CIS192 Python Programming
Wrap-Up Lecture

Harry Smith

University of Pennsylvania

December 6, 2017
Outline

1. A Brief Discussion on Projects
   - Development Strategies

2. Looking Forward
   - CIS 192 in Review
   - Last Pythonic Tips
   - Big Takeaways
   - Next Steps
Personal Development

- When you started your projects, what did you do?
- As you get ready to submit them, how do you know that they work?
Idealized Personal Development

- When you start a project, what **should** you do?
- As you get ready to submit them, how do should you verify that they work?
Practical Idealized Personal Development

- When you start a project, what **can** you **reasonably** do?
- As you get ready to submit them, how do you verify that they work?
- Food for thought
A Proposed Solution

- Test Driven Development
- Blend unit tests and functional tests to create a story
- Do NOT write code until you write tests
- Have fun with it, it will keep you from going crazy.
1. A Brief Discussion on Projects
   - Development Strategies

2. Looking Forward
   - CIS 192 in Review
   - Last Pythonic Tips
   - Big Takeaways
   - Next Steps
Outline

1. A Brief Discussion on Projects
   - Development Strategies

2. Looking Forward
   - CIS 192 in Review
   - Last Pythonic Tips
   - 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
  - Or maybe just what you fear it does.

- `'n` converts integer n to a string!
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. A Brief Discussion on Projects
   - Development Strategies

2. Looking Forward
   - CIS 192 in Review
   - Last Pythonic Tips
   - 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. A Brief Discussion on Projects
   - Development Strategies

2. Looking Forward
   - CIS 192 in Review
   - Last Pythonic Tips
   - 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
What *Isn’t* Next?

- Build a game!
  - PyGame book
  - More rudimentary game advice
What *Isn’t* Next?

- Manage a database!
  - Advice on database management
What *Isn’t* Next?

- Learn Django!
  - Django Tutorial
  - Mozilla’s Guides (very nice)
What *Isn’t* Next?

- Deploy Applications!
  - Fabric (for SSH/sysadmin)
  - Launch Flask Apps on AWS
  - Launch Django Apps on AWS
What *Isn’t* Next?

- Build an interpreter
- Write your own programming language
- Build a theorem prover
- Automate boring computer tasks
- Scrape the web
- Do beautiful data analysis
- Generate other code automatically
- Handle 3D animation and modeling in Maya
- Send emails automatically
- Quickly create UI
- Build maps and GIS
- Control traffic
- Simulate drug action