python interpreter

(Note for Chris: run md preview with Control-Shift-v with Markdown Preview Enhanced VS Plugin)

- We'll use Python 3 (3.7) for this class
- Python 2 is now no longer supported, but there's still a lot of Python 2 code around
- You (usually) cannot simply run python 2 .py files in Python 3
- Biggest obvious difference: Python 2: `print "hello"` (statement) vs Python 3: `print("hello")` (function)
- It's pretty easy to convert Python 2 code to Python 3 either manually or with the `2to3` commandline tool

- python is interpreted - you need a python interpreter installed to run any python code - if you installed anaconda, the python interpreter executable ("app") is anaconda3/python.exe (win) or anaconda/bin/python (mac)

- the interpreter "eats" code **one line at a time**
- a python script (.py file) or a jupyter python cell can contain multiple lines, which are run/executed/interpreted from top to bottom
- typically a script or cell is run via the IDE (Visual Studio code), possible with a debugger

- You could also use python in the command line (**terminal**):
    - open a terminal (or use the TERMINAL in VSC)
    - To run a script in the same folder e.g. `hello.py` :
type `python hello.py` and hit Enter
    - Run a super short literal script (1-liner):
type `python -c "print('hello')"` and hit Enter
    - Start an interactive session:
        - type 'python` and hit enter
        - you'll see `>>>` , which means the interpreter is waiting for you to type in a line of code (interactive python shell)
        - type `print('hello)' , then hit return
        - your line is interpreted, the result is shown and it's again ready for another line.
        - this is useful if you want to play around or test something
        - to exit the interactive shell, type `exit()`

- byte code files
    - importing modules can create .pyc files in a local `__pycache__` folder
    - .pyc files are not compiled but rather .py files "compressed" into bytecode

- bytecode is not human readable but substantially smaller than text
- it's OK to manually delete `__pycache__` folders, they will be recreated if needed