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!
Announcements

- 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
Announcements

- 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
Announcements

● Due Tuesday night
  ● Labs 6/7
  ● Signup for museum tour
Announcements

- 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
• 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.
Announcements

- Chapter 21 for today
- Handy references for lab
  - *The JavaScript Phrasebook*
  - W3 Schools JavaScript tutorial
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
Play it again, Sam.

ITERATION
Definitions

- Iteration, or looping, is the process of repetition:
  - looping through a sequence of statements to repeat them
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

```
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

D.A. Clements, MLIS, UW Information School
11/9/2009
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)
Consider a computation on declared variables \( j \) and \( \text{text} \)

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

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

(Control specification)
Consider a computation on declared variables \( j \) and \( \text{text} \)

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

**Continuation condition**
How a **for** Loop Works

Consider a computation on declared variables **j** and **text**

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

Step size or increment
How a **for** Loop Works

**Demo:**

```javascript
let text = "The two-year-old said ";
for (let j = 1; j <= 3; j = j + 1) {
  text = text + "No! ";
  alert(text);
}
```
**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

```plaintext
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 21.1 The sequence of operations on `j` from the *for* loop with control specification `(j=1; j<=3; j=j+1)`

<table>
<thead>
<tr>
<th>Operation</th>
<th>Operation Result</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>j = 1</code></td>
<td><code>j</code>’s value is 1</td>
<td>Initialize iteration variable</td>
</tr>
<tr>
<td><code>j &lt;= 3</code></td>
<td><code>true, j</code> is less than 3</td>
<td>First <code>&lt;continuation&gt;</code> test, continue</td>
</tr>
<tr>
<td><code>j = j + 1</code></td>
<td><code>j</code>’s value is 2</td>
<td>First <code>&lt;next iteration&gt;</code> operation</td>
</tr>
<tr>
<td><code>j &lt;= 3</code></td>
<td><code>true, j</code> is less than 3</td>
<td>Second <code>&lt;continuation&gt;</code> test, continue</td>
</tr>
<tr>
<td><code>j = j + 1</code></td>
<td><code>j</code>’s value is 3</td>
<td>Second <code>&lt;next iteration&gt;</code> operation</td>
</tr>
<tr>
<td><code>j &lt;= 3</code></td>
<td><code>true, j</code> is equal to 3</td>
<td>Third <code>&lt;continuation&gt;</code> test, continue</td>
</tr>
<tr>
<td><code>j = j + 1</code></td>
<td><code>j</code>’s value is 4</td>
<td>Third <code>&lt;next iteration&gt;</code> operation</td>
</tr>
<tr>
<td><code>j &lt;= 3</code></td>
<td><code>false, j</code> is greater than 3</td>
<td>Fourth <code>&lt;continuation&gt;</code> test, terminate</td>
</tr>
</tbody>
</table>
The Iteration Variable: $j = 1$
- Must be declared, and follow rules for variable identifiers
- $i$, $j$, and $k$ are the most common choices

The Starting Point
- Iteration can begin anywhere, including negative numbers
Rules: Continuation & Step Size

