Software for DHTC Part 2: Interpreted Languages

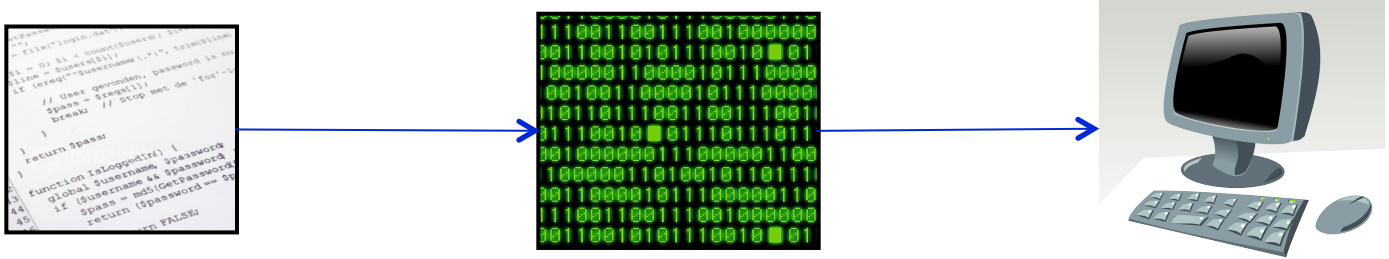
Christina Koch (ckoch5@wisc.edu)
Research Computing Facilitator
University of Wisconsin - Madison

Recap

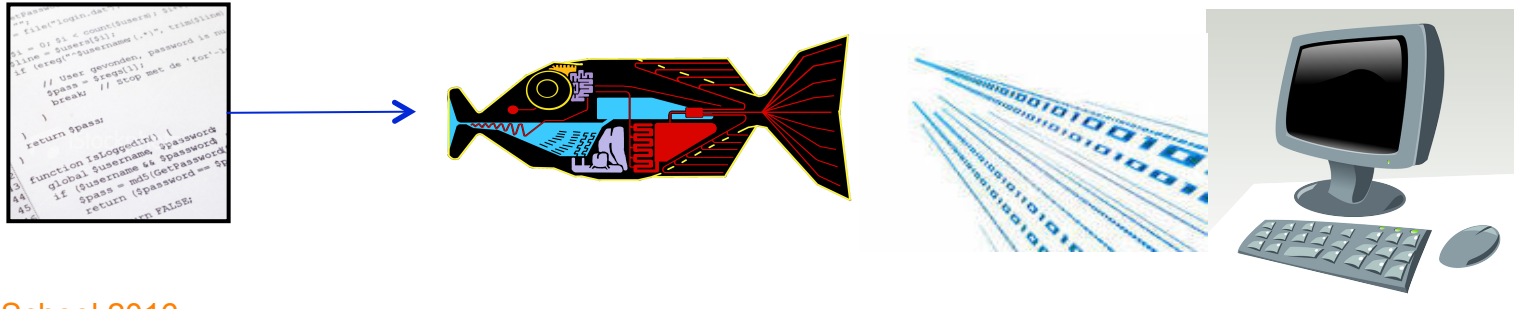
- Previous techniques:
  - Compiled code
    - Download compiled binaries
    - Build yourself
  - Wrapper script
    - Run compiled code
    - Use a pre-built installation ← we're going to explore this further in this session

Interpreted code

- Instead of being compiled and then run…

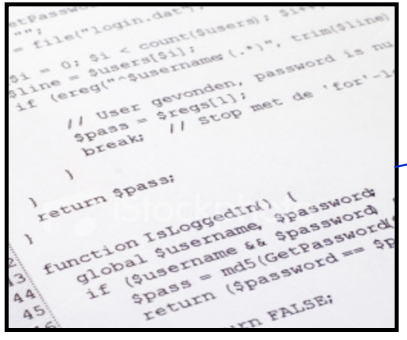


- …interpreted languages are translated into binary code “on the fly”

