Computational Linguistics HC2

Introduction to Python

Meaghan Fowlie

Lecture 2

Today

Python

- Jupyter Notebooks
- IDEs & Mini-assignment
- Terminal

Git

Outline for Section 1

Python

- This course uses the programming language Python

Python

- This course uses the programming language Python
- If it's completely new to you, that's fine! We teach Python from scratch

Python

- This course uses the programming language Python
- If it's completely new to you, that's fine! We teach Python from scratch
- If you have never programmed before, you may need some extra help, but the materials are designed for beginners.

Python

- Python interpreter: turns Python code, line by line, into something the computer can execute

Python

- Python interpreter: turns Python code, line by line, into something the computer can execute
  - Installing Anaconda installs Python

Python

- Python interpreter: turns Python code, line by line, into something the computer can execute
  - Installing Anaconda installs Python
  - You can use Anaconda to make an “environment” (A Python interpreter and some Python packages) for a specific use.

Python

- Python interpreter: turns Python code, line by line, into something the computer can execute
  - Installing Anaconda installs Python
  - You can use Anaconda to make an “environment” (A Python interpreter and some Python packages) for a specific use.
  - There are several ways to make a Python interpreter run on some Python code. We’ll see as we go along today.

Python

- Python interpreter: turns Python code, line by line, into something the computer can execute
  - Installing Anaconda installs Python
  - You can use Anaconda to make an “environment” (A Python interpreter and some Python packages) for a specific use.
  - There are several ways to make a Python interpreter run on some Python code. We’ll see as we go along today.
- Python file: a plain-text file containing Python code

Python

- Python interpreter: turns Python code, line by line, into something the computer can execute
  - Installing Anaconda installs Python
  - You can use Anaconda to make an "environment" (A Python interpreter and some Python packages) for a specific use.
  - There are several ways to make a Python interpreter run on some Python code. We'll see as we go along today.
- Python file: a plain-text file containing Python code
  - File suffix `.py`, e.g. `my_python_file.py`

Python

- Python interpreter: turns Python code, line by line, into something the computer can execute
  - Installing Anaconda installs Python
  - You can use Anaconda to make an "environment" (A Python interpreter and some Python packages) for a specific use.
  - There are several ways to make a Python interpreter run on some Python code. We'll see as we go along today.
- Python file: a plain-text file containing Python code
  - File suffix `.py`, e.g. `my_python_file.py`
  - Technically can open in any plain-text file editor, but usually it's best to open it in an Integrated Development Environment like Spyder (comes with Anaconda) or VSCode.

Python

- Python interpreter: turns Python code, line by line, into something the computer can execute
  - Installing Anaconda installs Python
  - You can use Anaconda to make an "environment" (A Python interpreter and some Python packages) for a specific use.
  - There are several ways to make a Python interpreter run on some Python code. We'll see as we go along today.
- Python file: a plain-text file containing Python code
  - File suffix `.py`, e.g. `my_python_file.py`
  - Technically can open in any plain-text file editor, but usually it's best to open it in an Integrated Development Environment like Spyder (comes with Anaconda) or VSCode.
  - Warning: opening a Python file in some programs, like, I think, Word, can cause a bunch of weirdness in the formatting that is hard to see and hard to fix.

Python

- Python interpreter: turns Python code, line by line, into something the computer can execute
  - Installing Anaconda installs Python
  - You can use Anaconda to make an "environment" (A Python interpreter and some Python packages) for a specific use.
  - There are several ways to make a Python interpreter run on some Python code. We'll see as we go along today.
- Python file: a plain-text file containing Python code
  - File suffix `.py`, e.g. `my_python_file.py`
  - Technically can open in any plain-text file editor, but usually it's best to open it in an Integrated Development Environment like Spyder (comes with Anaconda) or VSCode.
  - Warning: opening a Python file in some programs, like, I think, Word, can cause a bunch of weirdness in the formatting that is hard to see and hard to fix.
  - All of your hand-in assignments include Python files.

