
Software Engineering

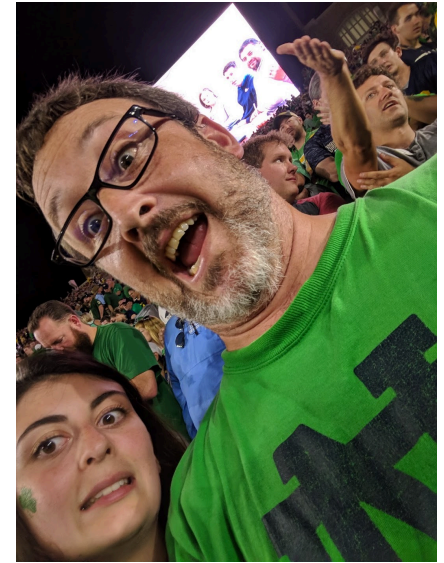
ITAO 40540

Introductions



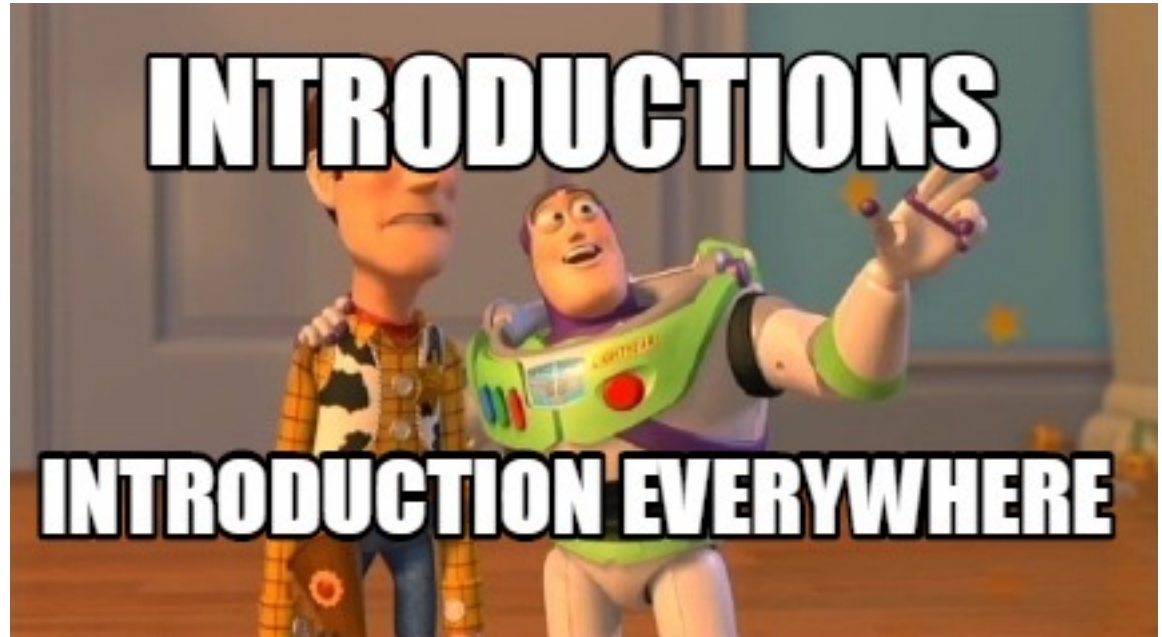
Introductions

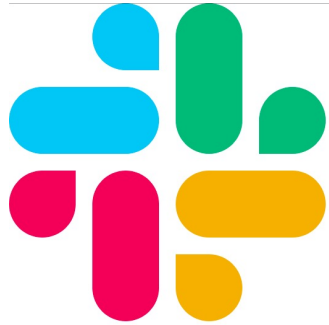
- Application Developer, Web Group
 - Lead Developer, Conductor CMS (~36K LOC)
- Concurrent Instructor
 - Building Web Applications (MGTI)
 - Data Analysis with Python (ITAO)
- At Notre Dame
 - 8 years in OIT
 - 2 years in Mendoza (Senior Web Developer)
 - 6+ years in MarComm



Introductions

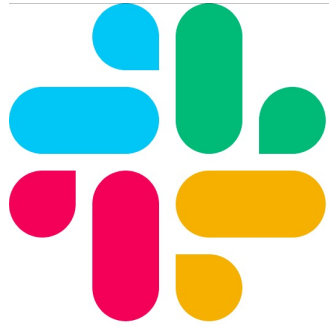
- Name
- Hometown
- What is your experience with Python?
- What are you look for out of this class?