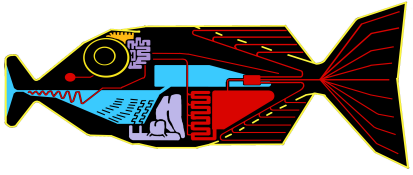


Interpretation

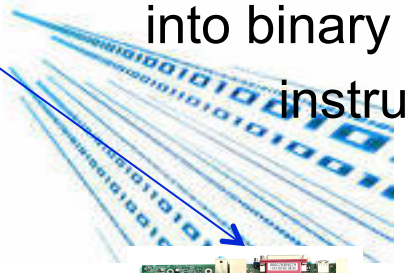
Script



Interpreter

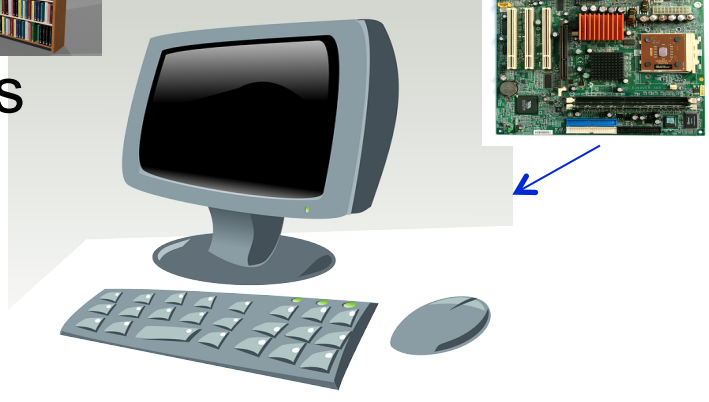


text turns into binary instructions

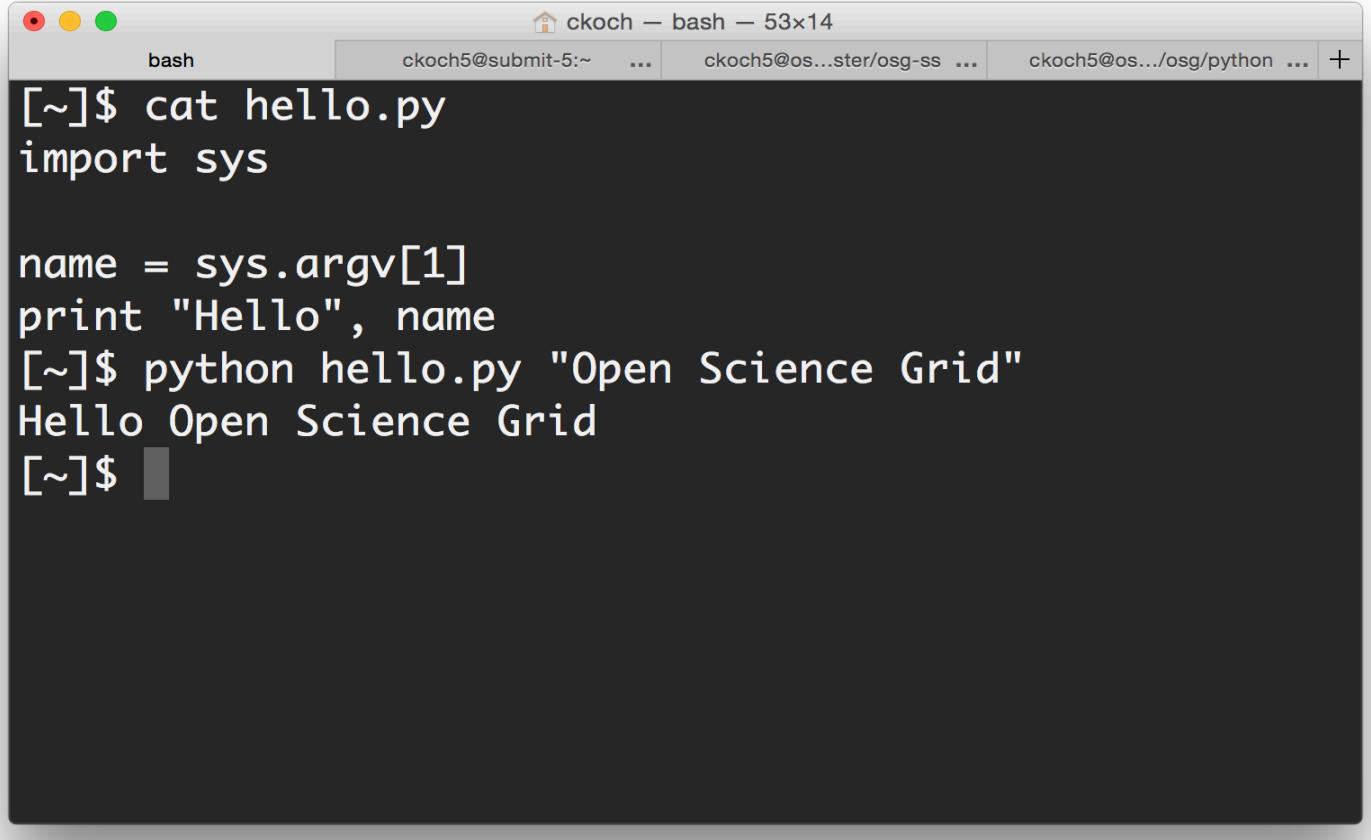


uses

Libraries



On the command line



Common interpreted languages*

- Python
- R
- Julia
- Ruby
- Matlab
- Perl
- Javascript

*Note: the line between interpreted/compiled languages can be fuzzy. Many languages support both options, with one method being more common.

Running interpreted code in jobs

General procedure

- Need to bring along interpreter and script
- Use a wrapper script as the executable
- Wrapper script will:
  - "Install" the interpreter
  - Run the script using the local installation

Python on DHTC

1. Create a portable Python installation (optional)
2. Bring along:
   - pre-built installation OR Python source code
   - your Python code
3. Use a wrapper script to:
   - unpack pre-built install OR install from source
   - run your Python script

Exercises

- Running Python Jobs
  - Exercise 4.1: Pre-building Python and using that installation
  - Exercise 4.2 (optional): Further Python job customizations
- Work on other exercises from today/ yesterday that you weren't able to finish

Questions?

- Now: Hands-on Exercises
  - 3:45 - 5:00pm