Welcome to Berkeley Computer Science!
The Course Staff

TAs hold discussion sections, labs, and office hours

Readers are your personal programming mentors

Lab Assistants ensure that you don’t get stuck for too long
An Introduction to Computer Science
What is Computer Science?

The study of computer science involves understanding:
- What problems can be solved using computation,
- How to solve those problems, and
- What design choices lead to effective solutions.

Systems
Artificial Intelligence
Graphics
Security
Networking
Programming Languages
Theory
Scientific Computing

Games
Robotics
Natural Language Processing

...
What is This Course About?

• A course about managing complexity
  ▪ Mastering abstraction
  ▪ Programming paradigms
  ▪ Not about 1’s and 0’s

• An introduction to Python
  ▪ All the features we really need: introduced today
  ▪ Understanding through implementation
  ▪ How computers interpret programming languages

• A challenging course that will demand a lot of you
What is This Course About?

Plone Conference. Photo courtesy of Kriszta Szita
Alternatives to This Course

CS 61AS: Self-paced 61A

CS 10: The Beauty and Joy of Computing
Course Policies

The purpose of this course is to help you learn

The staff is here to make you successful

All the details are online:

http://inst.eecs.berkeley.edu/~cs61A/fa13/about.html
Collaboration

• Discuss everything with each other
• **EPA**: Effort, participation, and altruism
• Homework can be completed with a partner
• Projects *should* be completed with a partner
• Find a project partner in your section (if you can)

The limits of collaboration

• One simple rule: Don’t share your code, except with partners
• Copying project solutions is a serious offense!
• We really do catch people who violate the rules
  • We also know how to search the web for solutions
  • We let computers detect copying for us
Expressions
Types of expressions

An expression describes a computation and evaluates to a value

\[ \frac{6}{23} \]

\[ \sin \pi \]

\[ \sqrt{3493161} \]

\[ f(x) \]

\[ 18 + 69 \]

\[ \frac{100}{i=1} i \]

\[ | - 1869| \]

\[ (69) \]

\[ (18) \]
Call Expressions in Python

All expressions can use function call notation
(Demo)
Anatomy of a Call Expression

Evaluation procedure for call expressions:

1. Evaluate the operator and operand subexpressions

2. Apply the function that is the value of the operator subexpression to the arguments that are the values of the operand subexpression
Evaluating Nested Expressions

\[ \text{mul} \left( \text{add}(2, \text{mul}(4, 6)), \text{add}(3, 5) \right) \]

\[ \text{mul}(26, 8) \]

\[ \text{add}(2, \text{mul}(4, 6)) \]

\[ \text{mul}(24) \]

\[ \text{add}(3, 5) \]

\[ \text{mul}(4, 6) \]

\[ \text{mul}(24, 8) \]
Evaluating Nested Expressions

Operand 0 “subexpression”

Expression tree
Data, Functions, and Interpreters

**Data:** The things that programs fiddle with

“The Art of Computer Programming”
2
Shakespeare’s 37 plays
Donald Knuth
(Ka–NOOTH)

**Functions:** Rules for manipulating data

Count the words in a line of text

Add up numbers
Pronounce someone’s name

**Interpreter:** An implementation of the procedure for evaluation