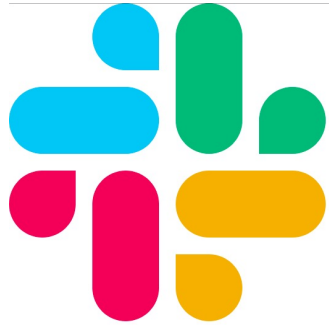
slack

- Invite link in Sakai
- Many ways to connect
 - Channels
 - Direct Message
 - Calls and Video Chat



slack

- Click on invite link and join
- **Update your image**
- Introduce yourself in the #general channel




slack

- Asking question in the #python channel
- If you have a question, someone else does as well
- To share code
 - Copy and paste
 - Don't take screenshots
 - Even better... use Git!

Anonymous Feedback Form

If you have any constructive feedback concerning this class, please use this form. It is anonymous and much appreciated.

 **jarp@nd.edu** (not shared) [Switch account](#) 

* Required

Feedback? *

Your answer

Submit

Clear form

Never submit passwords through Google Forms.

This form was created inside of University of Notre Dame. [Report Abuse](#)

Google Forms

Software Engineering...



```
types.Operator  
X mirror to the select  
object.mirror_mirror_x"  
error X"  
  
context):  
ive object is not
```

Software Engineering...



What is Software Engineering?

- What you may have experience with...
 - Basic Python
 - Looping
 - Conditionals
 - Dictionaries and Lists
 - Completing tasks with scripting
 - Working with Modules (Numpy, Pandas...)
-

Basic Python Scripting

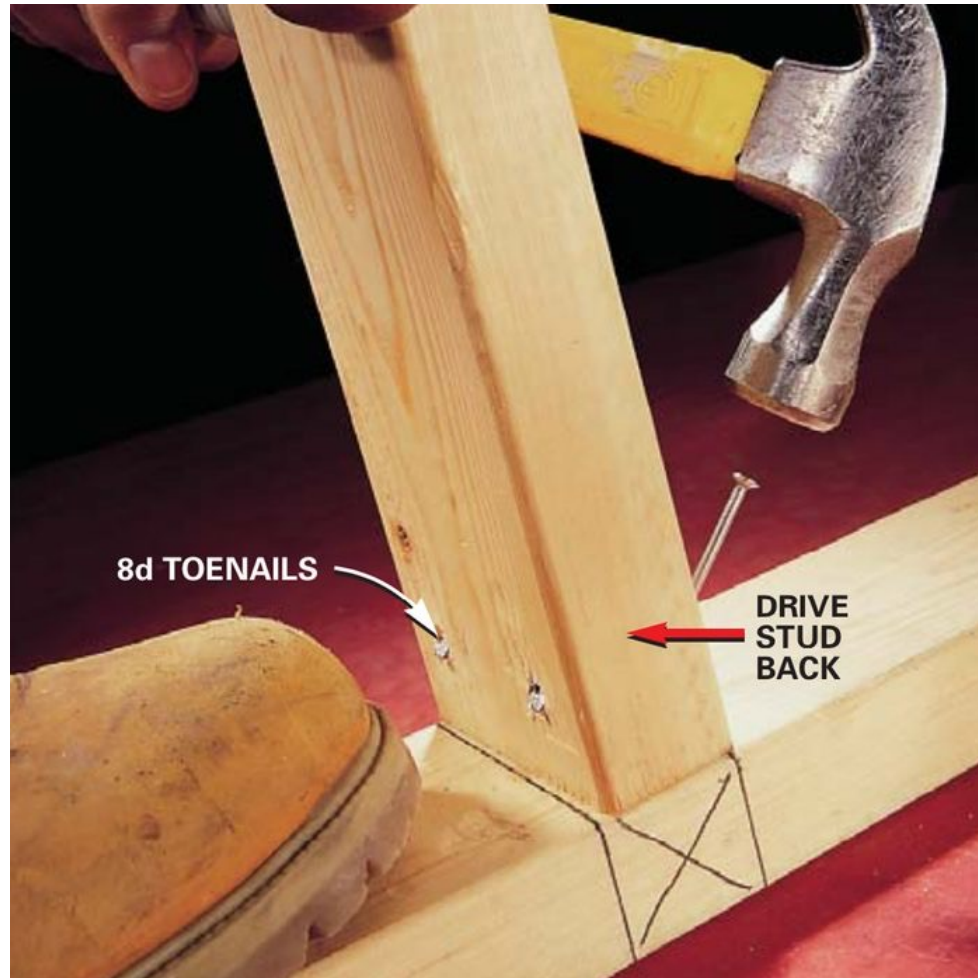
```
1 # Accept a sentence
2 sentence = input("Enter a sentence")
3
4 # Convert the sentence to a list of words using split() function
5 words_list = sentence.split()
6
7 # Create an empty dictionary that contains words and frequencies
8 words = {}
9
10 # Iterate through every word in the list
11 for word in words_list:
12     if word not in words:
13         words[word] = 1
14     else:
15         words[word] += 1
16
17     # Finally print the dictionary
18 print(words)
```

Enter a sentenceMy name is My name is My name is Jonathan
{'My': 3, 'name': 3, 'is': 3, 'Jonathan': 1}

