CS 5142
Scripting Languages

Introduction
Today’s Outline

- Introduction to Scripting Languages
- Course Mechanics
Scripting Languages

Facebook is written in PHP

Amazon makes heavy use of Perl CGI

BBC reports that 73% of all websites use JavaScript
Systems Languages

- Introduced as an alternative to assembly language
- Provide “higher level” instructions
  - 1 line of C code $\approx$ 3-7 assembly instructions
- Are strongly typed
  - The “type” determines how data can be used
Advantages of Strong Typing

- Can catch errors at compile time
  - Example: using a floating point instead of a pointer
- Can make large code bases more manageable
  - Clarify how things are used
- Compiler can use type information to generate more efficient code
  - Example: can generate integer instructions
Scripting Languages

- Often assume the existence of components which they “glue” together
- Different target programmer
- Tend to be weakly typed
- Usually interpreted (not compiled)
  - Tradeoff performance for expressiveness
- Encourage rapid prototyping and development
Advantages of Weak Typing

- No a-priori restrictions
- Easier to “hook things together”
  - Example: Unix shell commands, everything is just a string
- Encourages code re-use
  - Don’t need different interfaces for different types
- More succinct code
## Features for Rapid Development

<table>
<thead>
<tr>
<th>Feature</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low boiler-plate</td>
<td><code>print &quot;Ice cream!\n&quot;</code></td>
</tr>
<tr>
<td>Dynamic typing</td>
<td><code>$amount = 20 . &quot; grams&quot;;</code></td>
</tr>
<tr>
<td>Interpretation</td>
<td><code>eval &quot;print 'egg\n'&quot;;</code></td>
</tr>
<tr>
<td>String manipulation</td>
<td><code>$x = &quot;food&quot;; $x =~ s/o/e/g;</code></td>
</tr>
<tr>
<td>Associative arrays</td>
<td><code>$group{pasta} = 'carbs';</code></td>
</tr>
<tr>
<td>Properties</td>
<td><code>document.im1.src=&quot;meal.jpg&quot;;</code></td>
</tr>
<tr>
<td>Call-backs</td>
<td><code>&lt;input ... onClick=&quot;stir()&quot;&gt;</code></td>
</tr>
</tbody>
</table>
Scripting vs. Systems Languages

Language Popularity Index

- 5/10 most popular languages are scripting
- This class will cover the top 4 most popular (excluding Python)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td>C</td>
<td>17.346%</td>
<td>+1.18%</td>
<td>A</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td></td>
<td>Java</td>
<td>16.998%</td>
<td>-1.56%</td>
<td>A</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td></td>
<td>C++</td>
<td>9.825%</td>
<td>+0.68%</td>
<td>A</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>![Up arrow]</td>
<td>Objective-C</td>
<td>8.309%</td>
<td>+3.30%</td>
<td>A</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>![Down arrow]</td>
<td>C#</td>
<td>6.823%</td>
<td>-0.72%</td>
<td>A</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>![Down arrow]</td>
<td>PHP</td>
<td>5.711%</td>
<td>-0.80%</td>
<td>A</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>![Up arrow]</td>
<td>(Visual) Basic</td>
<td>5.457%</td>
<td>+0.86%</td>
<td>A</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>![Down arrow]</td>
<td>Python</td>
<td>3.819%</td>
<td>-0.76%</td>
<td>A</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td></td>
<td>Perl</td>
<td>2.805%</td>
<td>+0.57%</td>
<td>A</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>![Up arrow]</td>
<td>JavaScript</td>
<td>2.135%</td>
<td>+0.74%</td>
<td>A</td>
</tr>
<tr>
<td>11</td>
<td>10</td>
<td>![Down arrow]</td>
<td>Ruby</td>
<td>1.451%</td>
<td>+0.03%</td>
<td>A</td>
</tr>
<tr>
<td>12</td>
<td>26</td>
<td>![Up arrow]</td>
<td>Visual Basic .NET</td>
<td>1.274%</td>
<td>+0.79%</td>
<td>A</td>
</tr>
<tr>
<td>13</td>
<td>21</td>
<td>![Up arrow]</td>
<td>PL/SQL</td>
<td>1.119%</td>
<td>+0.62%</td>
<td>A</td>
</tr>
<tr>
<td>14</td>
<td>13</td>
<td>![Down arrow]</td>
<td>Delphi/Object Pascal</td>
<td>1.004%</td>
<td>-0.07%</td>
<td>A</td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td></td>
<td>Lisp</td>
<td>0.941%</td>
<td>-0.01%</td>
<td>A</td>
</tr>
<tr>
<td>16</td>
<td>24</td>
<td>![Up arrow]</td>
<td>Logo</td>
<td>0.839%</td>
<td>+0.35%</td>
<td>A--</td>
</tr>
<tr>
<td>17</td>
<td>17</td>
<td></td>
<td>Pascal</td>
<td>0.808%</td>
<td>+0.10%</td>
<td>A</td>
</tr>
<tr>
<td>18</td>
<td>18</td>
<td></td>
<td>Transact-SQL</td>
<td>0.654%</td>
<td>-0.04%</td>
<td>A--</td>
</tr>
<tr>
<td>19</td>
<td>16</td>
<td>![Down arrow]</td>
<td>Ada</td>
<td>0.649%</td>
<td>-0.10%</td>
<td>B</td>
</tr>
<tr>
<td>20</td>
<td>12</td>
<td>![Down arrow]</td>
<td>Lua</td>
<td>0.566%</td>
<td>-0.54%</td>
<td>B</td>
</tr>
</tbody>
</table>
#!/usr/bin/perl -w

