Announcements

- CLUE Tutoring
  - Wednesday nights 7-8:30PM MGH 058
  - 2 extra-credit points for each session you attend from last week through the end of the quarter
  - Sign the attendance list to get credit!

- Veteran's Day on Wednesday
  - Official UW holiday
  - CLUE Tutoring is on Monday night this week only
    - 7-8:30pm
  - If you have Wednesday lab section,
    - Attend a drop-in lab this week
  - Get 2 points extra credit for attending CLUE Tutoring—sign the attendance sheet

- Tour of Living Computer Museum
  - Opens to the public in January
  - Our tours:
    - This week: Thursday, Friday
    - Next week: Monday, Tuesday
    - Signup on WebQ linked from Calendar by Tuesday 10pm
    - Directions on GoPost
    - SODO near Sears & Qwest Field

- Due Tuesday night
  - Labs 6/7
  - Signup for museum tour

- Labs 8/9
  - Thursday this week and Monday/Tuesday labs next week

- The Museum Tour and Labs 8/9 are optional—for extra credit.
  - Choose one or the other
Announcements

- Repeat:
  - D.A.’s office hours have changed and moved to the drop-in lab
  - MGH 430 Tuesday nights 5-6pm
  - I’m always happy to answer questions after lecture, too.

---

Chapter 21 for today

- Handy references for lab
  - The JavaScript Phrasebook
  - W3 Schools JavaScript tutorial

---

Concepts of Algorithmic Thinking

Iterations, or Loops

Once is not Enough

D.A. Clements

Objectives

- Learn the syntax of loops
- Use loops to count down or count up
- Recognize the World-Famous Iteration
- Learn how to start, increment, and end a loop
- Describe the structure of nested loops

---

Iteration, or looping, is the process of repetition:

- looping through a sequence of statements to repeat them

ITERATION

D.A. Clements, MLIS, UW Information School

11/9/2009
Major Types of Iterations

- **For loop**
  1. Baby
  2. Count up
  3. Count down
- **While loop**
  4. Count up
  5. Count down

Try the examples in Week 5 on the course Web site!

Repetition is good

**FOR LOOPS**

The **for** Loop Basic Syntax

```javascript
for (<initialization>; <continuation>; <next iteration>)
{
    <statement list>
}
```

- The whole sequence of statements in the statement list is performed for each iteration
- Computer completes the whole statement sequence of the <statement list> before beginning the next iteration

Control specification

- The three operations in the parentheses of the **for** loop
  - Control the number of times the loop iterates
  - by using an iteration variable (must be declared)

How a **for** Loop Works

- Consider a computation on declared variables `j` and `text`
  ```javascript
text = "She said ";
for ( var j = 1; j <= 3; j = j + 1 )
{
    text = text + "Never! ";
}
alert(text);
```

How a **for** Loop Works

- Consider a computation on declared variables `j` and `text`
  ```javascript
text = "She said ";
for ( var j = 1; j <= 3; j = j + 1 )
{
    text = text + "Never! ";
}
alert(text);
```

Control specification
How a for Loop Works

Consider a computation on declared variables j and text

text = "She said ";
for ( var j = 1; j <= 3; j = j + 1 ) {
    text = text + "Never! ";
}
alert(text);

Starting point

Step size or increment

Continuation condition

Processing for loops

Example:

for ( j = 1; j <= 3; j = j + 1 ) {
    <statement list>
}

The first operation is the <initialization>

Sets the iteration variable's value for the first iteration of the loop. Done only once.

The next operation is <continuation>

Test. If the test has a false outcome, the <statement list> is skipped and control passes to the next statement after the for loop.

If the test has a true outcome, the <statement list> is performed. When the statements are complete, the <next iteration> operation is performed.

Repeats with the continuation test, performs same sequence of steps.

The World-Famous Iteration

for ( j = 0; j < n; j++ ) {
    ...
}

Most frequently written for loop of all time

Easy to see iteration count:

Always n times

When n is 3

0 is first loop

1 is second loop

2 is third loop

3 is fourth and it doesn't run.
Running through a for loop

<table>
<thead>
<tr>
<th>Operation</th>
<th>Operation Result</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>j = 1</td>
<td>Initialize iteration variable</td>
</tr>
<tr>
<td>2</td>
<td>j &lt;= 3</td>
<td>First continuation test, continue</td>
</tr>
<tr>
<td>3</td>
<td>j = j + 1</td>
<td>First continuation test, continue</td>
</tr>
<tr>
<td>4</td>
<td>j = j + 1</td>
<td>Second continuation test, continue</td>
</tr>
<tr>
<td>5</td>
<td>j = j + 1</td>
<td>Third continuation test, continue</td>
</tr>
<tr>
<td>6</td>
<td>j = j + 1</td>
<td>Fourth continuation test, terminate</td>
</tr>
</tbody>
</table>

Table 21.1: The sequence of operations on j from the for loop with control specifications: (j=1; j=3; j=j+1)

Rules: Continuation & Step Size
- Continuation/Termination Test: j <= 3
- Continuation is any expression resulting in a Boolean value (true/false)
- Continuation must involve iteration variable to avoid infinite loop
- Step Size: j = j + 1
- Amount of change from one iteration to the next
- Often called the increment or decrement
  - Increment: j + 1
  - Decrement: j - 1

Experiments with Flipping Coins

```javascript
function coinToss() {
    return Math.floor(Math.random());
}
```
Experiments with Flipping Coins

Demo—100 coin tosses

- Try the Coin Toss
  - Example 6 in Module 6 of our course Web site

Experiment 2—with Five Trials

- A Nested Loop
  - To run several trials, consider the entire loop we just looked at as one Trial
  - Create another for loop containing this Trial unit, adding a couple of needed statements
  - We have a loop within a loop (nested loop) which causes the Trial loop (0-99) to run five times
Experiment 2—outer loop

```java
for (i = 1; i < 10; i++) { // i = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
  if (i % 3 == 0) { // i = 3, 6, 9
    System.out.println("i is a multiple of 3.");
  }
}
```

Experiment 2—declare i and j

```java
int i, j;
```

Experiment 2—set heads, tails to zero

```java
head = 0;
tail = 0;
```

Experiment 2—how far from 50%?

```java
for (i = 1; i <= 10; i++) { // i = 1, 2, 3, ..., 10
  if (i % 2 == 1) { // odd numbers
    head++;
  } else { // even numbers
    tail++;
  }
  System.out.println("Head: "+head+", Tail: "+tail);
}
```

Demo—Five Trials

- Try the Five-Trial Coin Toss
  - Example 7 in Module 6 of our course Web site

Summary

- Learn the syntax of loops
- Use loops to count down or count up
- Recognize the World-Famous Iteration
- Learn how to start, increment, and end a loop
- Describe the structure of nested loops
Quiz topics for next week

- For loops

End papers...

- A computer lets you make more mistakes faster than any invention in human history— with the possible exceptions of handguns and tequila.

~ Mitche Ratcliffe

D.A. Clements, MLS, UW Information School