Basic Python Scripting



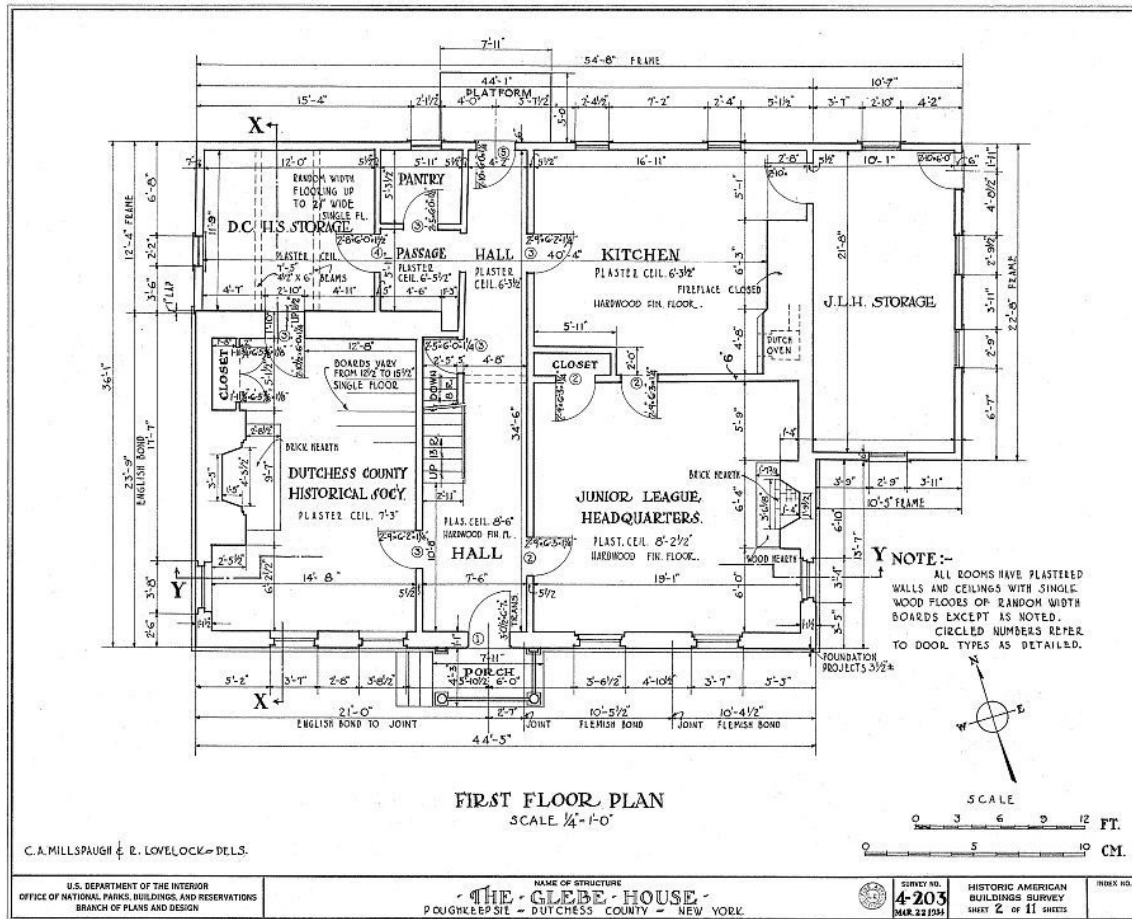
Basic Python Scripting



Basic Python Scripting



Basic Python Scripting



So... What is software engineering?

...software engineering is the process of examining and determining users needs, developing and designing it, building and forming, and testing the end-user software to see if it pleases their needs through software programming languages.

Software engineering is also used for bigger and more complex systems, which are mostly used by organizations and businesses.

https://en.wikipedia.org/wiki/Software_engineering

So... What is software engineering?

In business terms...

- Micro:

- Programs to solve specific problems
- Classes/Objects designed to encapsulate specific real world objects and their required behavior

- Macro:

- Larger programs that leverage other programs, objects, and scripting to work as a system
-

So... What is software engineering?

Software engineering is also concerned with tools and best practices that allow solutions to be robust, stable and extendible

- CLIs
 - Source Control
 - Object Oriented Programming and Design
 - File Structure with Modules and Packages
 - Other Code Libraries and their Dependencies
 - Persistence / Databases
 - Testing
 - Input Validation
 - Error Handling and Logging
-

Command Line Interfaces



Command Line Interfaces

```
...ip-172-31-253-138:~ -- zsh | ...pps.googleusercontent.com | ...jupyter-notebook • python | ... | ...by8/speakerdeck -- zsh | +
(base) jarp@jarp-mbp16 ansible-conductor % python -m pip install peppercorn
      (git)-[master]
Collecting peppercorn
  Downloading https://files.pythonhosted.org/packages/14/84/d8d9c3f17bda2b6f49406982546d6f6bc0fa188a4
3d4e3ba9169a457ee04/peppercorn-0.6-py3-none-any.whl
Installing collected packages: peppercorn
Successfully installed peppercorn-0.6
(base) jarp@jarp-mbp16 ansible-conductor % | (git)-[master]
```