%cup2g = ( flour => 110, sugar => 225, butter => 225 );
%volume = ( cup => 1, tbsp => 16, tsp => 48, ml => 236 );
%weight = ( lb => 1, oz => 16, g => 453 );

while (<>) {
    my ($qty, $unit, $ing) = /([0-9.]+) ([\w]+) ([\w]+)/;
    if ($cup2g{$ing} && $volume{$unit}) {
        $qty = 1.0 * $qty * $cup2g{$ing} / $volume{$unit};
        $unit = 'g';
    } elsif ($volume{$unit}) {
        $qty = 1.0 * $qty * $volume{ml} / $volume{$unit};
        $unit = 'ml';
    } elsif ($weight{$unit}) {
        $qty = 1.0 * $qty * $weight{g} / $weight{$unit};
        $unit = 'g';
    }
    printf("%d $unit $ing\n", $qty + .5);
}
Option Explicit

Sub LemonStar()
    Dim S As PowerPoint.Slide
    Set S = ActivePresentation.Slides( _
        ActivePresentation.Slides.Count)
    Dim I As Integer
    For I = 0 To 8
        Dim L As PowerPoint.Shape
        Const Dpi As Integer = 72 ' 72 dots per inch
        Set L = S.Shapes.AddLine( _
            BeginX:=Dpi*5, BeginY:=Dpi*3.75+I*Dpi/8, _
            EndX  :=Dpi*6, EndY  :=Dpi*4.75-I*Dpi/8)
        L.Line.ForeColor.RGB = RGB(I * 31, I * 31, 0)
    Next I
End Sub
<?php

$d = sqlite_open("data/sqlite2", 0666, $err);
if ($err) { die($err); }

sqlite_query($d, "select * from T", SQLITE_BOTH, $err);
if ($err) {
    echo "table does not yet exist, creating it ...
    $q = "create table T(I integer, S char(10))";
    sqlite_query($d, $q, SQLITE_BOTH, $err);
    if ($err) { die($err); }
    sqlite_query($d, "insert into T values(0, 'n')");
}

$rows = sqlite_query($d, "select I from T where S='n'");
$row = sqlite_fetch_array($rows, SQLITE_BOTH);
echo "T[S=n][I=="." . $row['I'] . "]; reload for ++
sqlite_query($d, "update T set I = I+1 where S='n'");
echo "delete data/sqlite2 to start over"
?>
<html>
<head><title>Form validation example</title>
<script>
function chk() {
    var v = document.myFm.num.value;
    if (v>=1 && v<=10) return true;
    alert("bad input " + v);
    return false; // abort commit
}
</script>
</head><body>
<form name="myFm" method="post" action="otherpage.htm">
Enter a number: <input size="4" type="text" name="num">
<input type="submit" value="OK" onClick="return chk()">
</form></body>
</html>
Today’s Outline

- Introduction to Scripting Languages
- Course Mechanics
Course Goals

Short-term:
- Survey of the most popular scripting languages
- Understand general PL concepts in the context of scripting

Long-term
- Use languages effectively
- Quickly learn new languages on your own
- Design and improve scripting languages
# Tentative Schedule

<table>
<thead>
<tr>
<th>Lecture topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
</tr>
<tr>
<td>End-user programming (VBA)</td>
</tr>
<tr>
<td>Objects, properties, call-backs</td>
</tr>
<tr>
<td>Textual data processing (Perl)</td>
</tr>
<tr>
<td>Contexts, objects, scripting as glue</td>
</tr>
<tr>
<td>Server-side scripting (PHP)</td>
</tr>
<tr>
<td>Client-side scripting (JavaScript)</td>
</tr>
<tr>
<td>Web applications and databases</td>
</tr>
<tr>
<td>Security for web applications</td>
</tr>
<tr>
<td>Other languages (Bash, Python, Ruby)</td>
</tr>
</tbody>
</table>
Grading

- 25% homework + 35% prelims + 40% final

- Homework
  - Due each Friday at 6pm
  - >= 1 minute late: 50% points
  - >= 3 hours late: 0%

- Contact me for circumstances beyond your control
Academic Integrity

Please see: http://cuinfo.cornell.edu/Academic/AIC.html

You may collaborate on homework, *but*:

- You must write up and turn in your own answers
- You must indicate who you collaborated with
Contact

- Instructor: Robert Soulé
  soule@cs.cornell.edu
- Office hours: Fridays after class (or by appointment)
- TA: To be determined.
No required textbooks

If you want more detail, there is a list of recommended books on the course website
Next Time

- End-User programming
- Introduction to VBA
- You will need access to Microsoft Powerpoint
  - Email me if you can’t get it