CMSC 330: Organization of Programming Languages

Introduction to Ruby
Clickers improve student engagement


Using clickers to improve student engagement and performance class.

Addison S¹, Wright A, Milner R.

Author information

Abstract
Students say

ren  
@reennnn__

Clickers are the invention of satan I'm convinced.
5:45 PM - 26 Nov 2012 · San Diego, CA, United States

Rachel Paddock  
@RachelPaddock

Whoever invented clickers.... I despise you.
11:33 AM - 29 Nov 2012

Cait Corf  
@caitcorf

BUT WHY MUST I BE SO STUPID?! The only reason I stayed is because it this class has I clickers,guess what I forgot to bring to class today?
12:18 PM - 15 Mar 2013
I have my clicker

A. True
B. False
Introduction

- Ruby is an object-oriented, imperative, dynamically typed (scripting) language

  - “I wanted a scripting language that was more powerful than Perl, and more object-oriented than Python. That's why I decided to design my own language.”

  - “I believe people want to express themselves when they program. They don't want to fight with the language. Programming languages must feel natural to programmers. I tried to make people enjoy programming and concentrate on the fun and creative part of programming when they use Ruby.”

  — Yukihiro Matsumoto (”Matz”)
Ruby

- An *object-oriented, imperative, dynamically typed (scripting) language*
  - Created in 1993 by Yukihiro Matsumoto (Matz)
  - “Ruby is designed to make programmers happy”
  - Core of Ruby on Rails web programming framework (a key to its popularity)
  - Similar in flavor to many other scripting languages
    - Much cleaner than perl
  - Full object-orientation (even primitives are objects!)
Books on Ruby

- Earlier version of Thomas book available on web
  - See course web page
Applications of Scripting Languages

- Scripting languages have many uses
  - Automating system administration
  - Automating user tasks
  - Quick-and-dirty development

- Motivating application

Text processing
% wc *
  271  674   5323 AST.c
  100  392   3219 AST.h
  117 1459  238788 AST.o
1874  5428  47461 AST_defs.c
1375  6307   53667 AST_defs.h
  371  884   9483 AST_parent.c
  810 2328  24589 AST_print.c
  640  3070  33530 AST_types.h
  285  846   7081 AST_utils.c
   59  274   2154 AST_utils.h
   50  400  28756 AST_utils.o
  866  2757  25873 Makefile
  270  725   5578 Makefile.am
  866  2743  27320 Makefile.in
   38  175   1154 alloca.c
2035  4516  47721 aloctypes.c
   86  350   3286 aloctypes.h
  104 1051   66848 aloctypes.o

...
Climate Data for IAD in August, 2005

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6A</th>
<th>6B</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>AVG</td>
<td>MX</td>
<td>2MIN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DY</td>
<td>MAX</td>
<td>MIN</td>
<td>AVG</td>
<td>DEP</td>
<td>HDD</td>
<td>CDD</td>
<td>WTR</td>
<td>SNW</td>
<td>DPTH</td>
<td>SPD</td>
<td>SPD</td>
<td>DIR</td>
<td>MIN</td>
<td>PSBL</td>
<td>S-S</td>
<td>WX</td>
<td>SPD</td>
</tr>
</tbody>
</table>

