Announcements

Can already program, just want to pick up Python? Take CS1133, Transition to Python, instead. 5 weeks S/U; doesn't fulfill college requirements. Contact Craig Frey, ccf27@cornell.edu, for more info.

AEWs start this week. Can still enroll on Student Center. Contact Anshul Sacheti (AEW Lead), as885@cornell.edu, and/or follow the link from the CS1110 webpage for more info.

Added late, missed lab 1? Do it on your own ASAP, bring to your section TA to check it in at *the beginning of this week's lab*. Everything, including lecture video and handouts, at http://www.cs.cornell.edu/courses/cs1110/2013sp

Lab 1 grades posted on CS1110 CMS. Yours missing? Contact your lab TA (Who? see Times & Places page). Not in CS1110 CMS? See FAQ page.

Moved to a later lab as requested to make room for other students? *THANK YOU!!*

Readings for this week

- 2.6, python files (referred to there as "scripts"); 2.9, comments; 4.9, docstring
- 3.3, for `import` and dot notation
- 8.1, 8.2, 8.4, 8.5, 8.8, about string operations, including the dot notation. Don't worry about the "method" terminology (yet).
- 3.1-3.6, about functions

[xkcd.com]
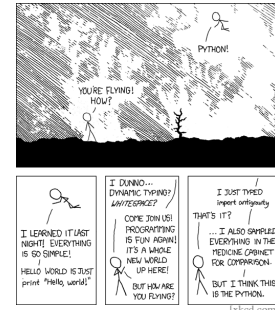
Operations for Extracting Data from Strings
(from last lecture's handout)

- s = 'abc d'

      0 1 2 3 4
    | a | b | c | | d |

- Access portions with []
  - s[0] is 'a'
  - s[4] is 'd'
  - s[5] causes an error
  - s[0:2] is 'ab' (excludes c)
  - s[2:] is 'c d'
- Called "string slicing"

s = 'abracadabra'

A '#' marks a *comment* for the reader *(including the code's author)*. Python ignores the rest of the line.

```
# the following all evaluate to True
'a' in s == True
'cad' in s == True
'foo' in s == False
s.index('a') == 0
s.index('rac') == 2
s.count('a') == 5
len(s) == 11
s.strip('a') == 'bracadabr'
' cs1110 '.strip() == 'cs1110'
```

A String Puzzle

Given: variable data contains a string with at least two 'L's.
> *Example*: data='PROF. LILLIAN LEE'

Goal: give an expression for the part of the string starting with the **2ⁿᵈ** 'L'. *(How can we use the index operation?)*

(1) Store in variable i the index of the first 'L'.

(2) Store in variable tail the part of data starting *after* i

(3) Give an expression for the part of tail starting with 'L'

Given: `info` contains a comma-separated string with last name, difficulty, execution, and penalty.
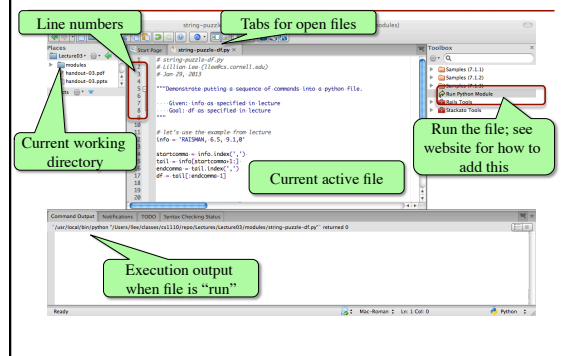 - *Example:* info = 'RAISMAN, 6.5, 9.1,0'

Goal: store the difficulty as a string, with no extra spaces or punctuation, in variable df

Where, in the following sequence of commands, does the first (conceptual) error occur?

A: `startcomma = info.index(',')`
B: `tail = info[startcomma+1:]`
C: `endcomma = tail.index(',')`
D: `df = tail[:endcomma-1]`
E: this sequence achieves the goal

Writing a Python File in Komodo

Line numbers

Tabs for open files

Current working directory

Current active file

Run the file; see website for how to add this

Execution output when file is "run"

Req'd Format for CS1110 Python Files

```
# string-puzzle-df.py
# Lillian Lee (LLEE@cs.cornell.edu)
# Tue Jan 29, 2013
```

Header: file name, authoring info

```
""" Demonstrates putting a sequem
    of commands into a ...."""
```

Docstring: *note the triple quotes*. Multi-line comment explaining the purpose & function of the file.

```
x = 1+2
x = 3*x
x
```

Note: Unlike with the command prompt, evaluating an expression produces nothing when a Python file (script, module) is run. The author probably wanted `print x` here.

Function Calls

- Python supports expressions with math-like functions
- Function expressions have the form **fun**(x,y,…)

function name   **argument**

- Examples of *built-in* functions:
  - Type conversion: int(), float()
  - Dynamically type an expression: type()
- Getting user input: raw_input()
- Help function: help()

Using a Function From Another File
(such files are called *modules*)

Example: what if we want 'Raisman', not 'RAISMAN'?

Lucky us: someone has written a module (file) string that contains a function capwords.

```
import string          # Tell Python to access this module
name = info[:info.find(',')]   # name is 'RAISMAN'
print string.capwords(name)  # output is 'Raisman'
```

Python Comes with Many Modules

- **io**
  - Read/write from files
- **math**
  - Mathematical functions
- **random**
  - Generate random numbers
  - Can pick any distribution
- **string**
  - Useful string functions
- **sys**
  - Information about your OS

- Complete list:
- http://docs.python.org/library
- **Library**: built-in modules
  - May change each release
  - Why version #s are an issue

Reading the Python Documentation