

Plan For The Day (PFTD)

- **Practice solving problems**
 - Some solved with a computer, some with Python
 - Differences in solving non-computing problems?
- **Learning about vocabulary and sentences**
 - We'll work with English and Python
- **Practice using tools for Duke Compsci courses**
 - Eclipse, APT, ambient, Skulpt, Python-tutor
 - Sakai, Piazza, Feedback
- **Reveling in the wonder of thinking and working**
 - How do we know when something works?

Compsci 101, Fall 2012

2.1

What is an APT? cheeseburger APT

- **Automated/Algorithmic Problem Test**
 - Write one function, 2-30 lines, solve a problem
 - Tested automatically in Eclipse or the browser
 - Test test test ... Quality of code not a factor in grading
- **Start simple, build toward more complex**
 - What is a function? A function call?
 - What is a parameter? Argument?
 - How do you execute a program?
 - Run is a synonym
- Skulpt and Python Tutor
 - See course web pages



Compsci 101, Fall 2012

2.2

Vocabulary, grammar, rules: Python

- **Naming**
 - The power of abstraction and parameterization
 - What is abstraction?
 - What are parameters? What has them?
- **Types**
 - What's used in computing? What's used in Python?
 - Determine names of types in Python
- **Expressions and operators in Python**
 - Arithmetic: +, -, *, /, %, **, ...
 - Boolean: <, ==, >, and, ...
 - String: +, *, [], [:], [::]

Compsci 101, Fall 2012

2.3

Variables, Types, Values

- **Variable is a name associated with "container/stuff"**
 - Assign a value: `x = 5`
 - Print value of variable: `print x`
 - Use variable in expression: `y = x * 55`
- **String is a types and has a value**
 - Assign: `x = "hello"`
 - Print value of variable: `print x`
 - Use in expression
 - `print len(x)`
 - `print x + " world"`
- **There are more types, this is enough to start!**



Compsci 101, Fall 2012

2.4

Types and values in Python

- **Numbers are important, but not everything is a ...**
 - What is a number? In mathematics, in Python, in Java,
 - Integers, floating-point numbers, complex numbers, ...
 - In Python and other languages, integers are smaller/faster, but you don't need to know this now!
 - 1,2,3 compared to 3.1415, 1.75 compared to 3 + 5i
- **Strings are sequences of characters, "python.org"**
 - Somewhere these are converted to numbers: 0's and 1's
 - No real need to know this now.
- **In Python different things done to numbers/strings**

Compsci 101, Fall 2012

2.5

Expressions, Operators, Types

- **Why is $3+5*4$ different than $(3+5)*4$?**
 - Where can you find information about precedence?
- **Why is $5/3$ different than $5.0/3$?**
 - What will happen in Python 3? Accommodate in 2.7?
- **What happens when operators go bad?**
 - What is "apple" + 3? What is "apple" + "pi"?
 - What is "apple" * 3? What is "apple" * "pi"?
- **What is a variable in Python?**
 - Does it have a name? Does it have a type?

Compsci 101, Fall 2012

2.6

Latanya Sweeney

I am a computer scientist with a long history of weaving technology and policy together to remove stakeholder barriers to technology adoption. My focus is on "computational policy" and I term myself a "computer (cross) policy" scientist. I have enjoyed success at creating technology that weaves with policy to resolve real-world technology-privacy clashes.



<http://latanyasweeney.org/>

Identify 87% of US population using (dob,zip,gender). Director of Harvard Data Privacy Lab, instrumental in HIPAA because of *de-identification* work

Compsci 101, Fall 2012

2.7

Counting words in a file: Python redux

```
name = "/data/poe.txt"
ff = open(name)
st = ff.read()
words = st.split()
print "# words in", name, "=", len(words)
```

- **What are the *names* in the code above?**
 - Why are names important?
- **What are the *types* in the code above?**
 - How do we get Python to help us answer this question
- **How do we re-use this code more generally**
 - The power of names! The power of functions!

Compsci 101, Fall 2012

2.8

Names and Types summarized

- There are *rules* for what a valid name is in Python
 - In addition there are *conventions* we will use for names
- In code shown we see *variables, constants, functions, and methods*
 - This is more vocabulary, talking Python to others?
 - What are each of those italicized words?
 - `type()` and interactive console/interpreter
- Types for variables and expressions
 - We see file, string, list, int, float; later: list, set, and more
- Always ask: what's name, what's type, what's value?

Compsci 101, Fall 2012

2.9

Interlude

- Use word-counting code in Eclipse
 - Python console, type and see
 - Using names, what is a .py file, user-defined functions
 - Modules and functions: re-use with minimal re-typing
 - Function is abstraction, parameterization over code
 - Module is abstraction over functions
- Python functions and expressions, practicing solving problems
 - APTs for next week
 - Gravity, cheeseburger, Bogword
 - BMI, Heron's formula

Compsci 101, Fall 2012

2.10

Getting ready to code in Python

- We need a programming environment
 - Eclipse, PyDev, Python, Ambient
 - Open source or free for academics
- We need a computer with an operating system
 - Installing the suite of tools can be cumbersome
 - We'll help, when things don't work, start over ☹
- Getting used to the environment can take time
 - Once you've got it, second nature!
 - Easy to reuse with a new language (and some new tools)

Compsci 101, Fall 2012

2.11

Nathan Myhrvold



● Who is he?

- <http://bit.ly/myhrvold>

We invent for fun. Invention is a lot of fun to do. And we also invent for profit. The two are related because the profit actually takes long enough that, if it isn't fun you wouldn't have the time to do it.

<http://bit.ly/zwlqxn>

Compsci 101, Fall 2012

2.12