- Continuation/Termination Test $j \leq 3$
  - Test 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

```html
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
<head>
    <meta http-equiv="content-type"
          content="text/html; charset=utf-8" />
    <meta http-equiv="Content-Style-Type" content="text/css" />
    <title>For loop example with coin toss</title>
    <script type="text/javascript">
        var heads=0, tails=0;          // Counters
        var i;                         // Iteration variable
        for (i=0; i<100; i++) {
            if (randNum(2) == 1)
                heads++;
            else
                tails++;
        }
        alert("Heads: " + heads + " and Tails: " + tails);
        function randNum(range)
        {
            return Math.floor(range*Math.random());
        }
    </script>
</head>
</html>
```
Experiments with Flipping Coins

```html
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
  <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <meta http-equiv="Content-Style-Type" content="text/css" />
    <title>For loop example with coin toss</title>
    <script type="text/javascript">
      var heads = 0, tails = 0; // Counters
      var i; // Iteration variable
      for (i = 0; i < 100; i++)
      {
        if (randNum(2) == 1)
          heads++;
        else
          tails++;
      }
      alert("Heads: \" + heads + \" and Tails: \" + tails);
      function randNum(range)
      {
        return Math.floor(range*Math.random());
      }
    </script>
  </head>
</html>
```
Experiments with Flipping Coins

```html
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
<head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
    <meta http-equiv="Content-Type" content="text/css"/>
    <title>For loop example with coin toss</title>
    <script type="text/javascript">
        var heads = 0, tails = 0; // Counters
        var i; // Iteration variable
        for (i = 0; i < 100; i++) {
            if (randNum(2) == 1) {
                heads++;
            } else {
                tails++;
            }
        }
        alert("Heads: " + heads + " and Tails: " + tails);
        function randNum(range) {
            return Math.floor(range * Math.random());
        }
    </script>
</head>
</html>
```
Experiments with Flipping Coins

```html
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
<head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <meta http-equiv="Content-Style-Type" content="text/css" />
    <title>For loop example with coin toss</title>
    <script type="text/javascript">
        var heads=0, tails=0;  // Counters
        var i;  // Iteration variable
        for (i=0; i<100; i++)  
        {
            if (randNum(2) == 1)  
                heads++;
            else  
                tails++;
        }
        alert("Heads: " + heads + " and Tails: " + tails);
        function randNum(range)  
        {
            return Math.floor(range*Math.random());
        }
    </script>
</head>
</html>
```

0 = 1x
99 = \(\frac{+99x}{100}\) times!
Experiments with Flipping Coins

```html
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
  <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <meta http-equiv="Content-Style-Type" content="text/css" />
    <title>For loop example with coin toss</title>
    <script type="text/javascript">
      var heads = 0, tails = 0; // Counters
      var i; // Iteration variable

      for (i = 0; i < 100; i++) {
        if (randNum(2) == 1)
          heads++;
        else
          tails++;
      }
      alert("Heads: " + heads + " and Tails: " + tails);
      function randNum(range) {
        return Math.floor(range * Math.random());
      }
    </script>
  </head>
</html>
```
Demo—100 coin tosses