| 1 | 87 | 66 | 77 | 1 | 0 | 12 | 0.00 | 0.0 | 0 | 2.5 | 9 | 200 | M | M | 7 | 18 | 12 | 210 |
| 2 | 92 | 67 | 80 | 4 | 0 | 15 | 0.00 | 0.0 | 0 | 3.5 | 10 | 10 | M | M | 3 | 18 | 17 | 320 |
| 3 | 93 | 69 | 81 | 5 | 0 | 16 | 0.00 | 0.0 | 0 | 4.1 | 13 | 360 | M | M | 2 | 18 | 17 | 360 |
| 4 | 95 | 69 | 82 | 6 | 0 | 17 | 0.00 | 0.0 | 0 | 3.6 | 9 | 310 | M | M | 3 | 18 | 12 | 290 |
| 5 | 94 | 73 | 84 | 8 | 0 | 19 | 0.00 | 0.0 | 0 | 5.9 | 18 | 10 | M | M | 3 | 18 | 25 | 360 |
| 6 | 89 | 70 | 80 | 4 | 0 | 15 | 0.02 | 0.0 | 0 | 5.3 | 20 | 200 | M | M | 6 | 138 | 23 | 210 |
| 7 | 89 | 69 | 79 | 3 | 0 | 14 | 0.00 | 0.0 | 0 | 3.6 | 14 | 200 | M | M | 7 | 1  | 16 | 210 |
| 8 | 86 | 70 | 78 | 3 | 0 | 13 | 0.74 | 0.0 | 0 | 4.4 | 17 | 150 | M | M | 10 | 18 | 23 | 150 |
| 9 | 76 | 70 | 73 | -2| 0 | 8  | 0.19 | 0.0 | 0 | 4.1 | 9 | 90  | M | M | 9 | 18 | 13 | 90  |
| 10| 87 | 71 | 79 | 4 | 0 | 14 | 0.00 | 0.0 | 0 | 2.3 | 8 | 260 | M | M | 8 | 1  | 10 | 210 |

...
<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
</table>
| u108_S  | DC, 000, 01, 0000001, 572059, 72264, 572059, 12.6, 572059, 572059, 572059, 0, 0, 0, 0, 572059, 175306, 343213, 2006, 14762, 383, 21728, 14661, 572059, 527044, 15861, 7, 340061, 1560, 14605, 291, 1638, 10272, 45015, 16689, 3152, 446, 157, 92, 20090, 4389, 572059, 268827, 3362, 3048, 3170, 3241, 3504, 3286, 3270, 3475, 3939, 3647, 3525, 3044, 2928, 2913, 2769, 2752, 2933, 2703, 4056, 5501, 5217, 4969, 13555, 24995, 24216, 23726, 20721, 18802, 16523, 12318, 4345, 5810, 3423, 4690, 7105, 5739, 3260, 2347, 303232, 3329, 3057, 2935, 3429, 3326, 3456, 3257, 3754, 3192, 3523, 3336, 3276, 2989, 2838, 2824, 2624, 2807, 2871, 4941, 6588, 5625, 5563, 17177, 27475, 24377, 22818, 21319, 20851, 19117, 15260, 5066, 6708, 4257, 6117, 10741, 9427, 6807, 6175, 572059, 9, 536373, 370675, 115963, 55603, 60360, 57949, 129440, 122518, 3754, 3168, 22448, 9967, 4638, 14110, 16160, 165698, 61049, 47694, 13355, 71578, 60875, 10703, 33071, 35686, 7573, 28113, 248590, 108569, 47694, 60875, 140021, 115963, 58050, 21654, 36396, 57913, 10355, 4065, 6290, 47558, 25229, 22329, 24058, 13355, 10703, 70088, 65737, 37112, 21742, 12267, 9475, 9723, 2573, 2314, 760, 28625, 8207, 7469, 738, 19185, 18172, 1013, 1233, 4351, 3610, 741, 248590, 199456, 94221, 46274, 21443, 24831, 47947, 8705, 3979, 4726, 39242, 25175, 14067, 105235, 82928, 22307, 49134, 21742, 11776, 211, 11565, 9966, 1650, 86, 1564, 8316, 54, 8262, 27392, 25641, 1751, 248590, 115963, 4999, 22466, 26165, 24062, 16529, 12409, 7594, 1739, 132627, 11670, 32445, 23225, 21661, 16234, 12795, 10563, 4034, 248590, 115963, 48738, 28914, 19259, 10312, 4748, 3992, 132627, 108569, 19284, 2713, 1209, 509, 218, 125...
A Simple Example