Jupyter Notebooks

Jupyter Notebook (formerly IPython Notebook): a (technically plain-text) file that contains Python code AND Markdown code

- Opens in a web browser or in some IDEs like Visual Studio if they have the right plug-ins

Jupyter Notebooks

Jupyter Notebook (formerly IPython Notebook): a (technically plain-text) file that contains Python code AND Markdown code

- Opens in a web browser or in some IDEs like Visual Studio if they have the right plug-ins
- All of your exercises are in Jupyter Notebooks

Jupyter Notebooks

Jupyter Notebook (formerly IPython Notebook): a (technically plain-text) file that contains Python code AND Markdown code

- Opens in a web browser or in some IDEs like Visual Studio if they have the right plug-ins
- All of your exercises are in Jupyter Notebooks
- File suffix `.ipynb` e.g. `my_jupyter_notebook.ipynb`

Jupyter Notebooks

Jupyter Notebook (formerly IPython Notebook): a (technically plain-text) file that contains Python code AND Markdown code

- Opens in a web browser or in some IDEs like Visual Studio if they have the right plug-ins
- All of your exercises are in Jupyter Notebooks
- File suffix `.ipynb` e.g. `my_jupyter_notebook.ipynb`
- Code is divided into "cells" and each cell can be edited and read by the Python interpreter individually

Jupyter Notebooks

Jupyter Notebook (formerly IPython Notebook): a (technically plain-text) file that contains Python code AND Markdown code

- Opens in a web browser or in some IDEs like Visual Studio if they have the right plug-ins
- All of your exercises are in Jupyter Notebooks
- File suffix `.ipynb` e.g. `my_jupyter_notebook.ipynb`
- Code is divided into "cells" and each cell can be edited and read by the Python interpreter individually
- Useful for teaching, demonstrating, note-taking....

Jupyter Notebooks

Exercise

- From BB > Weekly Content > Week 1 > HC 2 download both Jupyter Notebooks and put them in the folder you are using for this class. (If you don't have one yet, make one!)
- Open Anaconda Navigator (from the Start/Application menu; Linux: `anaconda-navigator` in the terminal)
- Click on Jupyter Notebook (Jupyter Lab probably works fine too)
- It should open in a web browser automatically.
- Navigate to the class folder
- Open `jupyter-notebooks.ipynb`

Spyder

- An Integrated Development Environment (IDE) that comes with Anaconda

Spyder

- An Integrated Development Environment (IDE) that comes with Anaconda
- You'll learn more about using it in a couple of weeks

Spyder

- An Integrated Development Environment (IDE) that comes with Anaconda
- You'll learn more about using it in a couple of weeks
- You can create Python files here

Spyder

- An Integrated Development Environment (IDE) that comes with Anaconda
- You'll learn more about using it in a couple of weeks
- You can create Python files here
- You can run the Python interpreter on Python files here

Spyder

- An Integrated Development Environment (IDE) that comes with Anaconda
- You'll learn more about using it in a couple of weeks
- You can create Python files here
- You can run the Python interpreter on Python files here
- You can open an IPython console and write lines of Python code here.

Spyder

- An Integrated Development Environment (IDE) that comes with Anaconda
- You'll learn more about using it in a couple of weeks
- You can create Python files here
- You can run the Python interpreter on Python files here
- You can open an IPython console and write lines of Python code here.
- Use Spyder or another IDE for your hand-in assignments

Spyder

Exercise

- From Anaconda Navigator, open Spyder
- It will open with a new Python file, probably `temp.py`
- Open a new file and name it `my_number.py`, in your course folder
- Copy and paste your Python code from `jupyter-notebooks.ipynb` into it.
- Save
- Push Play, and see the same output as you had in the Jupyter Notebook appear in the Console.
- Copy one line of your code into the console and hit enter, and see the same output for that line as you had in the Jupyter Notebook appear in the Console.

This is your solution to Mini-assignment 1. You can upload it to BB.

Console/Terminal

- You can run Python on a Python file from a console or terminal

Console/Terminal

- You can run Python on a Python file from a console or terminal
- You'll learn more about this in a couple weeks

Console/Terminal

- You can run Python on a Python file from a console or terminal
- You'll learn more about this in a couple weeks
- Recommendation: use the Anaconda console in Anaconda Navigator, if you have it and especially on Windows. It is most likely to be correctly connected to your Python interpreter.