- Try the Coin Toss
  - Example 6 in Module 6 of our course Web site
Nested `for` Loop Basic Syntax

```plaintext
for (<initialization j >; <continuation j >; <next iteration j >)
{
    for (<initialization i >; <continuation i >; <next iteration i >)
    {
        <statement list>
    }
    <more statements>
}
```
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—the original trial

```html
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
<head>
  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  <meta http-equiv="Content-Style-Type" content="text/css" />
  <title>For loop example with coin toss</title>
  <script type="text/javascript">
    var heads = 0, tails = 0;
    var i, j, text=''; //Iteration vars
    for (j = 0; j < 5; j++) //Outer loop start
    {
      for (i=0; i<100; i++) //Trial line 1
      {
        if (randNum(2) == 1) //Trial line 2
          heads++; //Trial line 3
        else
          tails++; //Trial line 4
      }
      //Trial line 5
      text = text + 'Trial ' + j + ': '; //Add line to message that will print at end
      for (i = 0; i < (Math.abs(heads-50)); i++) //Trial line 6
      {
        text = text + '*'; //Add to message
      }
      text = text + '\n'; //Add line break (html <br /> cannot be used for alerts)
      heads = 0; tails = 0; //Additional
    }
    //Outer loop end
    alert(text);
    function randNum(range)
    {
      return Math.floor(range*Math.random());
    }
  </script>
</head>
</html>
```
Experiment 2—outer loop

```html
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
<head>
  <meta http-equiv="content-type" content="text/html; charset=utf-8" />  
  <meta http-equiv="Content-Style-Type" content="text/css" />
  <title>For loop example with coin toss</title>
  <script type="text/javascript">
    var heads = 0, tails = 0;
    var i, j, text='';
    //Iteration vars
    for (j = 0; j < 5; j++)
    {
      for (i=0; i<100; i++)
      {
        if (randNum(2) == 1)  //Trial line 2
          heads++;
        else
          tails++;
        text = text + 'Trial ' + j + ': '; //Add line to message that will print at end
        for (i = 0; i < (Math.abs(heads-50)); i++)
        {
          text = text + '*'; //Add to message
        }
      text = text + '\n'; //Add line break (html <br /> cannot be used for alerts)
      heads = 0; tails = 0; //Additional
    }
  }
  alert(text);
  function randNum(range)  
  {
    return Math.floor(range*Math.random());
  }
  </script>
</head>
</html>
```
Experiment 2—declare i and j

```html
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
<head>
  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  <meta http-equiv="Content-Style-Type" content="text/css" />
  <title>For loop example with coin toss</title>
  <script type="text/javascript">
    var heads = 0, tails = 0;
    var i, j, text = ''; // Iteration vars
    for (j = 0; j < 5; j++) // Outer loop start
      {
        for (i=0; i<100; i++) // Trial line 1
          {
            if (randNum(2) == 1) // Trial line 2
              heads++; // Trial line 3
            else
              tails++; // Trial line 4
            }
        text = text + 'Trial ' + j + ': '; // Add line to message that will print at end
        for (i = 0; i < (Math.abs(heads-50)); i++) // Trial line 5
          {
            text = text + '*'; // Add to message
          }
        text = text + '\n'; // Add line break (<br /> cannot be used for alerts)
        heads = 0; tails = 0; // Additional
      } // Outer loop end
    alert(text);
    function randNum(range)
      {
        return Math.floor(range*Math.random());
      }
  </script>
</head>
</html>
```
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
<head>
    <meta http-equiv="content-type" content="text/html;charset=utf-8" />
    <meta http-equiv="Content-Style-Type" content="text/css" />
    <title>For loop example with coin toss</title>
    <script type="text/javascript">
        var heads = 0, tails = 0;
        var i, j, text = ''; //Iteration vars
        for (j = 0; j < 5; j++) //Outer loop start
        {
            for (i=0; i<100; i++) //Trial line 1
            {
                if (randNum(2) == 1) //Trial line 2
                    heads++; //Trial line 3
                else //Trial line 4
                    tails++; //Trial line 5
            } //Trial line 6

            text = text + 'Trial ' + j + ': '; //Add line to message that will print at end
            for (i = 0; i < (Math.abs(heads-50)); i++)
            {
                text = text + '*'; //Add to message
            }
            text = text + '\n'; //Add line break (html <br /> cannot be used for alerts)
        } //Additional
        heads = 0; tails = 0; //Additional
        alert(text);
        function randNum(range)
        {
            return Math.floor(range*Math.random());
        }
    </script>
</head>
</html>
Experiment 2—how far from 50%?

```html
<head>
  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  <meta http-equiv="Content-Style-Type" content="text/css" />
  <title>For loop example with coin toss</title>
  <script type="text/javascript">
    var heads = 0, tails = 0;
    var i, j, text='';
    // Iteration vars
    for (j = 0; j < 5; j++) // Outer loop start
    {
      for (i=0; i<100; i++) // Trial line 1
      {
        if (randNum(2) == 1) // Trial line 2
          heads++;
        else // Trial line 3
          tails++;
        // Trial line 4
        text = text + 'Trial ' + j + ': '; // Add line to message that will print at end
        for (i = 0; i < (Math.abs(heads-50)); i++)
        {
          text = text + '*'; // Add to message
        }
        text = text + '\n'; // Add line break (html <br /> cannot be used for alerts)
        heads = 0; tails = 0; // Additional
        // Outer loop end
      }
      alert(text);
      function randNum(range)
      {
        return Math.floor(range*Math.random());
      }
    }
  </script>
</head>
```
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
A computer lets you make more mistakes faster than any invention in human history—with the possible exceptions of handguns and tequila.

~Mitche Ratcliffe