- Let’s start with a simple Ruby program

**ruby1.rb:**

```ruby
# This is a ruby program
x = 37
y = x + 5
print(y)
print("\n")
```

```
ruby -w ruby1.rb
42
```

# This is a ruby program
x = 37
y = x + 5
print(y)
print("\n")

comments begin with #, go to end of line
variables need not be declared
no special main() function or method
line break separates expressions (can also use “;” to be safe)
Run Ruby, Run

There are two basic ways to run a Ruby program

- **ruby -w filename** – execute script in *filename*
  - tip: the `-w` will cause Ruby to print a bit more if something bad happens
  - Ruby filenames should end with `.rb` extension
- **irb** – launch interactive Ruby shell
  - Can type in Ruby programs one line at a time, and watch as each line is executed
    ```ruby
    irb(main):001:0> 3+4
    => 7
    ```
  - Can load Ruby programs via `load` command
    - Form: `load string`
    - String must be name of file containing Ruby program
    - E.g.: `load ‘foo.rb’`

- Ruby 1.9.3 is installed on Grace cluster (upgrading to 2.4)
Run Ruby, Run (cont.)

- `fxri` – launch standalone interactive Ruby shell

```bash
#!/usr/local/bin/ruby

print("Hello, world!\n")
```

![Screenshot of fxri interface](Image)
Run Ruby, Run (cont.)

Suppose you want to run a Ruby script as if it were an executable (e.g. “double-click”, or as a command)

- **Windows**
  - Must associate .rb file extension with ruby command
  - If you installed Ruby using the Windows installer, this was done automatically
  - The Ruby web site has information on how to make this association
Run Ruby, Run (cont.)

- Suppose you want to run a Ruby script as if it were an executable (cont.)
  - *nix (Linux / Unix / etc.)
    ```ruby
    #!/usr/local/bin/ruby -w
    print("Hello, world!\n")
    ```
  - The first line ("shebang") tells the system where to find the program to interpret this text file
  - Must `chmod u+x filename` first, or `chmod a+x filename` so everyone has exec permission
  - Warning: Not very portable: Depends on location of Ruby interpreter
    - `/usr/local/bin/ruby` vs. `/usr/bin/ruby` vs. `/opt/local/bin/ruby` etc.
Creating Ruby Programs

As with most programming languages, Ruby programs are text files.

- Note: there are actually different versions of “plain text”! E.g. ASCII, Unicode, Utf-8, etc.
- You won’t need to worry about this in this course.

To create a Ruby program, you can use your favorite text editor, e.g.

- notepad++ (free, much better than notepad)
- emacs (free, infinitely configurable)
- vim
- Eclipse (see web page for plugin instructions)
- Many others
Some Ruby Language Features

- Implicit declarations
  - Java, C have explicit declarations

- Dynamic typing
  - Java, C have (mostly) static typing

- Everything is an object
  - No distinction between objects and primitive data
  - Even “null” is an object (called nil in Ruby), as are classes

- No outside access to private object state
  - *Must* use getters, setters

- No method overloading

- Class-based and Mixin inheritance
Implicit vs. Explicit Declarations

- In Ruby, variables are implicitly declared
  - First use of a variable declares it and determines type
    - `x = 37;` // no declaration needed – created when assigned to
    - `y = x + 5`
      - `x, y` now exist, are integers

- Java and C/C++ use explicit variable declarations
  - Variables are named and typed before they are used
    - `int x, y;` // declaration
    - `x = 37;` // use
    - `y = x + 5;` // use
Implicit vs. Explicit Declarations

- Explicit declarations identify allowed names
  - Variables must be declared before used

```c
void foo(int y) {
    int x;
    x = y + 1;
    return x + y;
}
```

C, Java, C++, etc.
Implicit vs. Explicit Declarations

- Allowed names also declared implicitly
  - Variables do not need to be declared
    - Implicit declaration when first assigned to

```
Ruby
def foo(y)
    x = y + 1;
    return x + y;
end
```

Declared implicitly, when assigned

Use

Also: Perl, Python