Console/Terminal

- You can run Python on a Python file from a console or terminal
- You'll learn more about this in a couple weeks
- Recommendation: use the Anaconda console in Anaconda Navigator, if you have it and especially on Windows. It is most likely to be correctly connected to your Python interpreter.
- navigate with `cd` to the folder where your Python file is

Console/Terminal

- You can run Python on a Python file from a console or terminal
- You'll learn more about this in a couple weeks
- Recommendation: use the Anaconda console in Anaconda Navigator, if you have it and especially on Windows. It is most likely to be correctly connected to your Python interpreter.
- navigate with `cd` to the folder where your Python file is
- type `python my_python_file.py` and hit Enter

Console/Terminal

- You can run Python on a Python file from a console or terminal
- You'll learn more about this in a couple weeks
- Recommendation: use the Anaconda console in Anaconda Navigator, if you have it and especially on Windows. It is most likely to be correctly connected to your Python interpreter.
- navigate with `cd` to the folder where your Python file is
- type `python my_python_file.py` and hit Enter
- (You might have to type `python3` instead)

Console/Terminal

Exercise

- From Anaconda Navigator, open a terminal if you have one, otherwise however you usually do.
- If you don't have it in Navigator and you don't know what to do, look over a classmate's shoulder for this exercise.
- In the terminal, use `cd` to navigate to your class folder.
  - If you want to see the contents of the current folder, type `dir` (for Windows) or `ls` (otherwise) and hit enter.
  - type `cd path/to/my/cl/folder` and hit enter
  - or `cd path` enter `cd to` enter ... etc
- type `python my_number.py` and hit enter
- See the same output again!

Summary

1. Jupyter Notebooks: Practicum exercises
   - These are not Python files, but they do contain Python code
   - Can be “exported” as Python files from Jupyter Notebooks

Summary

1. Jupyter Notebooks: Practicum exercises
   - These are not Python files, but they do contain Python code
   - Can be “exported” as Python files from Jupyter Notebooks
2. Python files:
   - Plain text files containing Python code
   - Create in Spyder or another IDE
   - Test in Spyder (play button) or in the terminal

Summary

1. Jupyter Notebooks: Practicum exercises
   - These are not Python files, but they do contain Python code
   - Can be “exported” as Python files from Jupyter Notebooks
2. Python files:
   - Plain text files containing Python code
   - Create in Spyder or another IDE
   - Test in Spyder (play button) or in the terminal
   - All mini-assignments are Python files

Summary

1. Jupyter Notebooks: Practicum exercises
   - These are not Python files, but they do contain Python code
   - Can be “exported” as Python files from Jupyter Notebooks
2. Python files:
   - Plain text files containing Python code
   - Create in Spyder or another IDE
   - Test in Spyder (play button) or in the terminal
   - All mini-assignments are Python files
   - Both group assignments involve multiple Python files

Summary

1. Jupyter Notebooks: Practicum exercises
   - These are not Python files, but they do contain Python code
   - Can be “exported” as Python files from Jupyter Notebooks
2. Python files:
   - Plain text files containing Python code
   - Create in Spyder or another IDE
   - Test in Spyder (play button) or in the terminal
   - All mini-assignments are Python files
   - Both group assignments involve multiple Python files
   - Taught in more detail in a later practicum

Summary

1. Jupyter Notebooks: Practicum exercises
   - These are not Python files, but they do contain Python code
   - Can be “exported” as Python files from Jupyter Notebooks
2. Python files:
   - Plain text files containing Python code
   - Create in Spyder or another IDE
   - Test in Spyder (play button) or in the terminal
   - All mini-assignments are Python files
   - Both group assignments involve multiple Python files
   - Taught in more detail in a later practicum
3. Terminal
   - Probably best to open in Anaconda
   - run Python interpreter on a Python file

Outline for Section 2

Git version control tutorial

- Open `git_tutorial.html`
- This isn't really a stand-alone tutorial: please follow along with me.
- If you haven't already:
  - Install GitHub Desktop (see BB for instructions and links)
  - Make a GitHub account
  - Download all files from Blackboard
- Alexander and I are here to answer questions when you're doing exercises