

Plan For The Week (PFTW)

- **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
 - Sakai, Piazza, Feedback
- **Reveling in the wonder of thinking and working**
 - How do we know when something works?

Compsci 6/101, Spring 2012

2.1

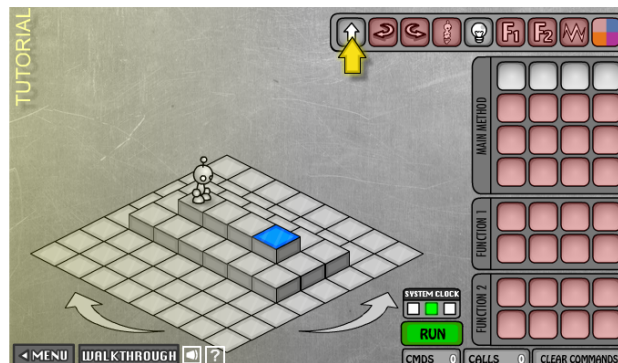
What else will we do this week?

- **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 6/101, Spring 2012

2.2

Lightbot 2.0, <http://bit.ly/litebot>



Compsci 6/101, Spring 2012

2.3

Commands in Lightbot 2.0

- **Forward, Jump, Turnleft, Turnright, Light, F1, F2**
 - Functions are abstractions: group commands for re-use
 - Practice with Basics [teleportation/ifs not needed]
 - Practice with <http://armor.ag/1TKrh>
 - Shows the need for functions
 - How many moves needed without functions?
- **Why are we using this?**
 - Community aspects, low-overhead/in-browser
 - Practice puzzle-like problem-solving
 - Turns out this isn't for everyone, but nothing is

Compsci 6/101, Spring 2012

2.4

Name and types in Lightbot

- **Commands have names**
 - Move, turn-right, and so on
- **Commands have an effect on the world**
 - Move robot, light squares
 - No "error" conditions, robot doesn't die or crash
- **Grouping commands into functions**
 - Un-informative names: F1 and F2
 - Functions are for a specific world, unusable between them

Counting words in a file: Python redux

```
name = "/data/poe.txt"
file = open(name)
str = file.read()
words = str.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!

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**

Names and Types summarized

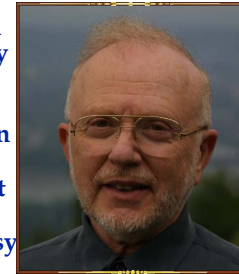
- **There are *rules* for what a valid name is in Python**
 - In addition there are *conventions* we will use for names
- **In code shown we see *variables, constants, functions, and methods***
 - This is more vocabulary, talking Python to others?
 - What are each of those italicized words?
 - `type()`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

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
 - APIs for next week
 - [BMI](#)
 - [Heron's formula](#)

David Parnas (entry in [Wikipedia](#))

"For much of my life, I have been a software voyeur, peeking furtively at other people's dirty code. Occasionally, I find a real jewel, a well-structured program written in a consistent style, free of kludges, developed so that each component is simple and organized, and designed so that the product is easy to change. "



"We must not forget that the wheel is reinvented so often because it is a very good idea; I've learned to worry more about the soundness of ideas that were invented only once. "

Expressions, Operators, Names

- 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 2.6?
- What happens when operators go bad?
 - What is "apple" + 3? What is "apple" + "pi"?
- What is a variable in Python?
 - Does it have a name? Does it have a type?

Alma Whitten

- Google: Engineering Lead for Privacy, Director for Privacy
 - Across marketing and engineering
 - Why Johnny Can't Encrypt



"It's more and more the case that every individual is going around with a cheap yet powerful data-capture device, and the ability to connect that device to powerful data services.

There's a whole interesting minefield to be picked through,"
(cnet news, October 2010)