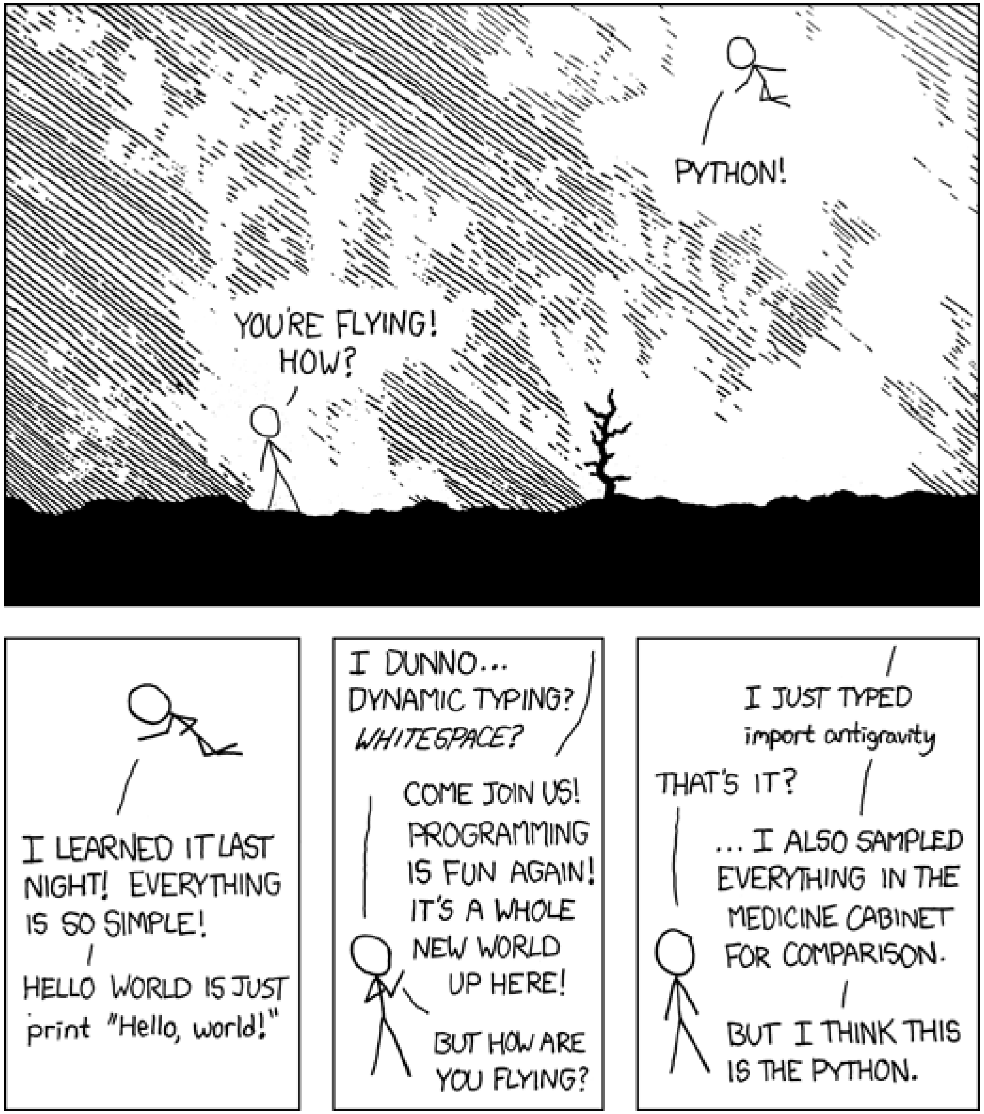
CS 240
Fall 2014

Mike Lam, Professor


PYTHON!
YOU'RE FLYING! HOW?
I LEARNED IT LAST NIGHT! EVERYTHING IS SO SIMPLE!
HELLO WORLD IS JUST print "Hello, world!"
I DUNNO... DYNAMIC TYPING? WHITESPACE?
COME JOIN US! PROGRAMMING IS FUN AGAIN! IT'S A WHOLE NEW WORLD UP HERE!
BUT HOW ARE YOU FLYING?
I JUST TYPED import antigravity
THAT'S IT?
... I ALSO SAMPLED EVERYTHING IN THE MEDICINE CABINET FOR COMPARISON.
BUT I THINK THIS IS THE PYTHON.

