ACM/CS 114
Parallel algorithms for scientific applications

Michael A. G. Aïvázis

California Institute of Technology

Spring 2012

Languages and programming paradigms

- a very active area of research
  - dozens of languages and runtime environments of the last 50 years
- the survivors:
  - procedural programming, and its offspring structured programming
  - functional programming
  - object oriented programming
- current areas of research:
  - component oriented programming
  - aspect programming
- languages are important:
  - they reflect an approach to computing
  - they shape what is easily expressible
- we'll take a quick tour of python
  - resources: `www.python.org`
  - overview of the language
  - interactive sessions with the interpreter
  - building extensions in C/C++

A python script

- python reads like pseudocode
- here is the code for the π estimator using Monte Carlo integration over the quarter disk

```
# get access to the random munber generator functions
import random
# sample size
N = 10**5
# initialize the interior point counter
interior = 0
# integrate by sampling some number of times
for i in range(N):
    # build a random point
    x = random.random()
    y = random.random()
    # check whether it is inside the unit quarter circle
    if (x*x + y*y) <= 1.0: # no need to waste time computing the sqrt
        # update the interior point counter
        interior += 1
# print the result:
print("pi: {0:.8f}".format(4*interior/N))
```

Overview

- built-in objects and their operators
  - numbers, strings, containers
  - files
- statements
  - evaluating expressions, explicit and implicit assignments, logic, iteration
- functions
  - scope rules, argument passing, callable objects
- modules and packages
  - name qualification, importing symbols
- user defined objects
  - declarations and definitions, inheritance, overloading operators
- exceptions
  - raising and catching, exception hierarchies