert between objects
  - string functions
- Concepts
  - naming variables
  - indexing strings

Tuesday

- Objects
  - **lists**
- Functions
  - list functions
- Concepts
  - for loops
  - if statements
  - error handling
  - importing functions from modules

Wednesday

- Objects
  - **dictionaries**
  - files
- Concepts
  - writing files
  - looping through dictionaries
  - writing your own functions

Thursday

- Concepts
  - interactive coding
  - writing scripts
  - running scripts
  - putting it all together

DATAFRAMES

Friday

- Objects
  - pandas dataframes
  - pandas series
  - plots
- Functions
  - dataframe functions
  - series functions
- Concepts
  - selecting data
  - filtering data
  - plotting data

```
for p in row:
    if p is Avenger:
        p.speak("Hello")
    elif p in Star Wars:
        p.speak("Luke!")
    else:
        pass
```

How would a computer know which of these characters is an Avenger or which is in Star Wars?

Dictionary

a collection of key: value pairs

dictionary syntax

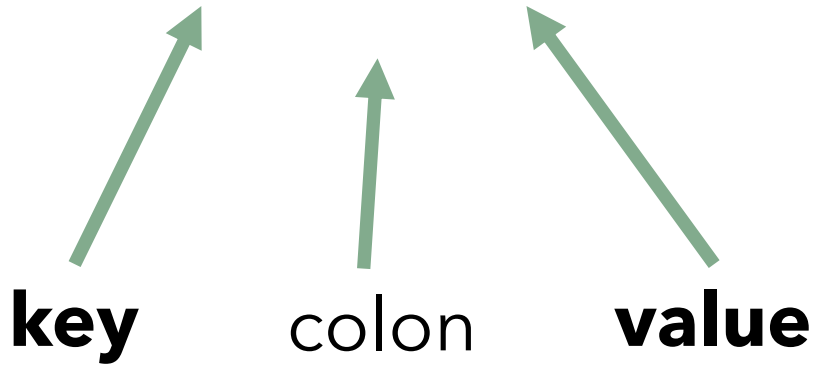
```
inches_dict = {"Jo": 60, "Rae": 68, "Tom": 65}
```

common shorthand
for dictionary

curly brackets

dictionary syntax

```
inches_dict = {"Jo": 60, "Rae": 68, "Tom": 65}
```



dictionary syntax

```
inches_dict = {"Jo": 60,
               "Rae": 68,
               "Tom": 65}
```

Can be written like this for clarity.

Options to open mondayLecture.ipynb:

If you are running Jupyter Lab on your own computer:

- Go to https://github.com/aGitHasNoName/pythonBootcampWednesday. Click on the green Clone button and choose Download ZIP. Unzip that folder. Open Anaconda Navigator and open Jupyter Lab. Navigate to the folder you just unzipped. Open wednesdayLecture.ipynb

If you are using Google Colab (online)

- Go to colab.research.google.com. Choose GitHub on the orange menu. Search for agithasnoname/pythonbootcampwednesday. Choose wednesdayLecture.ipynb

Prepare your desktop setup

- You can set my Zoom share up next to your own notebook
- You can run the code in your version of the notebook while just listening to me as I walk through it
- You can just sit back and watch me run through it until it is time to do an exercise

Let’s code!

Working with files

Opening files

Original (rarely use)

```
f = open("my_file.txt", "r")
do something with file
f.close()
```

This leaves the file needlessly open,
which takes up memory

Opening files

Original (rarely use)

```
f = open("my_file.txt", "r")
do something with file
f.close()
```

This leaves the file needlessly open, which takes up memory

'with/as' statement (almost always use)

```
with open("my_file.txt", "r") as f:
    save file as something else
    or save part of file
```

File automatically closes when you exit the indentation

Opening files syntax

```
with open("my_file.txt", "r") as f:
    save file as something else
    or save part of file
```

File automatically closes when you exit the indentation

Opening files syntax

```
with open("my_file.txt", "r") as f:
    save file as something else
    or save part of file
```

temporary variable,
just like in a for loop

Opening files

The open() function requires two arguments:

```
open(filename, mode)
```

what you're going to do with it

Opening files

The open() function requires two arguments:

```
open(filename, mode)
```

mode options:

- `"r"` read
- `"w"` write (wipes the file clean if it already exists) ⚠️
- `"a"` append (add to the end of whatever is already in the file)

Opening files

If you are accessing a file in your current working directory, you can just include the filename, but if the file is in a different directory, you must include either the relative or absolute path.

Let’s code!

Writing functions

Function calls

You already know how to **call** a function.

```
len(my_string)
```

```
open(my_file, "r")
```

Function definition syntax

This is the syntax to **define** a new function:

```
def function_name(any arguments needed):
    do something or create a new object
```

Function definition syntax

It is also good practice to include a **comment** that says what your function does. Comments start with a # and are ignored by Python.

```
def function_name(any arguments needed):
    #my function does something
    do something or create a new object
```

Let’s code!

WEDNESDAY HOMEWORK

Complete by 10 am Central tomorrow

Answer keys are in today's folder with the homework and quiz

- wednesdayHW1.ipynb
- wednesdayHW2.ipynb
- wednesdayHW3.ipynb
- wednesdayQuiz.ipynb

The homework contains new skills, it isn't just repetition.

See you (maybe) at 3 pm Central