Just-for-fun survey: `http://strawpoll.me/2421207`

Today

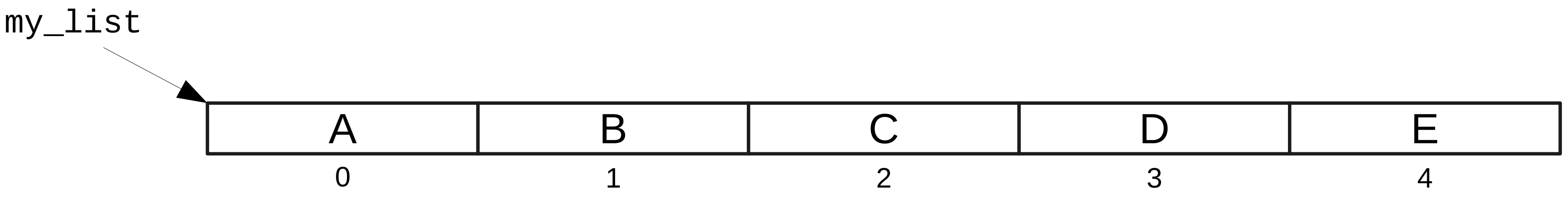
- Course overview
- Course policies
- Python

Motivation

- Computers are digital
  - Data is stored in binary format (1's and 0's)
  - Assembly programming uses this directly
  - This is tedious and error-prone!
- High-level languages (like Java) help
  - Variables (where are the bits located?)
  - Data types (what do the bits mean?)
  - But they don't inherently solve all problems

Basic Example

- Arrays vs. Linked Lists

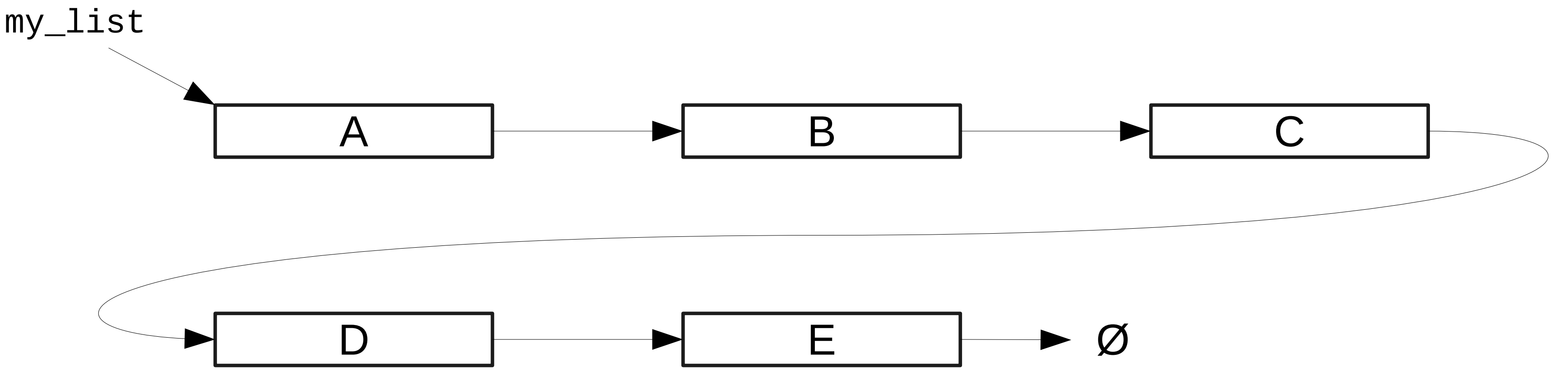


my_list

A B C D E

0 1 2 3 4

vs.



my_list

A B C

D E Ø

Overview

- Data structures
  - Abstractions for organizing data on a computer
    - “How is your data organized?”
  - Generally, the goal is to be as efficient as possible
  - Topics: arrays, linked lists, stacks, queues, trees, graphs, maps, hash tables

Overview

- Algorithms
  - Step-by-step procedure to solve a problem
    - “How exactly do I do ____?”
  - Generally, the goal is to be as efficient as possible
  - Topics: recursion, asymptotic analysis, Big-O notation, searching, sorting

Overview

- Lots of crossover
  - Most data structures are designed to help solve particular types of problems
  - Many algorithms take advantage of particular data structures

Key Concepts

- Abstract data types
  - Generic classes of data structures
  - Often provided with a language's standard library
  - Two goals: efficiency and reusability
- Algorithm analysis
  - Measuring efficiency (what metrics to use?)
  - Comparing algorithms (which is faster?)
  - Empirical (test it!) and analytical (math!)

