



Ruby 101

Intermediate Ruby: Strings and Escapes

Strings and Escapes:

If double quoted, we can format the output.

ex.

"This string will print\n on two lines"



Escape sequences / codes:

Start with \
followed by a character

Common Escape sequences:

\n : new line
\t : tab
\b : backspace
\\ : \
\" : "

More at http://en.wikipedia.org/wiki/ASCII#ASCII_control_code_chart
CF "Escape Code" column



ANSI Terminal Escapes:

Only work on ANSI standard terminals: But they are fairly standard.

Begin with the CSI (Control Sequence Introducer):

`\e[`

Common Codes:

`\e[r;cH` -> Set cursor to row *r* and column *c*

`\e[2J` -> Clear screen

More Codes

http://en.wikipedia.org/wiki/ANSI_escape_code#CSI_codes



More ANSI Terminal Escapes:

`\e[+ # +`

- A -> Move Cursor Up # of lines
- B -> Move Cursor Down # of lines
- C -> Move Cursor Right # of lines
- D -> Move Cursor Left # of lines
- E -> Move Cursor Down # of lines to beginning
- F -> Move Cursor Up # of lines to beginning
- G -> Move Cursor to Column #

More Codes

http://en.wikipedia.org/wiki/ANSI_escape_code#CSI_codes



Expanding our View Code:

```
class BasicView
```

```
  def clear_display
    print "\e[2J"
  end
```

```
  def set_cursor row = 1, column = 1
    print "\e[#{row};#{column}H"
  end
```

```
end
```



What if we wanted to center our output?

We need to know:

Terminal size

Length of string to display

`$stdin` - Standard Input IO Stream

`$stdout` - Standard Output IO Stream

`io/console`:

adds methods for manipulating console i/o streams.

require 'io/console'

`$stdin.winsize` => returns array of [rows, columns]




```
class BasicView
```

```
...
```

```
  def center text
```

```
    columns = $stdin.winsize[1]
```

```
    text_length = text.length
```

```
    column_location = columns / 2 - text_length / 2
```

```
    "\e[#{column_location}G#{text}"
```

```
  end
```

```
end
```



What about color?

`\e[SGR_Codes m`

Codes 30-37 Set basic text colors

Codes 40-47 Set basic background colors

ex:

`\e[31;40m` -> Sets black background with red text

`\e[0m` -> Resets SGR codes

More SGR_Codes can be found here: http://en.wikipedia.org/wiki/ANSI_escape_code#graphics

n.b. Not all code are well supported.



Intermediate Ruby:

```
class BasicView
  def red text
    "\e[31;40m#{text}\e[0m"
  end
end
```



Putting it all together:

view.rb

```
class FileDialogView < BasicView
  def display
    puts red(center("Select an Apache log file."))
  end
end
```



Putting it all together:

controller.rb

```
def initialize
  @log_file = LogFile.new
  @current_view = FileDialogView.new
  @current_view.clear_display
  @current_view.set_cursor
end
```