Source Control / Git

```
(base) jarp@jarp-mbp16 conductor % git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
(use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add <file>.." to update what will be committed)
  (use "git restore <file>.." to discard changes in working directory)
        modified:   app/controllers/admin/abstract_pages_controller.rb
        modified:   app/controllers/application_controller.rb
        modified:   app/models/directory_group.rb
        modified:   app/models/directory_group_page.rb
        modified:   config/nginx.conf
        modified:   test/unit/directory_group_page_test.rb

no changes added to commit (use "git add" and/or "git commit -a")
(base) jarp@jarp-mbp16 conductor % git commit -am "fixing how people are rendered in preview mode"
[master 966af21ff] fixing how people are rendered in preview mode
 6 files changed, 23 insertions(+), 13 deletions(-)
(base) jarp@jarp-mbp16 conductor % git pull origin master
git: 'pull' is not a git command. See 'git --help'.

The most similar command is
  pull
(base) jarp@jarp-mbp16 conductor % git pull origin master
remote: Enumerating objects: 13, done.
remote: Counting objects: 100% (13/13), done.
remote: Total 13 (delta 12), reused 13 (delta 12), pack-reused 0
Unpacking objects: 100% (13/13), 2.25 KiB | 143.00 KiB/s, done.
From github.com:ndwebgroup/conductor
 * branch                master       -> FETCH_HEAD
  dd24cb448..fc09ce1de   master       -> origin/master
Merge made by the 'recursive' strategy.
 Gemfile.lock           | 2 +-
 app/models/site_map.rb | 43 ++++++-----
 app/views/conductor/sitemap.xml.builder | 8 ++++----
 test/unit/site_map_test.rb | 16 ++++++-----
 test/unit/site_test.rb | 2 +-
 5 files changed, 46 insertions(+), 25 deletions(-)
(base) jarp@jarp-mbp16 conductor % git push origin master
```

And now for something completely different...



Farkle



Farkle

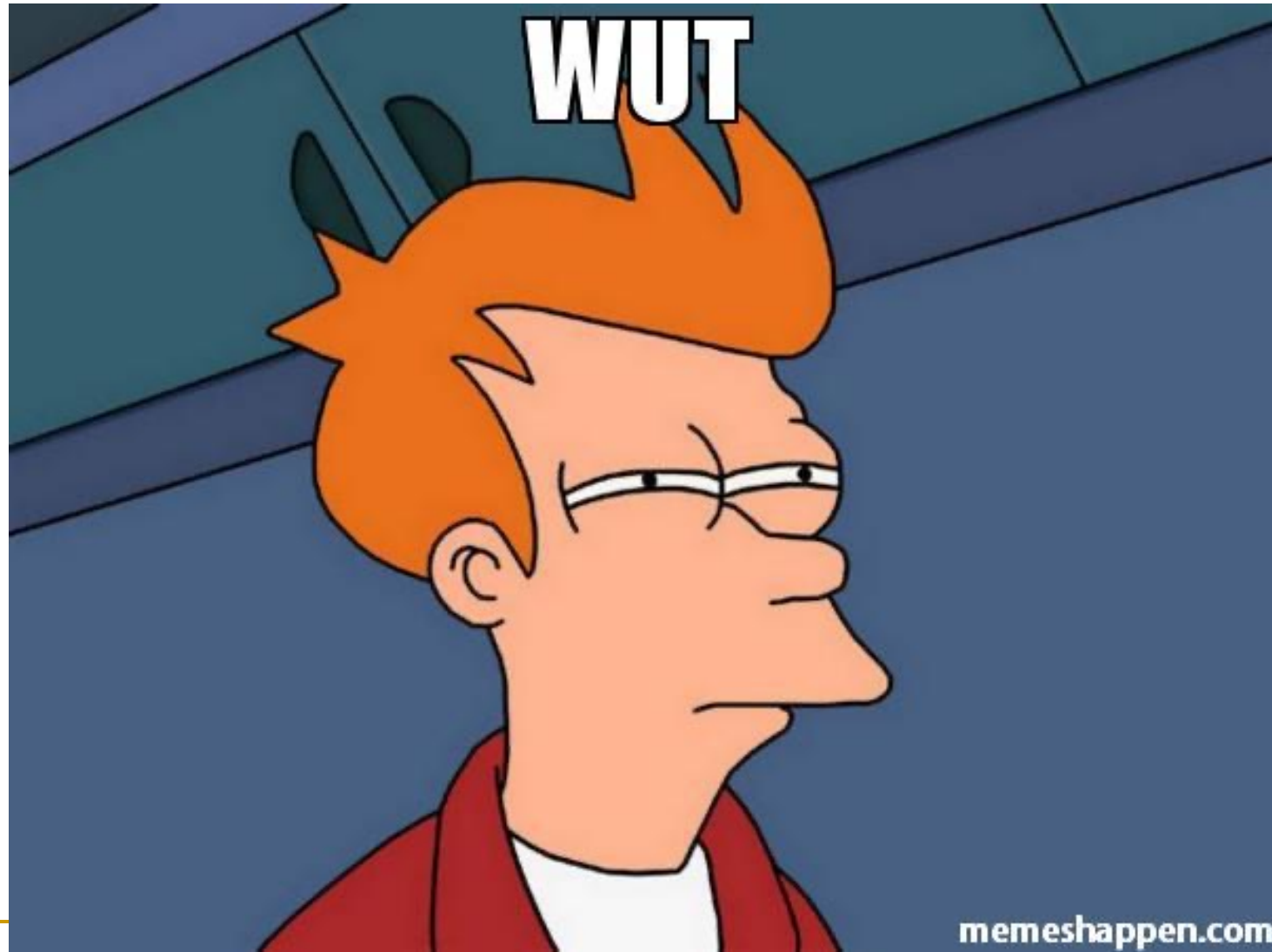
- Each player in turn rolls all six dice and checks to see if they have rolled any scoring dice or combinations.
 - Any dice that score may be set aside and then the player may choose to roll all the remaining dice
 - The player must set aside at least one scoring die of their choice if possible but is not required to set aside all scoring dice.
-