Goals

- Describe, implement, and analyze common data structures and algorithms
- Choose appropriate structures and algorithms to solve real-world problems
- Secondary goal: become familiar with a programming language that is not Java

Syllabus

- It's online:
  - `http://w3.cs.jmu.edu/lam2mo/cs240/syllabus.html`
- Read it!
  - Especially the parts marked in red
- The textbook is required
  - “Data Structures and Algorithms in Python”
    - Goodrich, Tamassia, Goldwasser
  - Available on Safari online (not always accessible)

Course Policies

- Attendance is not mandatory
  - If the class periods are not worth attending, tell me so that I can make them better!
- Slides will be posted on the website
  - Don't waste time writing down stuff from the slides
- Please silence your cell phones during class

Course Policies

- Submit programming projects as specified in the project description
  - No thumb drives, CDs, or emails (unless requeste
- Project grading will be based on automated tes results
- Late submissions up to 72 hours will receive a 10% penalty per 24 hr period

Course Policies

- The JMU Honor Code applies on ALL assignments
  - We will use software to detect plagiarism
  - Violations may be sent to the honor council
- All submitted code must be YOUR work entire
  - You may work in groups to discuss assignments (in fact, I encourage this), but do NOT share code!

Course Grades

Homework Assignments	30%
Programming Projects	25%
Midterm Exams	30%
Final Exam	15%

Course Policies

- Exams will be held in ISAT 243
- Final exam times are on the website
- If you ask for a re-grade, I may re-grade the entire assignment
  - This applies to homework and projects, too
- If you have to miss a due date or exam because of an excused absence, let me know ASAP

Course Website

- This is the main course website:
  - `http://w3.cs.jmu.edu/lam2mo/cs240/`
- Lots of useful stuff:
  - Syllabus
  - Calendar
  - Assignments
  - Resources
- Check it regularly!

Online Systems

- Canvas
  - Grades and online quizzes
- Piazza (accessed via Canvas)
  - Q&A and discussions
- Web-CAT
  - Project submissions
  - Not quite ready yet
- Make sure you have accounts on all of these!

Questions?

Python

- We will use Python 3 for all code examples an programming projects
- Important websites:
  - `www.python.org` - main Python website
    - For installation and documentation
  - `www.pythontutor.com` - web-based testbed
    - For quick experimentation and learning

Python

- Python is a “scripting language”
  - Interpreted, not compiled
    - Although compilers do exist
  - Dynamic typing
    - Only runtime objects have types
  - Good for quick prototyping

Python Example

```python
value = 1
for x in range(11):
    print("2 ^ "+ str(x) +" = "+ str(value))
    value *= 2
```

Why Python?

- It's a very popular scripting language
- It's available on most platforms
- It's relatively easy to learn
- There are many utility libraries
- Widely-used
  - YouTube, Reddit, Digg, Pinterest,Yelp, ILM, Rackspace, Phillips, Honeywell, etc.

Why Python 3?

- Backwards-incompatible
- Designed to fix fundamental design flaws
- Some major changes:
  - `print "hello"` → `print("hello")`
  - all strings are Unicode
  - iterators instead of lists in many places
  - removed old class implementation
- 2to3 utility for semi-manual conversion

Python

- Whitespace is important
  - These are different:

```python
if a < b:
    print("less than")
    print("good to go!")
```

```python
if a < b:
    print("less than")
print("good to go!")
```

- Be careful!
  - Use a Python-aware editor (some suggestions are on the website)

Python

- We will use the PEP 8 style guide:
  - `http://legacy.python.org/dev/peps/pep-0008/`
  - Link is also on the class website (“Resources”)
    - See also the summarized version
- Keep your code clean
- Include documentation
  - Use docstrings (we will discuss this later)

Learning Python

- We will spend a couple of weeks learning Python
  - Hybrid lecture/labs and dedicated lab time
  - Major programming project TBD
- There are MANY tutorials available online if you wish to move faster or if you are having trouble with particular topics
- I recommend spending lots of time outside class experimenting
- You will also need an IDE or a text editor and console

Homework

- If you have a personal computer, install Python 3 and either 1) an IDE or 2) text editor and console
  - Alternatively, go into ISAT 248 and familiarize yourself with the software on the lab computers
  - Change the code snippet from the earlier slide to calculate Fibonacci numbers rather than powers of two
- Complete course survey and reading quiz by Wednesday (both are on Canvas)

Contacting Me

- Questions? Try Piazza!
- Email: `lam2mo`
  - I will attempt to respond as quickly as possible, but do not expect a response in under 24 hours
- Office: ISAT 227
  - Office hours TBD
    - Please fill out the course survey
  - Appointments preferred outside office hours

Good luck!

- Have a great semester!