

MIT OpenCourseWare
<http://ocw.mit.edu>

6.00 Introduction to Computer Science and Programming
Fall 2008

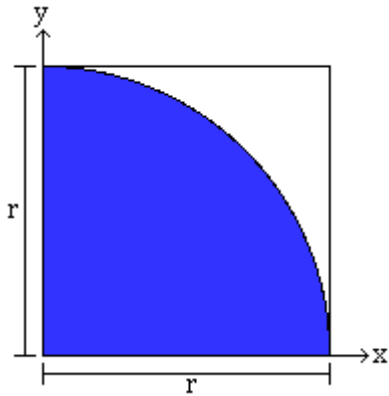
For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.

6.00 Handout, Lecture 20
(Not intended to make sense outside of lecture)

```
from pylab import *
import random, math

def flipTrial(numFlips):
    heads, tails = 0, 0
    for i in xrange(0, numFlips):
        coin = random.randint(0, 1)
        if coin == 0: heads += 1
        else: tails += 1
    return heads, tails

def simFlips(numFlips, numTrials):
    diffs = []
    for i in xrange(0, numTrials):
        heads, tails = flipTrial(numFlips)
        diffs.append(abs(heads - tails))
    diffs = array(diffs)
    diffMean = sum(diffs)/len(diffs)
    diffPercent = (diffs/float(numFlips))*100
    percentMean = sum(diffPercent)/len(diffPercent)
    hist(diffs)
    axvline(diffMean, color = 'r', label = 'Mean')
    legend()
    titleString = str(numFlips) + ' Flips, ' + str(numTrials) + ' Trials'
    title(titleString)
    xlabel('Difference between heads and tails')
    ylabel('Number of Trials')
    figure()
    plot(diffPercent)
    axhline(percentMean, color = 'r', label = 'Mean')
    legend()
    title(titleString)
    xlabel('Trial Number')
    ylabel('Percent Difference between heads and tails')
```



$$\frac{\text{Number in shaded area}}{\text{Number in square}} = \frac{\pi * r^2 * .25}{r^2}$$

$$\pi = \frac{4 * \text{Number in shaded area}}{\text{Number in square}}$$

```
#Tell Python which local standard to use
import locale
locale.setlocale(locale.LC_ALL, 'en_US.UTF-8')

#Format ints according to local standard
def formatInt(i):
    return locale.format('%d', i, grouping = True)

from pylab import *
import random, math

def throwDarts(numDarts, shouldPlot):
    inCircle = 0
    estimates = []
    for darts in xrange(1, numDarts + 1, 1):
        x = random.random()
        y = random.random()
        if math.sqrt(x*x + y*y) <= 1.0:
            inCircle += 1
        if shouldPlot:
            piGuess = 4*(inCircle/float(darts))
            estimates.append(piGuess)
    if darts%1000000 == 0: #So I know it's making progress
        piGuess = 4*(inCircle/float(darts))
        dartsStr = locale.format('%d', darts, True)
        print 'Estimate with', formatInt(darts), 'darts:', piGuess
    if shouldPlot:
        xAxis = arange(1, len(estimates)+1)
        semilogx(xAxis, estimates)
        titleString = 'Estimations of pi, final estimate: ' + str(piGuess)
        title(titleString)
        xlabel('Number of Darts Thrown')
        ylabel('Estimate of pi')
        axhline(3.14159)
    return 4*(inCircle/float(numDarts))

def findPi(numDarts, shouldPlot=False):
    piGuess = throwDarts(numDarts, shouldPlot)
    print 'Estimated value of pi with', formatInt(numDarts), 'darts:', piGuess

findPi(10000, True)
findPi(100000000)
show()
```