Farkle

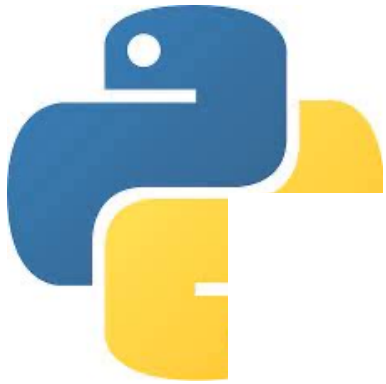
1	100 points
5	50 points
Three 1's	1,000 points
Three 2's	200 points
Three 3's	300 points
Three 4's	400 points
Three 5's	500 points
Three 6's	600 points
1-2-3-4-5-6	3000 points
3 Pairs	1500 points

The first player to score a total of 10,000 or more points wins

Farkle



Class Tools



git



Class Sessions

- Presentations
 - Introduce Tools and Concepts
 - Coding
 - Illustrate tools and concepts through coding the Farkle CLI program
 - Follow the progress outside of class via Git
-

Grading

■ Code Journal

- ❑ An important part of software engineering is learning through curiosity
 - ❑ Due at the end of the Semester/Session
 - ❑ Will be submitted through Git
 - ❑ Should have commits spanning the entire semester
 - ❑ Graded based on creativity and effort
-

