- Previous Lecture:
  - Branching
  - Logical operators and values

- Today's Lecture:
  - Iteration using **for**
  - Introduce **while**

- Announcement
  - Register your clicker!
  - Adhere to the Code of Academic Integrity

Question

A stick of unit length is split into two pieces. The breakpoint is randomly selected. On average, how long is the shorter piece?

Physical experiment?
Thought experiment? → analysis
Computational experiment! → simulation

Need to repeat many trials!

February 5, 2008   Lecture 5   3

Repeat n times

```
% one trial of the experiment
breakPt= rand(1);
shortPiece= min(breakPt, 1-breakPt);
```

Take average

Print result

February 5, 2008   Lecture 5   7

```
n= 10000;  % number of trials
total= 0;  % accumulated length so far

for  k= 1:n
    % one trial of the experiment
    breakPt= rand(1);
    shortPiece= min(breakPt, 1-breakPt);
    total= total + shortPiece;
end

aveLength= total/n
fprintf('Average length is %f\n', ...
                           aveLength)
```

February 5, 2008   Lecture 5   8

Pattern for doing something *n* times

```
n= _____
for k= 1:n

    % code to do
    % that something

end
```

Definite iteration

February 5, 2008   Lecture 5   9

Syntax of the **for** loop

```
for <var>= <start value>:<incr>:<end bound>

    statements to be executed

end
```

Loop body

February 5, 2008   Lecture 5   10

Syntax of the **for** loop

for *<var>* = *<start value>*:*<incr>*:*<end bound>*

> *statements to be executed*

end

Loop header specifies all the values that the index variable will take on, one for each pass of the loop. E.g, **k= 3:1:7** means **k** will take on the values 3, 4, 5, 6, 7, one at a time.

---

for loop examples

```
for k= 2:0.5:3        k takes on the values 2,2.5,3
    disp(k)           Non-integer increment is OK
end
for k= 1:4            k takes on the values 1,2,3,4
    disp(k)           Default increment is 1
end
for k= 0:-2:-6        k takes on the values 0,-2,-4,-6
    disp(k)           "Increment" may be negative
end
for k= 0:-2:-7        k takes on the values 0,-2,-4,-6
    disp(k)           Colon expression specifies a bound
end
for k= 5:2:1
    disp(k)
end
```

Example: "Accumulate" a solution

```
% Average 10 numbers from user input

n= 10;      % number of data values

for k= 1:n
  % read and process input value
    num= input('Enter a number: ');
    total= total + num;
end
ave= total/n;  % average of n numbers
fprintf('Average is %f\n', ave)
```

How many passes through the loop will be completed?
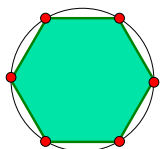
A: 0
A: 1
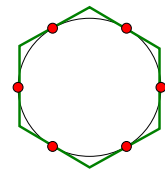A: 9
A: 10
A: 11

Important Features of Iteration

- A task can be accomplished if some steps are repeated; these steps form the loop body
- Need a starting point
- Need to know when to stop
- Need to keep track of (and measure) progress

Example: *n*-gon → circle

Inscribed hexagon
$(n/2) \sin(2\pi/n)$

Circumscribed hexagon
$n \tan(\pi/n)$

As *n* approaches infinity, the inscribed and circumscribed areas approach the area of a circle. How big should *n* be?

Find *n* such that **outerA** and **innerA** converge

First, itemize the tasks:

- *define how close is close enough*
- *select an initial n*
- *calculate innerA, outerA for current n*
- *diff= outerA – innerA*
- *close enough?*
- *if not, increase n, repeat above tasks*