CIS192 Python Programming
Wrap-Up Lecture

Harry Smith
University of Pennsylvania
April 26, 2017
Outline

1. Quick Challenges
   - Last Pythonic Tips
   - Code Golf

2. Looking Forward
   - CIS 192 in Review
   - Big Takeaways
   - Next Steps
A few neat tricks

- **Pseudo-quaternions**
  - `[res] if [condition] else [other_res]`
  - `return(b,a) [a<b]`

- **Unpacking**
  - `a=b=c=0 instead of a = 0, b = 0, c = 0`
  - `a,b,c = "123" instead of a = "1", b = "2", c = "3"`

- **Beautiful conditional**
  - `if 1 < a < 3 > b > 2 does what you hope it does`

- `'n` converts integer n to a string!
Outline

1. Quick Challenges
   - Last Pythonic Tips
   - Code Golf

2. Looking Forward
   - CIS 192 in Review
   - Big Takeaways
   - Next Steps
What’s that??

- Write a program in the fewest number of bytes!
- https://ethproductions.github.io/bytes/?e=utf-8
Challenge 1

- Write a program that prints the decimal numbers from 1 to 100 inclusive. But for multiples of three print Fizz instead of the number and for the multiples of five print Buzz. For numbers which are multiples of both three and five print FizzBuzz.
Write a tree data structure class. The class must support adding nodes, printing the entire tree in some reasonable way, and getting a subtree rooted at a given node. Further, the tree should have distinguishable nodes and leaves.
Count the numbers between 1 and N inclusive that have consecutive equal digits.
Outline

1. Quick Challenges
   - Last Pythonic Tips
   - Code Golf

2. Looking Forward
   - CIS 192 in Review
   - Big Takeaways
   - Next Steps
Topics We’ve Covered

- Python Basics & Fundamentals
- Functional Programming
- Object-Oriented Programming
- Iterators, Generators, Exceptions & IO
- Regular Expressions & Other Modules
- HTTP Requests / HTML Parsing
- Data Analysis
- Machine Learning
- Natural Language Processing
- Web Apps
- Artificial Intelligence
- Probability & Simulations
Outline

1. Quick Challenges
   - Last Pythonic Tips
   - Code Golf

2. Looking Forward
   - CIS 192 in Review
   - Big Takeaways
   - Next Steps
If there’s anything I want you to remember...

- Generators! Use this space-efficient tool in place of iterators.
- Lambdas & Functional Programming: they look scary but are incredibly useful
  - Comparators
  - Elements of good design: strategy design pattern
- Make your own classes!
- Using "with open ... as f:" to safely open files!
- List/Set/Generator Comprehensions
- Dictionaries
- Use pip, installing binaries, and taking advantage of libraries
- The power of Python function headers (default kwargs, arbitrarily many positional args).
- Testing code is easy
- Use Decorators to avoid repeating yourself
- Python is excellent for scripting. It’s very useful for solving simple math/probability problems.
- Python makes coding fun.
Outline

1. Quick Challenges
   - Last Pythonic Tips
   - Code Golf

2. Looking Forward
   - CIS 192 in Review
   - Big Takeaways
   - Next Steps
Scratching the Surface

- Each special topic has MUCH more depth than what we’ve covered this semester.
- Many topics we haven’t mentioned
  - Parallel & Distributed Computing
  - Concurrency
  - Graphical User Interfaces
  - etc.
- Check the website for access to some of these.
What’s Next?

- I hope you find the skills you’ve acquired from CIS 192 useful!
- Build your own side-projects, big or small!
- Learn more about Python!
  - PyCon conference recordings
  - Obey the Testing Goat (TDD in Python + Web Dev)
  - New Coder: practical tutorials in Python
- Join the Python community!
- /r/Python subreddit
- Trending GitHub Python repositories
- StackOverflow Python questions