Grading

- Final Exam

- Written, closed book, exam
- Covers concepts and coding knowledge



Grading

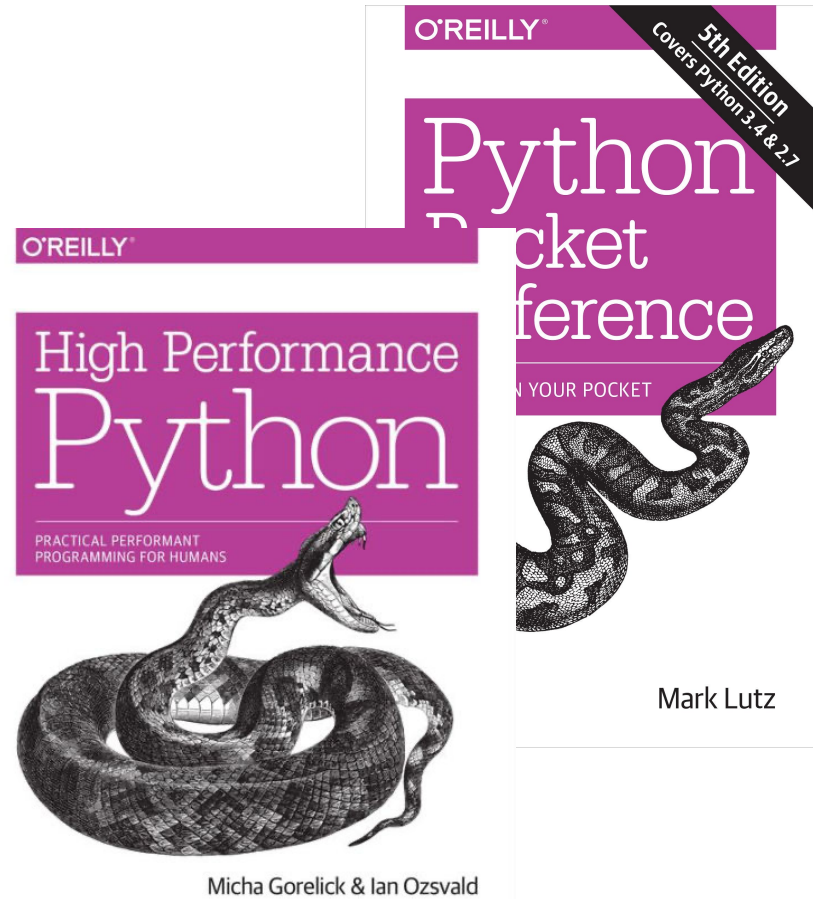
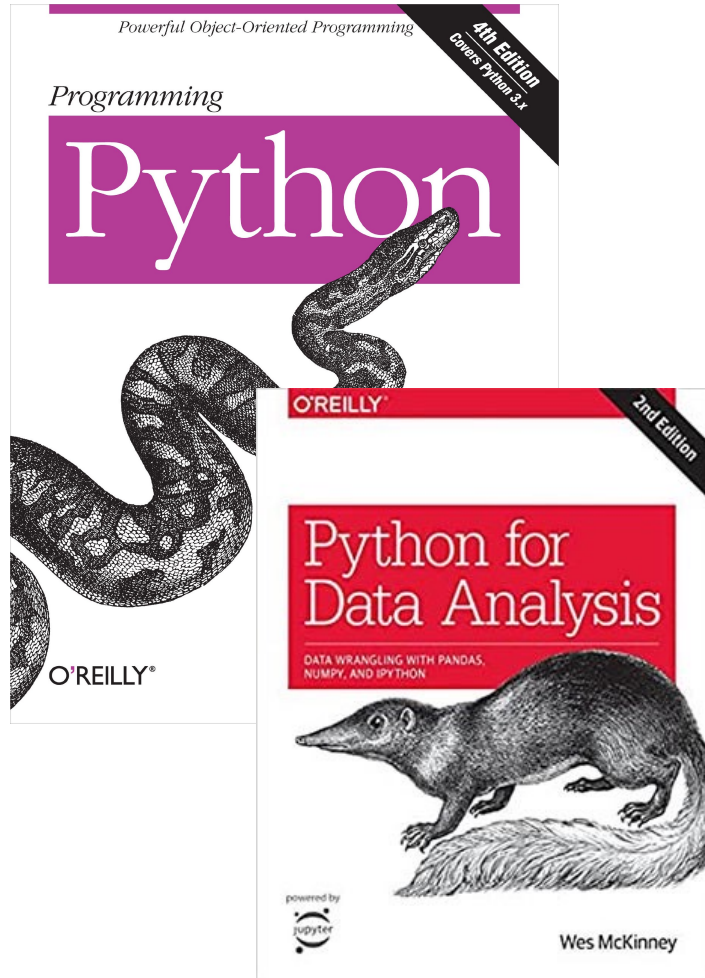
■ Quizzes

- Beginning of Class
 - 5 minutes to complete
 - Simple questions about what was assigned for reading
-

Grading

■ About Readings

- No books required for this course
 - Everything you need is online
 - Part of the life of a developer, is searching for answers and solutions
-



The Internet will do the remembering



Googling
the Regex

Every. Damn. Time.

O'REILLY?

@ThePracticalDev

Cutting corners to meet arbitrary management deadlines



Essential

Copying and Pasting
from Stack Overflow

O'REILLY®

The Practical Developer
@ThePracticalDev

© 2011

It's chaotic, but we make it work



Getting Stuff
to Work

The Practical Developer
@ThePracticalDev

Grading

- Group Project
 - Groups of 3
 - Build your own CLI program
 - Due at the end of the semester
 - Will be submitted through Git
-

Lab Computers vs. Personal Laptops

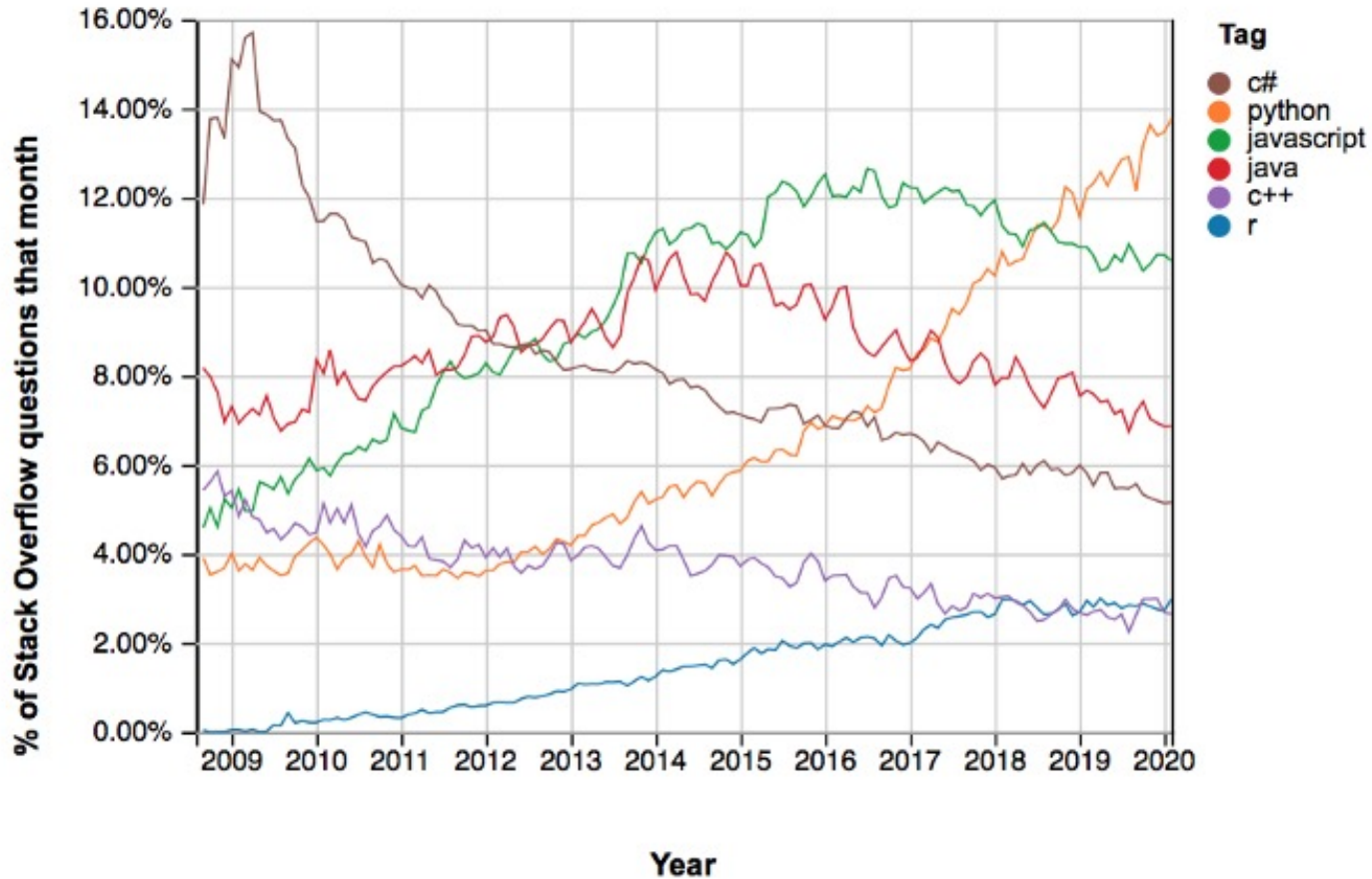
- ❑ Lab Computers
 - Running Windows
 - Available to you

 - ❑ Personal Laptops
 - Running Windows / Mac OS / *Linux*
-

Why Python?

- Mendoza is already using Python for courses
 - Very very popular
 - Easy to learn
 - Cross platform (mostly)
-

Python in the world of Programming



Python in the world of Programming

- Loosely Typed
 - Variables can be any data type (strings, ints, dates, arrays...)
 - Variables can change types
 - Interpreted
 - Write code in Python, run in C (usually)
 - Python code gets ***interpreted*** into C and compiled from there
-

Python in the world of Programming

- Jupyter Notebooks
 - Web interface to Python console
 - Write code and run
 - Python script files
 - Write a script in a .py file
 - Run it with `python myscript.py`
 - Python console
 - Open console with `> python`
 - Write code and run on Enter
-

Python in the world of Programming

- ~~Python Notebooks~~
 - ~~web interface to Python console~~
 - ~~write code and run~~
 - Python script files
 - Write a script in a .py file
 - Run it with `python myscript.py`
 - Python console
 - Open console with `> python`
 - Write code and run on Enter
-

Python in the world of Programming

- Python is slow
 - Python has to work to get to a place that it can run
 - This is overhead
 - Execution Errors
 - Some bugs/errors won't surface until AFTER you run the code
 - Integrated Development Environments (IDE)
 - IDEs for interpreted languages are less powerful and feature rich
-

Interpreted v. Compiled Languages

- Interpreted languages are faster to build and slower to run
 - Interpreted languages can run on most Operating systems
 - Compiled languages must be compiled to Machine Code
 - Dependent on OS and even CPU architecture
-

Interpreted v. Compiled Languages

- Python
 - Ruby
 - Swift
 - R
 - Julia
-

Interpreted v. Compiled Languages

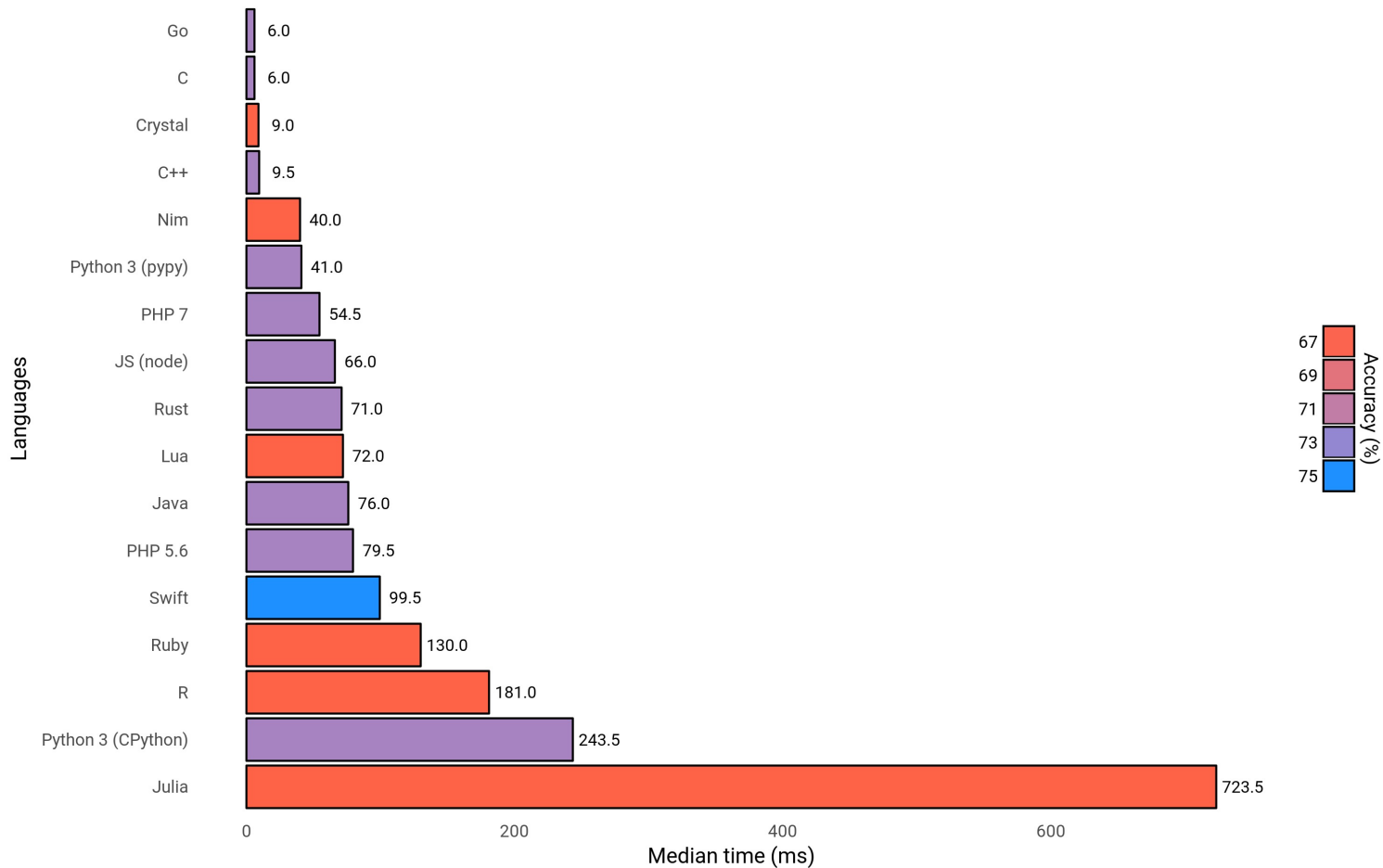
- C
- C++
- Go



Interpreted v. Compiled Languages

Speed comparison of various programming languages

Method: calculating π through the Leibniz formula x times



<https://github.com/niklas-heer/speed-comparison>

<https://github.com/niklas-heer/speed-comparison>

Strongly v. Loosely Typed

```
>>> print('Hello, World!')
Hello, World!
>>> █
```

Strongly v. Loosely Typed

Let's explore how Java "Hello, World!" program works.

Note: You can use our [online Java compiler](#) to run Java programs.

Java "Hello, World!" Program

```
// Your First Program

class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}
```

Output

```
Hello, World!
```

```
jarp — python — 80x24
(base) jarp@jarp-mbp16 ~ % python
Python 3.7.4 (default, Aug 13 2019, 15:17:50)
[Clang 4.0.1 (tags/RELEASE_401/final)] :: Anaconda, Inc. on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Next Class")
Next Class
>>> print("All about the command line")
All about the command line
>>> print("And also a review of Python data structures and syntax")
And also a review of Python data structures and syntax
>>> █
```